OmicsBox User Manual

Version {{ version }}

None

Table of contents

1. Welcome to OmicsBox - Your Gateway to Streamlined Bioinformatics Analysis

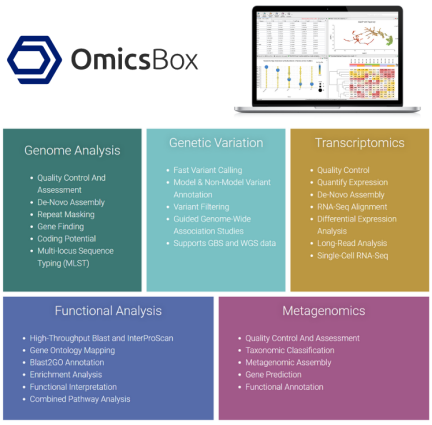
OmicsBox is your comprehensive bioinformatics solution for genomics data analysis. In the ever-evolving realm of genomics research, OmicsBox stands as a beacon of simplicity and efficiency, empowering both seasoned experts and newcomers to extract valuable biological insights from their data effortlessly.

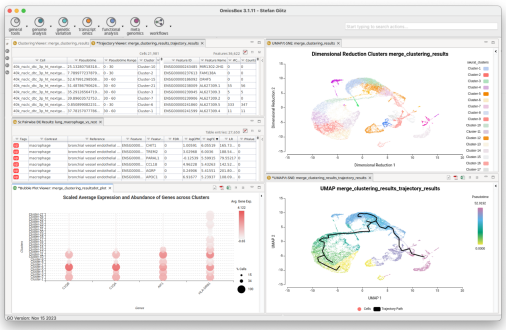
1.1 Bioinformatics Made Easy

OmicsBox serves as a valuable tool for simplifying the intricate field of genomics analysis. With its precise design and user-friendly interface, it transforms complex tasks into accessible processes, making it an ideal companion for both exploring new genomes and refining analytical skills.

1.2 Desktop Powerhouse

OmicsBox is a sophisticated desktop application designed to meet the needs of industry professionals, academic researchers, and government scientists. It provides access to advanced bioinformatics tools, enabling users to perform intricate analyses directly from their standard desktop computers.





1.3 Unleash the Power of Genomics

OmicsBox provides a user-friendly interface for bioinformatics work, allowing users to create workflows, run advanced tools, and visualize outcomes with ease. Its intuitive design reduces the learning curve, resulting in faster, more reliable results that can be reproduced consistently.

1.4 Seamless Integration, Minimal Setup

OmicsBox is designed to minimize setup hassles and facilitate effortless updates through automatic processes. It eliminates the need for high-performance computing facilities, extensive computational expertise, and constant maintenance, making it an ideal tool to simplify your bioinformatics journey.

1.5 Explore OmicsBox Modules

Discover the versatile toolkit of OmicsBox, which offers a range of modules designed to meet your research requirements:

- Genome Analysis
- Genetic Variation
- Transcriptomics
- Functional Analysis
- Metagenomics

1.6 About BioBam - Pioneers in Functional Genomics

OmicsBox is a product of BioBam, an internationally recognized leader in functional genomics. The company has received over 15,000 scientific research citations, demonstrating its strong commitment to advancing genomics research. BioBam Bioinformatics takes pride in developing and maintaining cutting-edge software solutions such as OmicsBox, OmicsCloud, and Blast2GO.

Join the global community of researchers who trust OmicsBox to accelerate their genomics research. We're headquartered in Spain and dedicated to providing bioinformatics solutions, cloud services, consulting, and analysis services to fuel your scientific endeavors.

Welcome to OmicsBox, where the complexities of bioinformatics are simplified, making genomics analysis accessible to everyone.

2. Setup

2.1 Software Installation and Activation

2.1.1 Installation

OmicsBox is a desktop application and the installer can be downloaded here.

Windows

Once downloaded double click on the installer and follow the instructions. OmicsBox will be installed in `C:\Users\[username]\AppData\Local\OmicsBox` by default.

INSTALL IN UNATTENDED MODE

```
.\OmicsBox_windows-x64_2_0_36.exe -q -dir "C:\OmicsBox" -console
```

The parameters are:

`-q` run in unattended mode

`-dir "C:\OmicsBox"` set the installation path. We recommend setting a common path where all users have read/write permissions. This will allow the tool to update itself if there is an update when started. An alternative is to install it on each user's home folder. If this parameter is not set, the installation is done in the default installation path mentioned above.

`-console` see the installation progress in the terminal. Installation is done when you see the "Finishing installation ..." message.

Linux

Extract the installer from the folder, double click on the .sh file and follow the instructions. OmicsBox will be installed in `/home/[username]/OmicsBox` by default.

From the terminal

```
unzip OmicsBox_unix_2_0_36.zip
sudo chmod +x OmicsBox_unix_2_0_36.sh
./OmicsBox_unix_2_0_36.sh
```

MacOS

Once downloaded double click on the .dmg file and the installer will open (Figure 2).

Double click on the OmicsBox installation App.

If Apple cannot check the OmicsBox installer for malicious software a message will be displayed (Figure 3). If there is no Open button, try opening the OmicsBox installer App using right-click and open. You should see the same warning message, but this time there should be an option to open.

Click Open and follow the instructions to complete the installation. OmicsBox will be installed in `/Applications/OmicsBox` by default

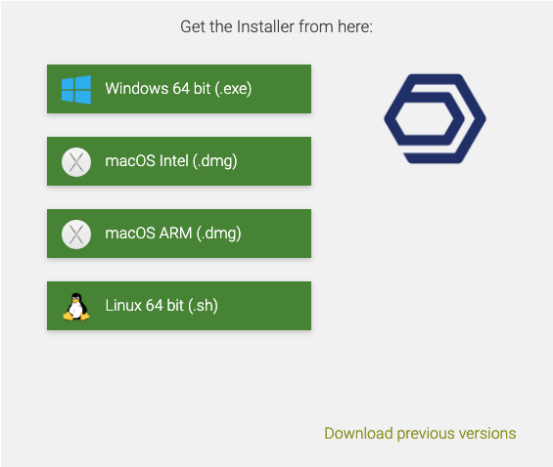


Figure 1: *Download OmicsBox Installer*

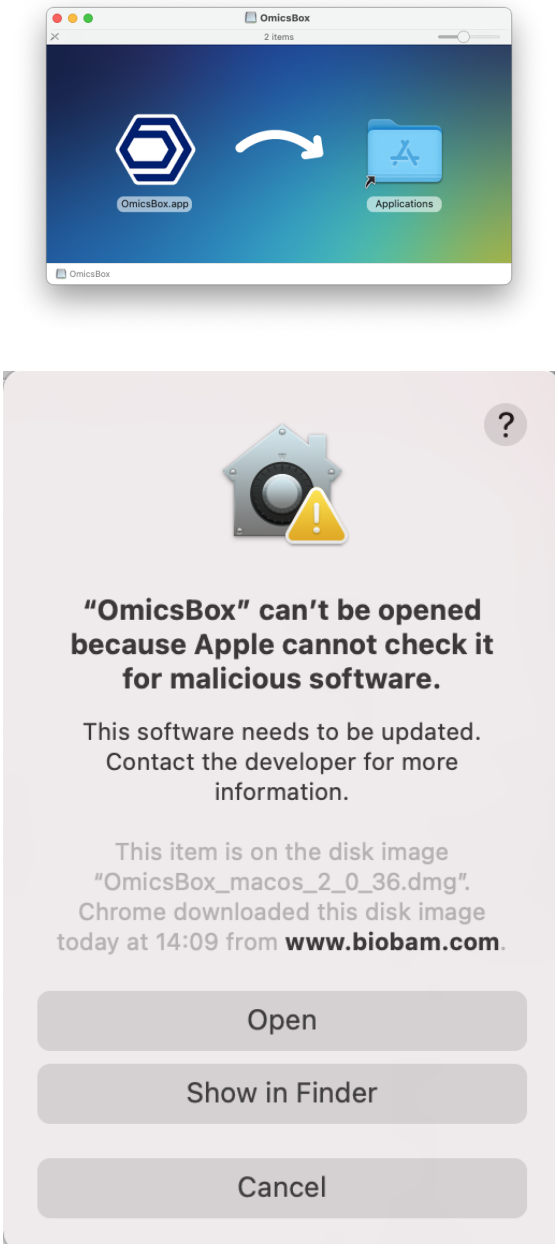


Figure 3: *Apple Message*

2.1.2 Activation

The software asks for activation the first time it is opened.

If you only need to open OmicsBox files but won't run any analyses, the software can be used in Limited Mode, without activation. The only available options will be related to loading and visualizing OmicsBox projects. This mode can be entered by clicking on the "Limited Mode" button in the Activation Window.

OmicsBox can be fully activated with a subscription key, which can be obtained from the BioBam website. A subscription can activate the different modules of the software that have been purchased. Additionally, it will give access to the cloud services like analyzing the data in the BioBam Cloud infrastructure, updates and support. More details about the advantages of a subscription can be found online on our website.

A free Trial of a subscription can be requested via the website as well.

The software requires access to the Internet for activation. If necessary, proxy settings can be adjusted directly from the activation dialog.

Figure 4: *Activation dialog of OmicsBox: Full or Limited Mode*

2.1.3 OmicsBox Login Dialog

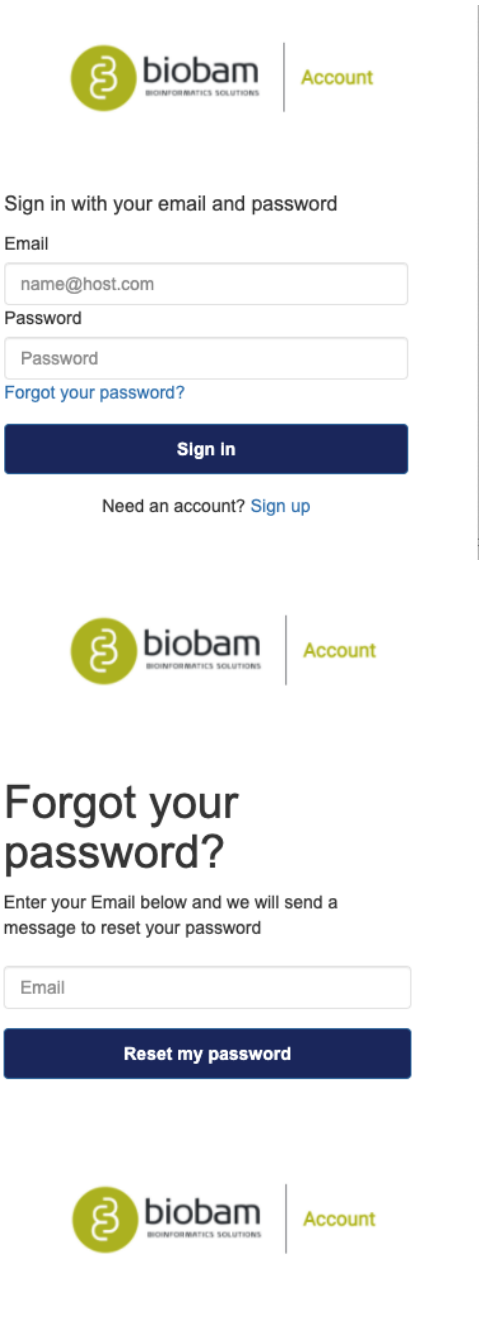
Every user of OmicsBox can have an individual account, separate from the subscription activation.

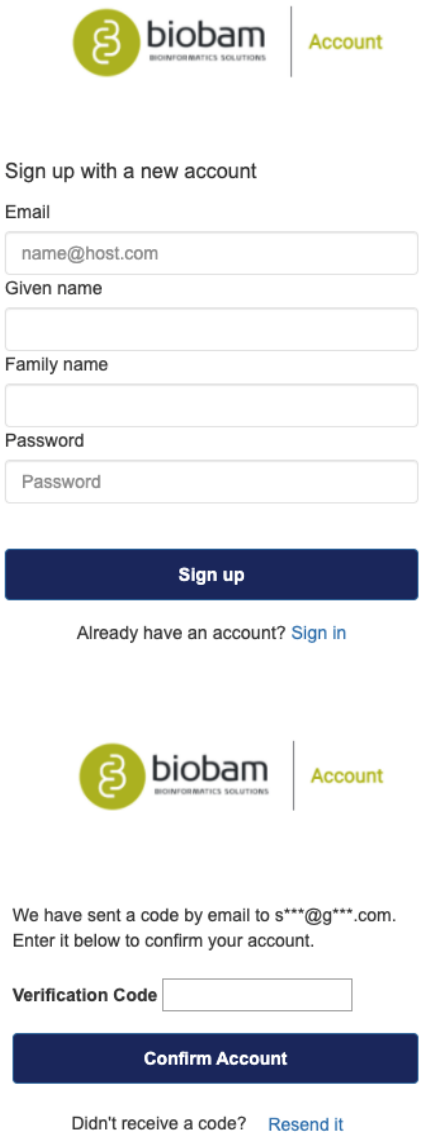
When the user logs in to an OmicsBox installation, it gets associated in the subscription account as a user of the subscription. The associated users can then be reviewed and managed by the subscription owner. With this, we offer a more advanced subscription usage management for Cloud Units consumption and control on who can use the software. The subscription owner can limit the access to cloud units by user for the shared OmicsBox installation while seeing how the consumption evolves. By default, any user is automatically associated to the subscription. This can be modified from the BioBam Account website.

The login account and credentials (email and password) are the same as the ones used on the BioBam Account website. If you do not have a BioBam Account you can now create one from within OmicsBox using the "Sign up" option.

The Sign In dialog allows you to do the following:

- **Sign-Up:** Create a new user with a new password. This option is possible if a user with the same email address does not exist. The password has to fulfil the indicated security requirements. The sign-up process has to be completed with a verification code sent via email. Make sure to check the classified and spam folders in your email account.
- **Sign-In:** Authenticate with your login by providing your account email and password.
- **Password Recovery:** This option allows to set a new password for an existing account, by confirming a verification code sent via email. Make sure to check the classified and spam folders in your email account.
- **Manage License Key:** This option allows you to change your OmicsBox Subscription Licence Key without logging in.





2.1.4 Network License

OmicsBox can also be activated with the license server. Various clients can share a pool of licenses on a license server in the local network. To activate a network license, click **Network License**on the bottom left in the activation dialog and provide the license server IP address and cloud key if you own one. The cloud key allows to use CloudBlast and CloudIPS, all the other cloud features are already available without a key.

2.2 Technical Requirements and Support Services

2.2.1 Technical Requirements

- **Processor:** 64-bit, 1.6GHz or faster processor with 2 or more cores recommended.
- **Operating System:**
- **Windows:** Microsoft Windows 10 or later (64-bit).
- **Mac:** macOS 12 or later, compatible with both Intel and ARM architectures.
- **Linux (64-bit):** Ubuntu 18.04 and later, RHEL 8.9 and later. (Note: While the software is expected to function on other recent Linux distributions, we do not officially support or guarantee this. Linux systems must have GTK library version 3.22+ installed.)
- **Memory:** Minimum of 4GB RAM (8GB or more recommended).
- **Disk Space:**
- 2GB of available hard disk space for software installation.
- An additional 4GB for optional content downloads and temporary files, depending on the volume of input data.
- **Display Resolution:** Minimum resolution of 1280x800 (at 100% DPI scaling).
- **Internet Connection:**
- A continuous internet connection is required for product activation, content downloads, and cloud connections.
- We recommend a wired ethernet connection with a minimum bandwidth of 30 Mbps for file uploads and downloads. You can test your connection speed using fast.com or speedtest.net.
- **Installation:**
- The software must be installed on a local drive.
- An SSD is recommended for optimal performance.
- Installation on flash drives or network drives is not advisable.
- **Virtual Environments:** Supported with additional licensing conditions.

These requirements ensure optimal performance and user experience when using our software.

2.2.2 Language Versions

OmicsBox is only available in the English language.

2.2.3 Documentation

User Manuals online only at: https://help.biobam.com

2.2.4 Supported import/export formats

See user manual here.

2.2.5 Warranty Service and Technical Support

OmicsBox Subscription and Support

The OmicsBox subscription, delivered via email, includes comprehensive technical and bioinformatics support at no additional cost. Our primary goal is to ensure users can successfully utilize OmicsBox to its fullest potential. To achieve this, we provide several layers of support and resources:

1. Technical and Bioinformatics Support

Our dedicated support team, composed of experts in various bioinformatics fields covered by OmicsBox, is available to assist with any issues or questions users may have. This team ensures a timely and effective resolution to all user inquiries, with a guaranteed response time within a maximum of 48 hours.

2. Support Channels

- **Email Support:** Users can reach our support team directly via email at support@biobam.com. This channel is monitored continuously to ensure prompt responses.
- **Support Tickets with Online Portal:** Users can also submit support tickets through our online portal at https://account.biobam.com/ . This portal allows users to track the status of their inquiries and access a community of other OmicsBox users for shared knowledge and support.

3. Community and Knowledge Base

- **Online Community:** Through our online portal, users can engage with a community of fellow OmicsBox users. This community is a valuable resource for sharing experiences, solutions, and best practices.
- **Knowledge Base:** The online portal also includes a comprehensive knowledge base with FAQs, user guides, tutorials, and other documentation to help users troubleshoot common issues and learn how to maximize their use of OmicsBox.

4. Availability

This robust support service is available throughout the entire subscription period, ensuring users have continuous access to the help they need to make the most of OmicsBox.

5. Service Commitment

We are committed to providing exceptional support to our users, with the following service commitments:

- **Expert Assistance:** Access to a team of bioinformatics experts ready to assist with any technical or bioinformatics-related questions.
- **Timely Responses:** Guaranteed response to all inquiries within 48 hours.
- **Continuous Access:** Support available throughout the subscription period via email and our online portal.

2.2.6 Java

OmicsBox is built using Java technology and the installation comes bundled with a Java Runtime Environment (JRE) needed to run the Software. This JRE will not interfere with existing JREs on your computer and will only be used to run OmicsBox.

2.2.7 Scope of requirements

All system requirements on this page are requirements for the most recent versions of the products. For requirements for older versions, please refer to the user manual for that specific version.

2.3 Uninstalling OmicsBox

OmicsBox can be uninstalled from the different operating systems.
Below one can find instructions on how to uninstall OmicsBox.

2.3.1 Windows

On the folder where OmicsBox has been installed you will find an uninstall.exe file.
Double-click on the .exe file and follow the wizard to complete.

By default, the OmicsBox folder is located in Local folder (C:/Users/[username]/AppData/Local/OmicsBox).

It can also be uninstalled directly from **Programs** in the **Control Panel.**

2.3.2 Linux

On the folder where OmicsBox has been installed you will find an uninstall.sh file.
Open a command line, run the command and follow the wizard to complete:

```
./uninstall.sh
```

By default, the OmicsBox folder is located in your home directory (/home/[username]/OmicsBox).

2.3.3 MacOS

Move the whole OmicsBox folder to trash.

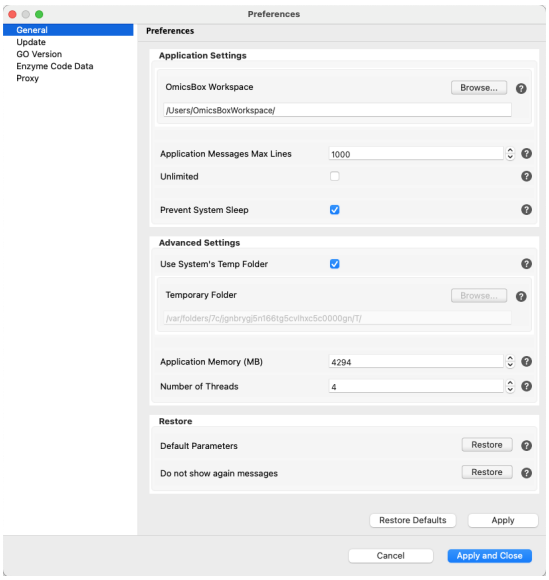
By default, the OmicsBox folder is located in the Applications directory (/Applications/OmicsBox).

3. OmicsBox Application

3.1 OmicsBox Application

3.2 Preferences

3.2.1 General Preferences



3.2.2 Update

OmicsBox allows automatic software updates during the application startup. These updates contain improvements, new features or bug fixes. It is possible to choose if you want to be notified of new updates or if you want to install software updates automatically (recommended).

It is also possible to specify the update behaviour of installed Apps. We differentiate between "Featured" and normal Apps. New "featured" Apps can be installed and updated automatically. Normal, non-featured Apps have to be installed manually but can be updated automatically.

3.2.3 GO Version

For the Functional Analysis Module OmicsBox contains the Gene Ontology database and all the information necessary to perform the mapping step i.e. to be able to link the different protein IDs to the functional information of the Gene Ontology database (see Gene Ontology Mapping section).

Here one can select the GO version available on OmicsBox servers as well as the corresponding .obo file to be used in the mapping step.

LOCAL OMICSBOX DATABASE

Local OmicsBox database installation: If you are interested in installing your own OmicsBox database locally with the aim to not depend on the OmicsBox server, you can find a tutorial on the OmicsBox website in the download section including a step-by-step installation guide. Basically will need a MySql server, the latest GO database dump and some additional "mapping tables" (NCBI and PIR flat-files). By following several few steps this data is imported into your database.

3.2.4 Enzyme Code Data

In OmicsBox it is possible to provide a file with the corresponding Enzyme Codes.

Figure 4: *Wizard to configure the OmicsBox update behaviour*

3.2.5 Proxy Settings

If a proxy server or a firewall is used to access the internet here you can define the proxy settings. An HTTP or a Socks proxy can be configured. In this window, you can configure the proxy settings only for OmicsBox and this will overcome the system-wide settings. If the Use Direct Connection check box is selected, the application will try to connect directly to the internet skipping any system settings. To use your defined proxy settings select the HTTP or Socks Proxy check box and complete the required fields.

CUSTOM CA ROOT CERTIFICATES

This option allows the import of custom CA root certificates to the OmicsBox trusted entities. It is located in the same Proxy setting page.

A certificate authority (CA) is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate (Common Name or CN in a certificate). A CA acts as a trusted third-party and allows OmicsBox to rely on the packets received through the connection to OmicsCloud or other internet sites. The format of these certificates is specified by the X.509 standard.

In a normal configuration, a secure connection is established between the end client (OmicsBox) and the server (e.g.: OmicsCloud) and the packages travel encrypted between client and server. In some custom firewall configurations where all traffic going through the network is inspected by the firewall, a secure connection is established between the client application and the firewall, and another connection is established between the firewall and the servers. The firewall can act as a man-in-the-middle to inspect the packages and will re-encrypt them using its own certificate.

This may cause connection problems if OmicsBox does not recognize the certificate used in the firewall as a trusted entity. The IT department of the institution will know if a custom certificate is used and can provide you with the CA root of this certificate, or the certificate itself, to be added to OmicsBox. This is usually a .crt file that can be provided in the wizard page.

It can also be obtained from a regular web browser by opening a page that is known to be inspected by the firewall, for example the connection to the OmicsCloud https://cloud.biobam.com by clicking on the padlock next to the url address the certificate option shows the path to the certificate and the signing authority. On our servers, the CA root entity is either *Amazon Root CA 1* or *GlobalSign Root CA*. Something else is an indicator of a custom CA Root certificate.

In **Windows and Linux**, the Root certificate can be saved from within the browser.

In **MacOS**, this needs to be searched in the keychain app and exported from there. Once this file is exported, it can be directly imported in OmicsBox Proxy preferences page.

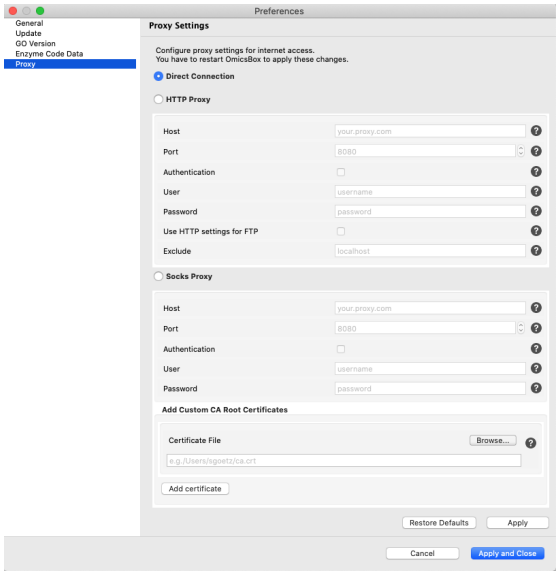
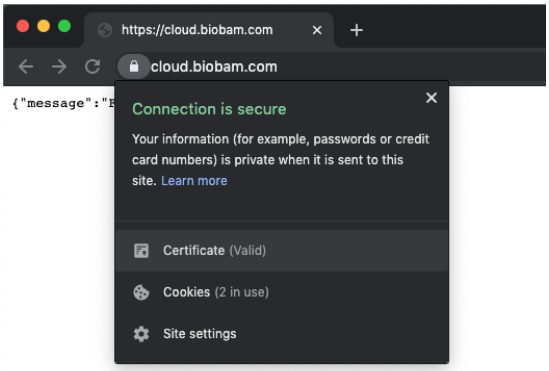
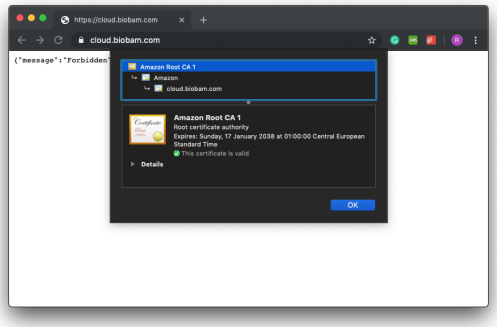


Figure 5: *Proxy settings dialog*





Whitelist domains for a Firewall

OmicsBox relies on a hybrid structure, melding desktop and cloud computing via Amazon Web Services (AWS). This setup requires uninterrupted online access to execute tasks and retrieve data from various resources, including key sites like Gene Ontology.

Given the cloud-based nature of our services, OmicsBox interacts with API endpoints with dynamic IPs, necessitating certain network allowances for full functionality.

Your network's firewall should explicitly allow outbound traffic to the following host domains to ensure proper operation. The *Core* category will make OmicsBox work with our cloud infrastructure. The *Optional* category entries include domains for third-party public services and resources used in OmicsBox.

Domain	Category	Description
*.biobam.com	Core	BioBam Resources
*.amazonaws.com	Core	BioBam Backend Infrastructure at AWS
*.cloudfront.net	Core	BioBam Backend Infrastructure at AWS (CDN)
*.salesforce.com	Core	In-App BioBam Support Mail
blast.ncbi.nlm.nih.gov	Optional	NIH NCBI BLAST Public Service
*.ebi.ac.uk	Optional	EMBL-EBI Public InterProScan Service
*.ensembl.org	Optional	Ensembl BioMart Public Services
geneontology.org ftp.geneontology.org	Optional	Gene Ontology Resources

This whitelisting is crucial for the software to communicate effectively with the necessary external resources. Permitting access to multiple subdomain levels (e.g., a.b.c.biobam.com) within the main domains is also important.

By configuring your firewall to recognize and authorize these domains, you help guarantee the smooth, uninterrupted operation of OmicsBox within your infrastructure.

3.3 Cloud Computation and Storage

3.3.1 Introduction

OmicsBox utilizes OmicsCloud, BioBam's Scientific Cloud Computing Platform, for executing many bioinformatics tools, providing a secure and scalable solution for data analysis. OmicsCloud is an AWS-based system designed for high-performance computing (HPC), optimized to support demanding bioinformatics algorithms efficiently. Users of OmicsBox can execute these bioinformatics tools and pipelines without the complexities of setup or maintenance, enabling scalable and parallelized cloud data analysis.

Additionally, OmicsBox offers CloudStorage services allowing users to save local storage space by storing data directly in the cloud, facilitating easy data access and sharing among colleagues.

3.3.2 Cloud Computation

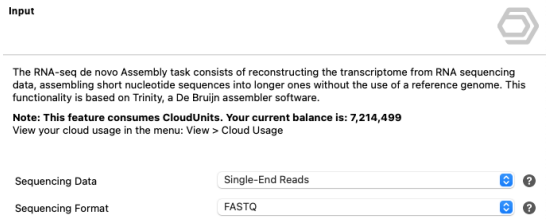
Cloud Units

OmicsBox offers a variety of tools that operate through cloud computing, incurring costs associated with CPU usage, data storage, and transfer. These expenses are quantified as Cloud Units, which reflect the consumption of computational resources like CPU time and data volume. The demand for computational power varies across different tools based on their algorithms, input data, and parameters set by users.

Despite these costs, most tools within OmicsBox can be accessed at no additional charge with an active subscription.

Currently, OmicsBox charges for cloud usage only for features related to sequence **Alignments** and **Assembly** (refer to the list below). The consumption of Cloud Units by these features is based on the duration (CPU seconds) required for their completion. All other functionalities within OmicsBox are available at no additional cost as part of your subscription.

- Functional Annotation Module
- **Alignments**: Blast, Custom Database Blast (used also via KEGG), Diamond and InterProScan
- Genome Analysis Module
- **Assembly**: Abyss, Spades, Flye
- **Alignments**: BWA, Bowtie
- Transcriptomics Module
- **Assembly**: Trinity
- **Alignments**: Star, BWA
- Metagenomics Module:
- **Assembly**: MegaHit, Meta-Spades



3.3.3 Cloud Files

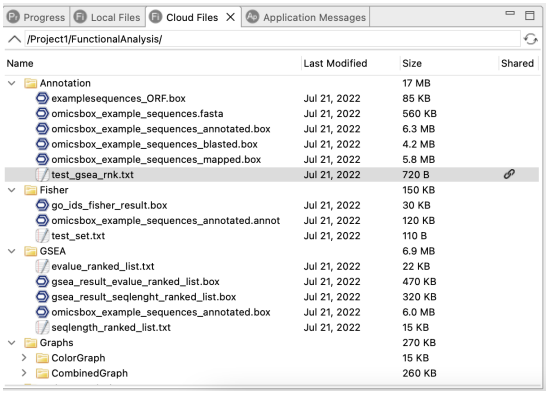
The "Cloud Files" view allows to navigate through the files stored in the cloud. This view can be opened from the menu View > Cloud Files.
Every user has an individual space in the cloud. The contextual menu allows to manage the files available in the cloud.
Drag&Drop can be used to copy a file or a folder either from the local computer or from the "Local Files" tab into the cloud.

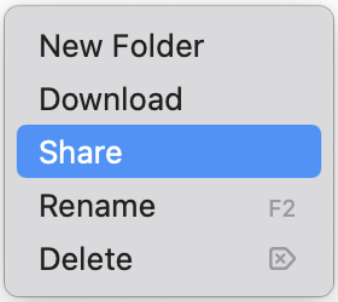
Share Files

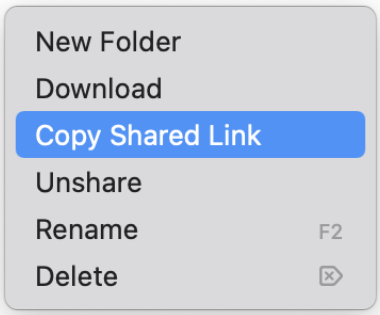
The "Share" option in the Cloud Files context menu allows to share private files with others using a link.
When a file is shared a link icon is displayed for that file in the "Shared" column.
Anyone with the shared file link can download the file using a web browser.
If a file is already shared, the "Copy Shared Link" menu option allows to copy the same link for the file.
The "Unshare" option can be used to make a file private again.
A link is not the same between shares if a file has been unshared. It is different.

Currently there are some limitations regarding the sharing functionality:

- You can share files, not directories.
- You can get a link to a file, but cannot share the file with another specific user only.







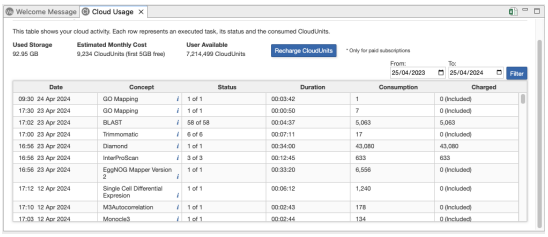
3.3.4 Review Cloud Usage

The computation and storage costs can be seen in the "Cloud Usage" view. This view can be opened from the menu View > Cloud Usage.

This view shows:

- Used Storage: The amount of storage that the user's data occupies in the cloud.
- Estimated Monthly Cost: The number of units that will be charged by the end of the month for the current amount of data stored. The first 5GB are free for each user every month. The data cost is measured daily.
- Available CloudUnits.

For more details on pricing and cloud units consumption please visit the Cloud Computation page on our website.



3.4 User Interface

3.4.1 User Interface

Introduction

This section describes the main components of the OmicsBox application interface.

The interface is organized into the following elements:

1. **Menu Bar:** The main application menu contains three sections:
    - File: Provides options to open, save, and close OmicsBox files. It also allows you to manage your license and open the application preferences.
    - View: Allows access to utility tabs such as the File Manager, Application Messages, or the Java Memory Monitor.
    - Help: Offers access to support resources. From here, you can review your subscription details, send an email to th Support Team, access your BioBam Account, open the OmicsBox User Manual, and more.
2. **Main Analysis Icons:** These icons provide access to the main OmicsBox Modules. Clicking on a module icon displays the bioinformatics analyses available within that module.
    - General Tools: Includes actions such as creating Venn diagrams, performing Fastq quality control and preprocessing, and opening the Genome Browser, etc.
    - Genome Analysis Module: Supports genome characterization and analysis, from raw reads to gene structures. Tools include eukaryotic and prokaryotic gene finding, repeat masking, DNA-Seq de novo assembly, and more.
    - Genetic Variation Module: Provides tools to identify and analyze genetic variation. Includes variant calling and filtering, variant annotation, genome-wide association studies (GWAS), and more.
    - Transcriptomics Module: Processes RNA-Seq data from raw reads to functional analysis. Functions include RNA-Seq de novo assembly, alignments, count table creation, differential expression analysis, single-cell RNA-Seq analysis, and long-reads analysis, and more.
    - Functional Analysis Module: Offers tools for functional characterization of sequences such as BLAST, InterProScan, GO Mapping, and Annotation, EggNOG, and more. Pathway annotation with Reactome and KEGG is also supported.
    - Metagenomics Module: Enables execution of analyses in a semi-automated way using predefined workflows.
    - Workflows: Enables execution of analyses in a semi-automated way using predefined workflows. It is also possible to create custom workflows de novo by combining individual analysis steps.
3. **Application Tabs:**
    - Progress: Displays running, queued, and finished jobs. Jobs can be stopped, and logs can be opened in a separate viewer. Each finished job shows its runtime. Individual jobs can be removed with the cross button, or all finished jobs can be cleared using the triangle button in the top-left corner.
    - Local Files: Allows navigation through local directories to view, open, and organize OmicsBox files. A context menu is available by right-clicking on files (Figure 2). The Merge option can combine multiple files of the same type (not available for all data types).
    - Cloud Files: Provides access to files stored in the cloud.
    - Application Messages: Shows progress updates and general messages from the application.
    - Job Messages: Displays additional logs for analysis steps. Logs can be accessed via the "Show Job Progress" icon next to the corresponding progress bar.
4. **Data Tabs:** Results generated by OmicsBox tools are displayed in two viewers:
    - Main Viewer: Displays the main results table of an analysis. It includes saving options (top-right corner) and a Side Panel (right side) with sections for Actions, Charts, and Export.
    - Side Viewer: A smaller viewer in the bottom-right corner. It displays secondary results from the Main Viewer, such as charts or summary reports.

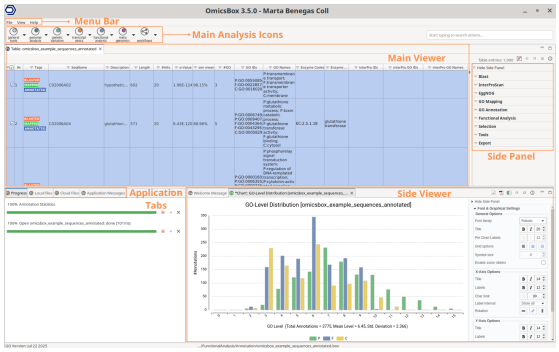


Figure 1: *OmicsBox Main User Interface*

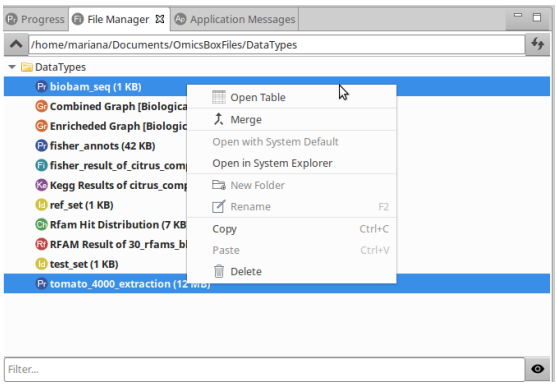


Figure 2: *File Manager menu*

Main Results Viewer

Results from bioinformatic analyses are displayed in the Main Viewer as a spreadsheet-style table (Figure 3). Columns can contain strings, numbers, or tags, this last one representing the status of each row depending on the object type.

Columns can be filtered by clicking on the filter icon to the left of the column name. They can be sorted by clicking on the column header. A right-click on any column header opens a checkbox menu to show or hide specific columns.

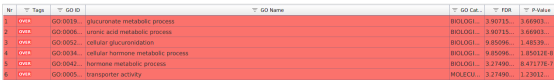


Figure 3: *OmicsBox Results Table*

TABLE CONTEXT MENU

Rows in the results table can be marked individually or in groups. Right-clicking on a marked row opens the context menu (Figure 4). * To select multiple rows, hold Ctrl (Windows/Linux) or ⌘ Command (Mac) while clicking. * To select all unfiltered rows, use Ctrl+A (Windows/Linux) or ⌘+A (Mac).

The context menu actions apply to the marked row(s). Actions at the top are specific to the type of result; their detailed explanation can be found in the corresponding manual section. Actions at the bottom are common to all OmicsBox result objects:

- **Extract Selection to new Tab:** Creates a new project including the marked rows. Available for most, but not all, OmicsBox objects.
- **Copy Row:** Copies the marked rows to the clipboard in tabular format.
- **Copy Cell:** Copies the content of a specific table value to the clipboard.
- **Create ID List of Column: [Column Name]**: Generated an ID list of a specific column and opens it in a new tab in the Main Viewer. ID-Lists can be used in some OmicsBox tools like Fisher's Exact Test or Heatmaps.
- **Create ID Value-List of [Column Name] and [Column Name]**: Opens an ID-List with two specific columns on a new tab.
- **Create Distribution Chart of Column: [Column Name]**: Generates a distribution chart of a certain column.
- **Create Category Chart of Column: [Column Name] and: [Column Name]**: Creates a category chart of two columns.

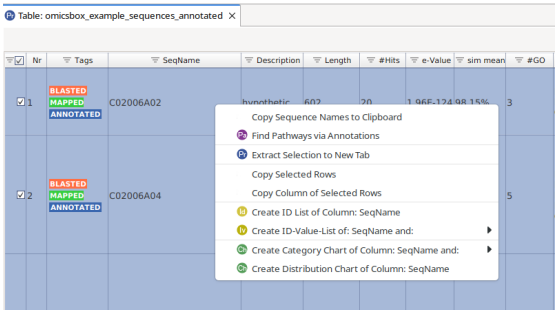


Figure 4: *Table Context Menu*

Filter Rows

The OmicsBox table allows filtering rows (Figure 5) based on search criteria defined for each column. A small filter icon at the left in each column header opens a context menu with filter options.

- Filters can be applied in various columns and are joined via an AND condition.
- Filter options depend on the column type. For example:
  - Numeric columns support conditions such as "Greater Than" or "Smaller Than".
  - String columns support conditions such as "Starts With" or "Contains".
- When a filter is applied on a column the filter icon turns red. Double-clicking the icon will remove the filter.
- A status message in the top-right corner shows how many sequences match the applied filters.
- The button next to the status message (crossed-out document icon) removes all filters at once.

Note: Algorithms are always executed on all rows of the table, not only on the filtered rows. To execute further analysis only on a subset, extract the desired rows to a new tab.

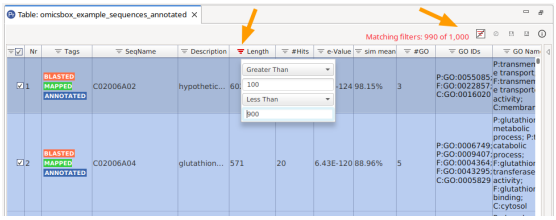


Figure 5: *Filter Criteria*

Configure Columns

This feature allows hiding columns in the results table.
Right-click on any column header to open a menu and select the columns you want to hide (Figure 6).

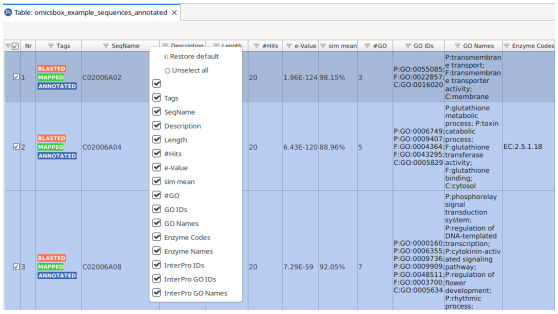


Figure 6: *Hide Columns*

Auto-Save

OmicsBox supports automatic and continuous saving of results after a defined time interval (Figure 7).

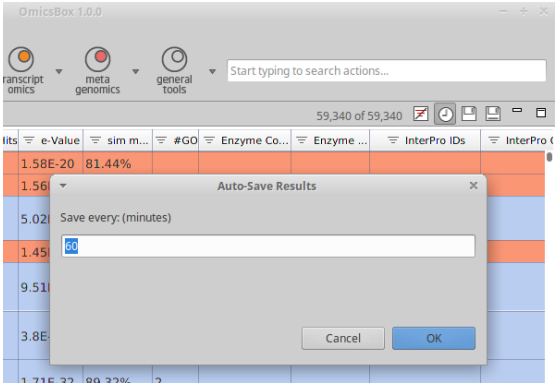


Figure 7: *Auto-save*

Export

Data stored in OmicsBox objects can be exported to external formats (e.g., FASTA, text, etc.) via the Export section of the Side Panel. Some export options are specific to the object type, whereas the general Export Table option is available for all objects.

The Export Table function exports all currently visible columns in the Main Viewer into a tab-separated values (TSV) file. Alternatively, you can manually select columns and customize their display names (Figure 8).

The Export Table will export all currently visible columns of the Main Viewer into a tab-separated values (TSV) file. It is also possible to enable column selection to choose specific columns and customize their display names.

- The left area of the wizard lists all available columns.
- The right area lists the columns selected for export.
- Click a column name and use the > and < icons to add or remove columns.
- Click a column name and use the up and down arrows to change column order.
- Click a column name and use the Rename Column Header option to edit column names before export.

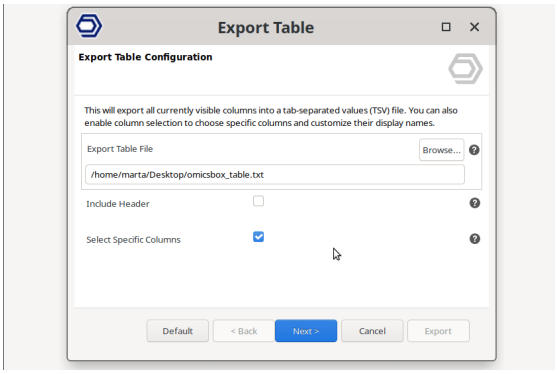


Figure 8: *Export Table Configuration Wizard.*

File Types

All OmicsBox results can be saved as .box files, which can be reopened within OmicsBox. File icons differ depending on the type of object (Figure 9).

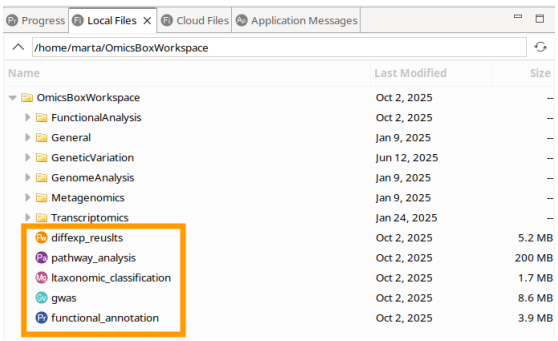


Figure 9: *Different OmicsBox object types.*

3.4.2 File Menu

Main Menu Items

	Description
Recent files	Allows reopening recently closed projects.
Open file	Open an OmicsBox project (.b2g/.box).
Save and Save as	Save the current project.
Close Tab	Close the selected tab.
Load	Allows loading ID and ID-Value lists into OmicsBox.
Export	Allows exporting the desired information to some file types, such as text or GFF.
Manage License	Check the license you currently have and change the activation key.
Preferences	Set the OmicsBox configuration.

DATA IMPORT AND EXPORT

Under the File menu and the Tools sub-menu, there are several useful features that can be used to manipulate sequence data.

LOAD

1. Extract and import sequences from a FASTA and a GFF/GTF file (figure 1). For further information, please link here.
2. Load Blast and InterProScan results.
3. Load ID lists.
4. Load GTF/GFF2/GFF3
5. Load Accession List: Load Gene Ontology annotations via an Accession list.
6. Load GeneSymbol List: Load Gene Ontology annotations via a GeneSymbol list.
7. Load GI-List: load Gene Ontology annotations via a GenInfo Identifier (gi) list. Please consider the identifier to be between vertical bar e.g. gi|356569257|.
8. Load Data from BioMart: Load Gene Ontology annotations from BioMart. For further details on how to load annotations, link here.
9. Load EggNOG, Metagenomics and PfamScan annotations.
10. Load metagenomic Kraken data.
11. Load Count Tables and Differential Expression results.
12. Load BAM and VCF files.

The Accession List and the GeneSymbol file should contain two columns (separated by tabs) per line. The first column the accession id or gene symbol and the second column may contain the corresponding taxonomy. The second column is optional.

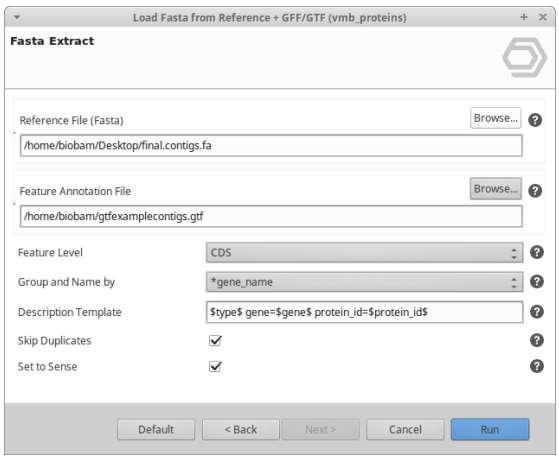


Figure 1: *Extract and import sequences from a FASTA and a GFF/GTF file.*

EXPORT MENU

The icon of each menu item indicates which object type is suitable for this menu item.

	Description
Export Table	Export the current Main Sequence Table for the selected sequences.
Generic Export	This option allows you to export all the desired information to a text file.
Export as Fasta	Export sequences of this project in fasta format.
Export as GFF	Export the annotations of this project as GFF file.
Export Blast Top-Hits	It will export the best-blast-hit for each sequence, this is the hit with the lowest e-value.
Export Mapping Results	Allows to export all the information obtained and used during the Gene Ontology mapping process as GFF formatted text file.
Export Annotations (.annot)	
Export Annotation Descriptions	
Export Annotations in GO Annotation File Format (GAF v.2)	
Export GO Propagation	Exports the GO parents up to the root for the annotated sequences.
Export Sequences per GO (Gene Sets)	
Export Metagenomics GO Annotations	
Export Kraken Data	Export the Kraken data as text file.
Export GFF	This option is only visible if a GFF file is loaded in OmicsBox or if a GFF has been generated from Gene Finding. It is also possible to export a GFF with GO terms. For further details, link here.
Export KEGG Data	KEGG related information as tab separated text file.

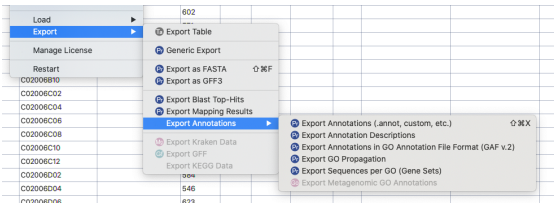
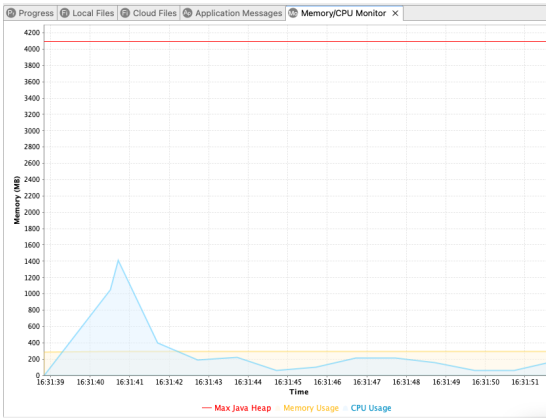


Figure 2: *OmicsBox File Export Menu*

3.4.3 View Menu

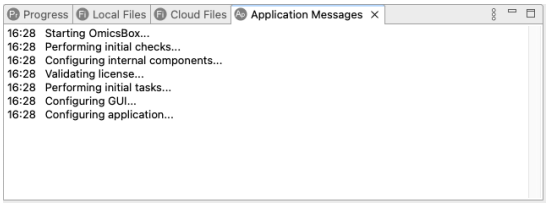
Memory/CPU Monitor

Shows the used and available memory for OmicsBox. In blue the CPU utilisation of the last minute can be seen.



Application Messages

Shows general application message as well as summary information of specific job executions.



Welcome Message

A window which provides information about application updates and new features.

Progress

This tab provides process information of any job execution in OmicsBox. Each job can be cancelled as well as a more detailed message tab can be opened.

File Manager

Cloud Usage

The Cloud Usage provides information about the number of consumed/ recharged Cloud Units or processed sequences and success jobs.

The Excel icon on the top right in Figure 1 allows exporting the table in csv format.

The Filter option allows to limit the table to a specific rage of dates.

The following features that run on the BioBam Bioinformatics Cloud Platform are:

- **Functional Analysis:** CloudIPS **(paid)**, CloudBlast **(paid)**, GO Mapping, EggNOG, Annotation
- **Transcriptomics:** Trinity*, BUSCO, TransDecoder, CD-HIT, STAR*, BWA*, RSEM*, EdgeR, maSigPro, NOISeq
- **Metagenomics:** Kraken, MEGAHIT, meta-SPAdes, FragGeneScan, Prodigal, PfamScan, EggNOG Mapper
- **Genome Analysis:** Flye,* ABySS*, SPAdes*, Pilon*, BWA*, RepeatMasker, Augustus, Glimmer
- **Apps:** CPAT, PSORTb, MLST
- These algorithms make use of free cloud computation resources. This is an introductory offer and may change in a future release depending on the overall resource consumption of these features.

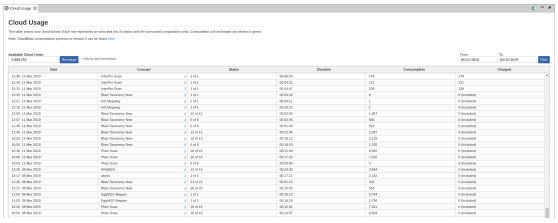


Figure 1: *Cloud Usage*

3.4.4 Help Menu

At the Help Menu, you can find this Manual, OmicsBox papers and information of the authors. In case of a bug or a malfunction of OmicsBox you can save the log file and send it to support@biobam.com or via the priority support.

- App Manager: This option allows you to install/ uninstall Apps available on OmicsBox website https://www.biobam.com/omicsbox-apps/ .
- Send Support Mail: Send an email to support with the log file already attached.
- Save Log to File.
- Startup Announcement
- Feedback
- User Manual: Opens a link with the user manual.
- Download Example Data: This option allows to download example data for each module to your computer.
- Account Information: Provides the information of the user account (available modules, subscription type, etc).
- About OmicsBox: Provides information of OmicsBox, Java and the computer where OmicsBox is installed

3.5 General Tools

3.5.1 General Tools

The General Tools section provides generic bioinformatics tools for data conversations or visualizations.

- Tag Statistics: Summarizes the project tags in a bar chart.
- Venn Diagram: Shows all possible logical relations between a finite collection of different sets.
- FastQ Tools
- FastQ Quality Check: Performs quality control checks of FastQ files.
- FastQ Preprocessing: Filter contamination sequences and adapters to obtain high-quality FastQ files.
- Fasta Tools
- Filter Fasta by Length
- Bam Tools
- Convert SAM/BAM to FastQ

3.5.2 FASTA Tools

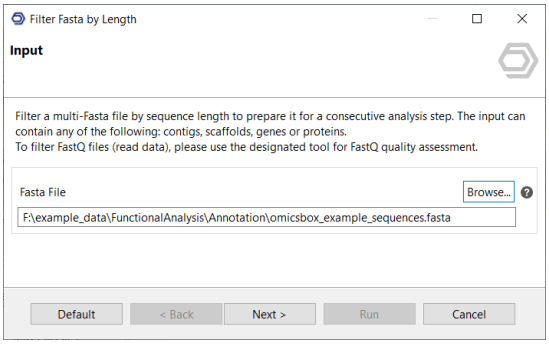
Filter Fasta by Length

Filter a multi-Fasta file by sequence length to prepare it for a consecutive analysis step. The input can contain any of the following: contigs, scaffolds, genes, or proteins.

To filter FastQ files (read data), please use the designated tool for FastQ quality assessment.

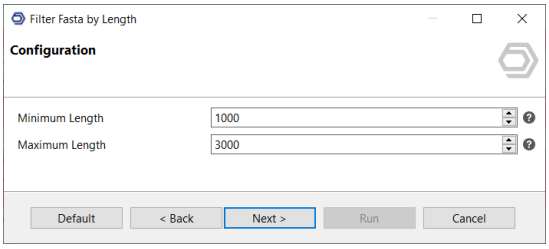
Input File

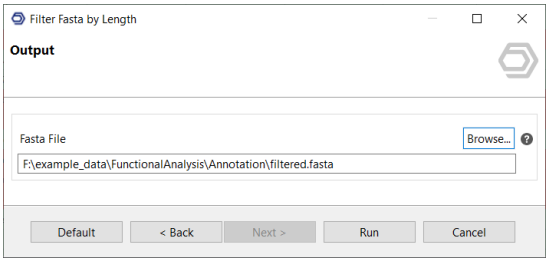
Select a file containing contigs, scaffolds, genes or proteins



Parameters

- Minimum Length: Sequences shorter than this will be excluded for the results file
- Maximum Length: Sequences longer than this will be excluded for the results file





3.5.3 FASTQ Tools

FASTQ Tools

OmicsBox provides the following tools to manipulate FASTQ files:

- FastQC Quality Check
- Fastq Preprocessing with Trimmomatic
- Long Read Quality Assessment with LongQC
- Merge FastQ and FastA Files
- Barcode Splitter
- Demultiplexing with Cutadapt

FastQC Quality Check

INTRODUCTION

This tool provides an easy way to perform a quality control check on sequence data coming from high throughput sequencing pipelines. The analysis is performed by nine modules which provide a quick overview of whether the data looks good and there are no problems or biases which may affect downstream analysis. Results and evaluations are returned in the form of charts and tables.

This tool is based on the popular **FastQC software**. Please cite FastQC as:
Andrews S (2010)."FastQC: a quality control tool for high throughput sequence data". Available online at:
http://www.bioinformatics.babraham.ac.uk/projects/fastqc

***Figure 1**: FASTQ Quality Check Interface*

RUN FASTQ QUALITY CHECK

This functionality can be found under **General Tools → FASTQ Tools → FASTQ Quality Check**. The wizard allows to select input files and adjust analysis parameters (Figure 2 and Figure 3).

Input

- **Raw Sequence Data:**Select the files containing the sequence data. These files are assumed to be in FASTQ format (or compressed in gzip format).

***Figure 2**: FASTQ Quality Check Input Page*

Configuration

- **Additional Adapter Sequences:**This option allows specifying a file that contains the list of adapter sequences that will be explicitly searched against the library. The file must contain sets of named adapters in the form of "Name Sequence". If this option is not set, OmicsBox searches for the following adapter sequences:

Default adapter sequences

- Illumina Universal Adapter: AGATCGGAAGAG
- Illumina Small RNA 3' Adapter: TGGAATTCTCGG
- Illumina Small RNA 5' Adapter: GATCGTCGGACT
- Nextera Transposase Sequence: CTGTCTCTTATA
- SOLID Small RNA Adapter: CGCCTTGGCCGT
- **Additional Contaminant Sequences:** This option allows specifying a file that contains the list of contaminants to screen over-represented sequences against. The file must contain sets of named contaminants in the form of "Name Sequence". If this option is not set, OmicsBox searches for a list of common contaminant sequences.
- **Chart Read Length Binning:**Enable grouping of bases for reads. If not, reports will show data for every base in the read.

Disabling this option on long reads (> 50 bp) can cause that the plots look very small.

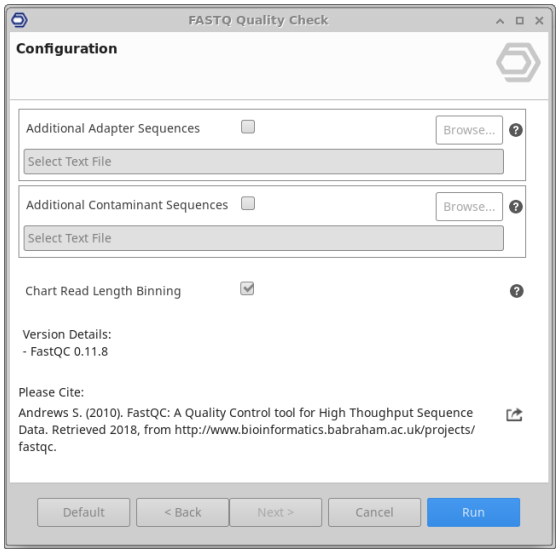


***Figure 3**: FASTQ Quality Check Configuration Page*

RESULTS

Once finished, a new tab is opened containing simple composition statistics of each analyzed file (Figure 4). Each row corresponds to an input file, and columns show the following information:

- Name: The name of the file which was analyzed.
- File type: This shows whether the file appeared to contain actual base calls or colorspace data which had to be converted to base calls.
- Encoding: This shows the ASCII encoding of quality values was detected in this file.
- Total Sequences: The total number of read sequences processed.
- Poor quality reads: Sequences flagged as poor quality reads.
- Sequence Length: Provides the length of the shortest and longest sequence in the set. If all sequences are the same length only one value is reported.
- %GC: The overall %GC of all bases in all sequences.

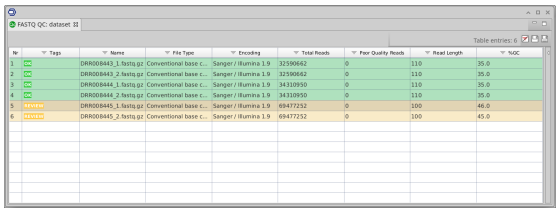


***Figure 4**: FASTQ Quality Check Project*

Furthermore, a result page will show a summary of the "FASTQ Quality Check" results (Figure 5). This page provides a quick evaluation of whether the results of each module seem entirely normal (pass), slightly abnormal (warning), or very unusual (fail).

Note that these evaluations must be taken in the context of what is expected from each library. For example, some experiments may be expected to produce libraries that are biased in particular ways. Therefore, the summary evaluations should be treated as pointers that guide the preprocessing of the libraries.

The result summary can be generated via **Side Panel → Summary Report.**Additionally, the report of each file can be opened by clicking on the button of the column "Report".

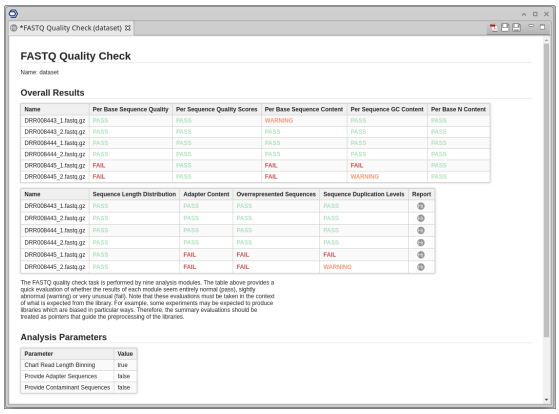


***Figure 5:**FASTQ Quality Check Report*

The results of each module for each file can be accessed as follows:

- To open the summary report of each file, right-click on a row and click on **Show report**. A new report is opened containing a summary of the statistics and results for the selected file (Figure 6).
- To open the result of each module for a file, right-click on a row and go to the **Show Statistics**submenu**.**These results also can be accessed by clicking on the buttons of the "Details" column of the results table.

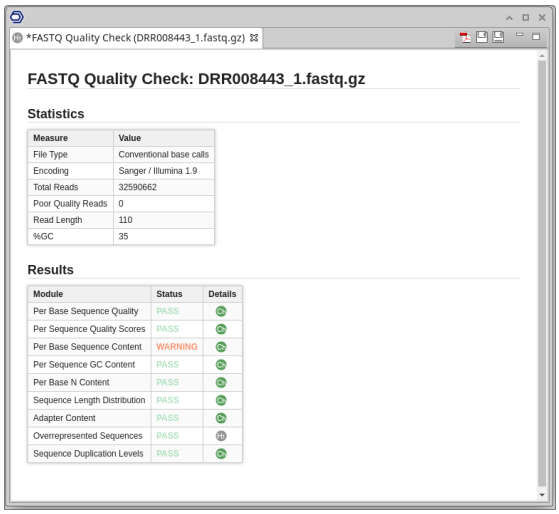


Figure 6:*Report of a FASTQ file*

Per Base Sequence Quality

This chart shows an overview of the range of quality values across all bases at each position in the FASTQ file (Figure 7).

For each position (x-axis), a box and whisker type plot is drawn:

- The central black line is the median value.
- The yellow box represents the interquartile range (25-75%).
- The upper and lower whiskers represent the 10% and 90% points.
- The blue line represents the mean quality.

The y-axis shows the quality scores. The background of the graph divides the y-axis into very good quality calls (green), calls of reasonable quality (orange), and calls of poor quality (red).

The title of the graph will describe the encoding that the input files used.

A **WARNING** is issued if the lower quartile for any base is less than 10, or if the median for any base is less than 25. This module raises a **FAIL**if the lower quartile for any base is less than 5 or if the median for any base is less than 20.

The most common reason for warnings and failures is a general degradation of quality over the duration of long runs. If the quality of the library falls to a low level then the most common procedure is to perform a quality trimming to truncate reads based on their average quality.

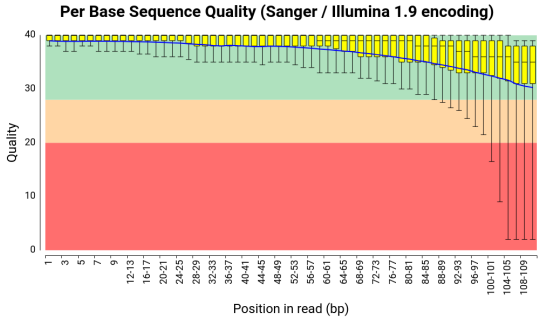


Figure 7:*Per Base Sequence Quality Chart*

Per Sequence Quality Scores

This chart displays the number of reads that have the same mean sequence quality (Figure 8). It allows seeing if a subset of your sequences has universally low-quality values.

A **WARNING**is raised if the most frequently observed mean quality is below 27 (0.2% error rate). A **FAIL**is raised if the most frequently observed mean quality is below 20 (1% error rate).

If a significant proportion of the reads in a run have overall low quality then this indicates some kind of systematic problem. This may be alleviated through quality trimming.

Figure 8:*Per Sequence Quality Scores Chart*

Per Base Sequence Content

This chart plots out the proportion of each base position in a FASTQ file for which each of the four normal DNA bases has been called (Figure 9). In a random library, it is expected that there would be little to no difference between the different bases of the sequence reads, so the lines in this plot should run parallel with each other.

A **WARNING**is issued if the difference between A and T, or G and C is greater than 10% in any position. A **FAIL**is raised if the difference between A and T, or G and C is greater than 20% in any position.

The common reasons for warnings and failures are:

- Overrepresented sequences (such as adapter dimers or rRNA in a sample).
- Biased fragmentation (nearly all RNA-Seq libraries will fail this module because of this bias).
- Biased composition libraries.
- If the library has been adapter trimmed.

Figure 9:*Per Base Sequence Content Chart*

Per Sequence GC Content

This module measures the GC content across the whole length of each sequence read in a file and compares it to a modeled normal distribution of GC content (Figure 10). Since the GC content of the genome is not known, the modal GC content is calculated from the observed data and used to build a reference distribution.

A **WARNING** is raised if the sum of the deviations from the normal distribution represents more than 15% of the reads. A **FAIL** indicates that the sum of the deviations from the normal distribution represents more than 30% of the reads.

Warnings and failures indicate a problem with the library (e.g. specific contaminant). An unusually shaped distribution could indicate a contaminated library. A normal distribution that is shifted indicates some systematic bias which is independent of base position.

If there is a systematic bias that creates a shifted normal distribution then this won't be flagged as an error by the module since it doesn't know what the genome's GC content should be.

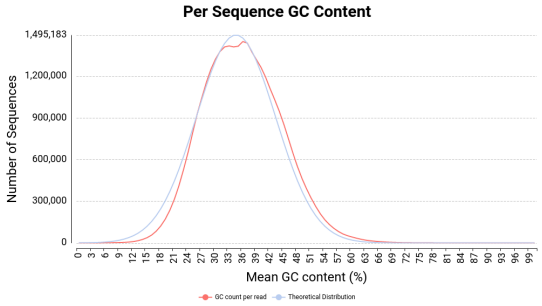


Figure 10: *Per Sequence GC Content Chart*

Per Base N Content

This module plots out the percentage of base calls at each position for which an N was called (Figure 11). N replaces a conventional base call when the sequence is unable to make a base call with sufficient confidence.

A **WARNING**is raised if any position shows an N content of >5%. A **FAIL**is raised if any position shows an N content of >20%.

It is not unusual to see a very low proportion of Ns appearing in a sequence (especially near the end of a sequence). However, if this proportion rises above a few percent it suggests that the analysis pipeline was unable to interpret the data well enough to make valid base calls.

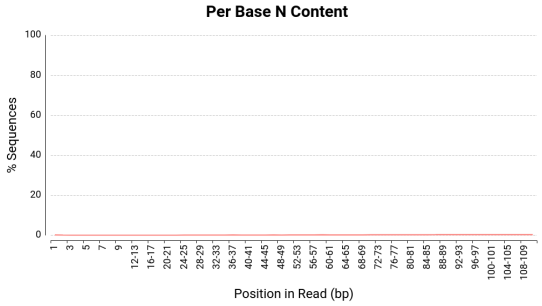


Figure 11:*Per Base N Content Chart*

Sequence Length Distribution

This chart shows the distribution of fragment sizes in the file which was analyzed (Figure 12). In many cases, this will produce a simple graph showing a peak only at one size, but for variable-length FASTQ files, this will show the relative amounts of each different size of sequence fragment.

A **WARNING** is raised if all sequences are not the same length. A **FAIL** is raised if any of the sequences have zero length.

For some sequencing platforms, it is entirely normal to have different read lengths so warnings here can be ignored.

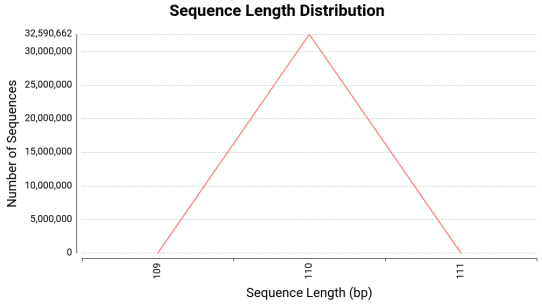


Figure 12:*Sequence Length Distribution Chart*

Adapter Content

This chart shows a cumulative percentage of the proportion of the library in which each of the adapter sequences at each position has been detected (Figure 13). Once a sequence has been detected in a read, it is counted as being present right through to the end of the read so the percentage increases as the read length continue.

A **WARNING**is issued if any sequence is present in more than 5% of all reads. A **FAIL** is issued if any sequence is present in more than 10% of all reads.

This module indicates if the sequences will need to be trimmed for adapters before proceeding with any downstream analysis.

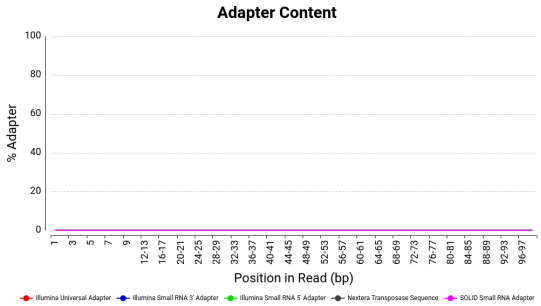


Figure 13:*Adapter Content Chart*

Overrepresented Sequences

This module lists all of the sequences which make up more than 0.1% of the total (Figure 14). To conserve memory only sequences that appear in the first 100,000 sequences are tracked to the end of the file. Therefore, it is possible that a sequence that is overrepresented but doesn't appear at the start of the file for some reason could be missed by this module.

For each overrepresented sequence, the program will look for matches in a database of common contaminants and will report the best hit that it finds. Hits must be at least 20 bp in length and have no more than 1 mismatch.

A **WARNING** is issued if any sequence is found to represent more than 0.1% of the total. A **FAIL** is issued if any sequence is found to represent more than 1% of the total.

This module will often be triggered when used to analyze small RNA libraries where sequences are not subjected to random fragmentation, and the same sequence may naturally be present in a significant proportion of the library.

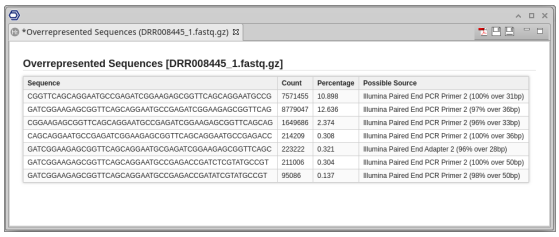


Figure 14:*Overrepresented Sequences Table*

Sequence Duplication Levels

This module counts the degree of duplication for every sequence in a library and creates a graph showing the relative number of sequences with different degrees of duplication (Figure 15). The chart shows the proportion of the library which is made up of sequences in each of the different duplication level bins.

There are two lines on the plot:

- The blue line takes the full sequence set and shows how its duplication levels are distributed.
- The red line displays the proportions of the sequences that are deduplicated which come from different duplication levels in the original data.

The module also calculates an expected overall loss of sequences when the library is deduplicated. This is shown at the top of the plot and gives a reasonable impression of the potential overall level of loss.

A **WARNING**is raised if non-unique sequences make up more than 20% of the total. A **FAIL** is raised if non-unique sequences make up more than 50% of the total.

In general, there are two potential types of duplicates in a library, technical duplicates arising from PCR artifacts, or biological duplicates which are natural collisions where different copies of exactly the same sequence are randomly selected.

In RNA-Seq libraries, sequences from different transcripts will be present at wildly different levels in the starting population. In order to be able to observe lowly expressed transcripts, it is therefore common to greatly over-sequence high expressed transcripts, and this will potentially create large sets of duplicates. This will result in high overall duplication in this test, and will often produce peaks in the higher duplication bins.

To reduce the memory requirements only the first 100000 sequences of each file are analyzed.

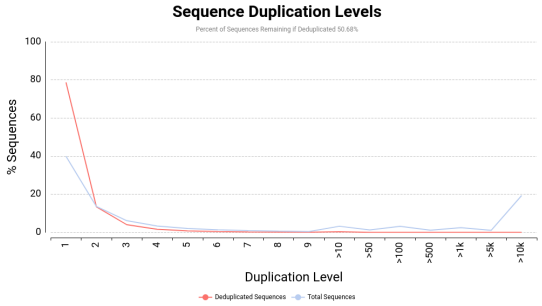


Figure 15:*Sequence Duplication Levels Chart*

Fastq Preprocessing with Trimmomatic

INTRODUCTION

As Next-Generation Sequencing (NGS) technology is used more broadly in scientific applications and research, sequencing data quality control is becoming more important. Experiments and sequencing processes always introduce errors and biases, so downstream sequence analyses are compromised by low-quality sequences, sequence artifacts, and sequence contamination. These problems eventually lead to erroneous conclusions in processes such as assembly and alignment, so a preprocessing step is necessary to produce better analysis results.

Preprocessing FASTQ files in OmicsBox consists of removing adapters and contamination sequences, trimming low-quality bases, and filtering short and low-quality reads. Before proceeding, it is advisable to carry out a quality control check of the sequencing data within OmicsBox (FASTQ Quality Check). In this way, problems and biases can be detected, which allows to better configure the preprocessing procedure.

The FASTQ Preprocessing tool uses the well-known preprocessing software **Trimmomatic**. Trimmomatic is a fast, multithreaded command-line tool that can be used to trim and crop sequencing data as well as to remove adapters. For further information visit the Trimmomatic web page.

Please, cite Trimmomatic as:

Bolger AM, Lohse M, Usadel B (2014). "Trimmomatic: A flexible trimmer for Illumina Sequence Data". Bioinformatics, btu170.

Adapter Removal

This step is used to find and remove adapters and contaminant sequences. The application uses two approaches to detect technical sequences within the reads:

- **Simple mode:** The simple mode approach works by finding an approximate match between the read and supplied technical sequences. These sequences can be detected in any location or orientation within the reads but require a minimum overlap between the read to prevent false positives. However, short partial adapter sequences cannot achieve this minimum overlap requirement, so they are not detectable.
- **Palindrome mode:**The palindrome mode approach is specifically aimed at detecting the common "adapter read-through" scenario whereby the sequenced DNA fragment is shorter than the read length. When "read-through" happens, both reads in a pair will consist of an equal number of valid bases, followed by contaminating sequences from the "opposite" adapters. Furthermore, the valid sequence within the two reads will be reverse complements. This mode can only be used with paired-end data but has considerable advantages in sensitivity and specificity over the "simple" mode.

Trimming

This step is used to remove low-quality bases from the reads. The application offers four trimming alternatives:

- **Sliding window trimming:** The sliding window approach works by scanning from the 5' end of the read and removes the remaining 3' end of the read when the average quality of a group of bases drops below a specified threshold.
- **Adaptive quality trimming:**The adaptive quality trim approach, also known as "Maximum information quality trimming", balances the benefits of retaining longer reads against the cost of retaining bases with errors.
- **Quality trimming:** The quality trimming approach removes low-quality bases from the beginning or the end of the read. As long as a base has a value below this threshold, the base is removed and the next base will be investigated.
- **Length trimming:** The length trimming approach removes a specified number of bases regardless of quality from the beginning or the end of the read.

Filtering

This step is used to filter out reads:

- **Filter by quality:**Remove reads that fall below the specified average quality.
- **Filter by length:**Remove reads that fall below the specified minimum length.

RUN FASTQ PREPROCESSING

This functionality can be found under **General Tools → FASTQ Tools → FASTQ Preprocessing.**The input data and the different preprocessing steps can be configured using the wizard (Figure 1, Figure 2, Figure 3, Figure 4, Figure 5 and Figure 6).

Input

- **Sequencing Data:** Choose the type of data to be preprocessed: single-end or paired-end reads. Note that if paired-end is selected, two files per sample are required.
- **Input Reads:**Provide the files containing sequencing reads. These files are assumed to be in FASTQ format.
- **Paired-end configuration:**In the case of paired-end reads, the pattern to distinguish upstream files from downstream files is required. The provided patterns are searched right before the extension, and the start of the name should be the same for both files of each sample.
- Upstream Files Pattern: Establish the pattern to recognize upstream FASTQ files.
- Downstream Files Pattern: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is named SRR037717_1.fastq and the downstream one SRR037717_2.fastq, you should establish "_1" as the upstream pattern and "_2" as the downstream pattern.

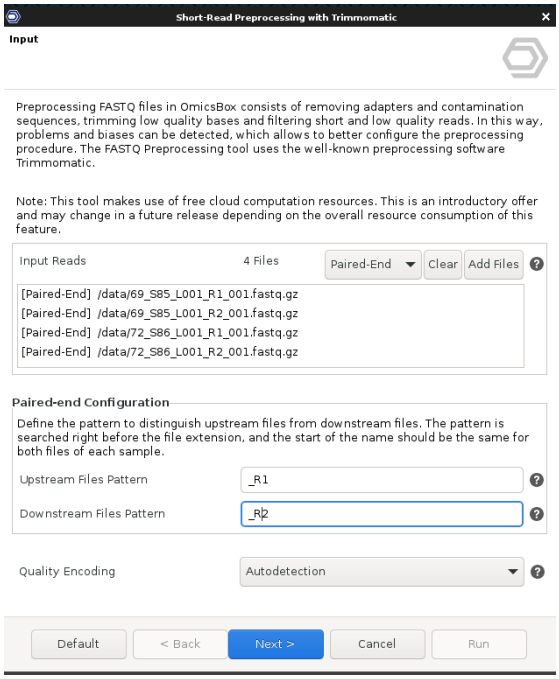


Figure 1:*Input Data Page*

Adapter Removal Configuration

- **Remove Adapters:**Enable the adapter removal step.
- **Use Adapters From:** Choose between using the default adapter sequences provided by Trimmomatic, or providing custom adapter sequences.
- Default Adapter Sequences: By default, the application provides adapter sequences for TruSeq2 (GAII machines), TruSeq3 (HiSeq and MiSeq machines), and Nextera, for both single-end and paired-end data.

If you use the FASTQ Quality Check tool, the "Adapter Content" and "Overrepresented Sequences" modules can help to choose which default adapter sequences are best suited for your data. "Illumina Single-End" or "Illumina Paired-End" sequences indicate single-end or paired-end TruSeq2 libraries. "TruSeq Universal Adapter" or "TruSeq

Adapter, Index..." sequences indicate TruSeq-3 libraries. Note that these sequences have not been extensively tested, so other sequences may work better for a given dataset.

- Custom Adapter Sequences: Specifies a FASTA file containing all the adapters and contaminant sequences to be removed. The names of sequences determine how they are used, especially for paired-end data.
- For the "palindrome" mode, matched pairs of adapter sequences must be supplied. The sequence names should start with "Prefix", and end in "/1" for the forward adapter and "/2" for the reverse adapter. The part of the name between "Prefix" and "/1" or "/2" must match exactly within each pair.

> PrefixPE/1
> TACACTCTTTCCCTACACGACGCTCTTCCGATCT
> PrefixPE/2
> GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT

- For "simple" mode, sequences with names ending in "/1" or "/2" will be searched only in the forward or reverse read respectively. Otherwise, sequences will be searched in both the forward and reverse read.

> Adapter_a
> AGATCGGAAGAGCTCGTATGCCGTCTTCTGCTTG
> Adapter_b
> AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT

- **Seed Mismatches:**Set the maximum mismatch count which allows performing a full match.
- **Simple Clip Threshold:**Establish how accurate the match between the adapter sequence must be against a read. This option is only considered for the simple mode.
- **Palindrome Clip Threshold:**Establish how accurate the match between the two "adapter-ligated" reads must be. This option is only considered for the palindrome mode.
- **Minimum Adapter Length:**Set a minimum length for adapters to be detected. This option is only considered for the palindrome mode.
- **Keep Both Reads:** Deleting adapters after read-through detection (palindrome mode) causes the reverse read to contain the same information as the forward read, although in reverse complement. This option allows retaining the reverse read. Otherwise, the reverse read will be discarded.

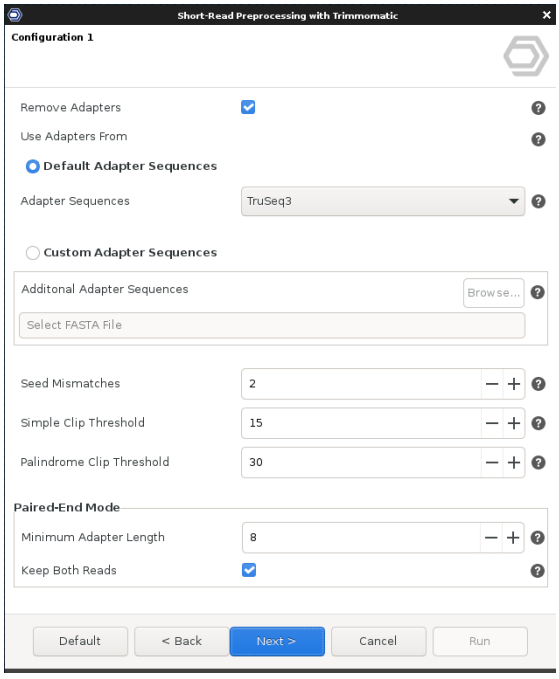


Figure 2:*Adapter Removal Page*

Trimming Configuration

There are two main categories of trimming, each comprising two distinct strategies. Please note that these categories are independently selected and executed sequentially, according to their order of appearance.

- **Edge trimming**: Trims sequences from the beginning and/or end.
- **Quality Trimming:**
  - Quality Trimming 3': Remove bases from the beginning of the sequence.
  - Trimming threshold 3': Establish a minimum quality required to keep bases from the beginning.
  - Quality Trimming 5': Remove bases from the end of the sequence.
  - Trimming threshold 5': Establish a minimum quality required to keep bases from the end.
  - **Length Trimming:**
  - Trimming from 3': Remove bases from the beginning of the sequence.
  - Trimming threshold 3': number of bases to be removed from the start of the read.
  - Trimming from 5': Remove bases from the end of the sequence.
  - Trimming threshold 5': Number of bases to be kept from the start of the read so that it has maximally the specified length after this step.
  - **Quality trimming**: Trims sequences based on their quality and length.
- **Sliding Window Trimming:**
  - Window Size: Set the number of bases that the window has to span to average the quality.
  - Required Quality: Set the average quality required to retain bases.
  - **Adaptive Quality Trimming:**
  - Target Length: Set the minimum read length which is likely to allow the location of the read within the target sequence.
  - Strictness: This value establishes the balance between preserving as much read length as possible versus removal of incorrect bases. It should be set between 0 and 1. A low value favors longer reads, while a high value favors read correctness.

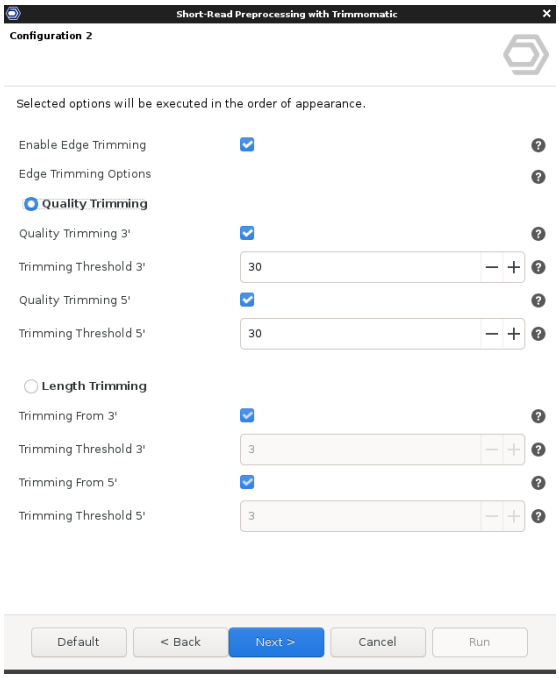


Figure 3:*Edge Trimming Page*

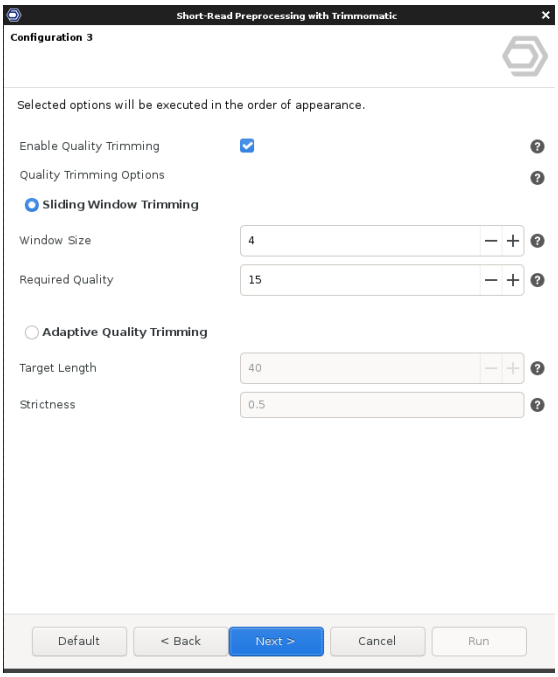


Figure 4:*Quality Trimming Page*

Filtering Configuration

- **Filter By Quality:** Enable the filtering by quality step.
- **Average Quality:** Minimum average quality of reads to be kept.
- **Filter By Length:** Enable the filtering by length step.
- **Minimum Length:**Minimum length of reads to be kept.

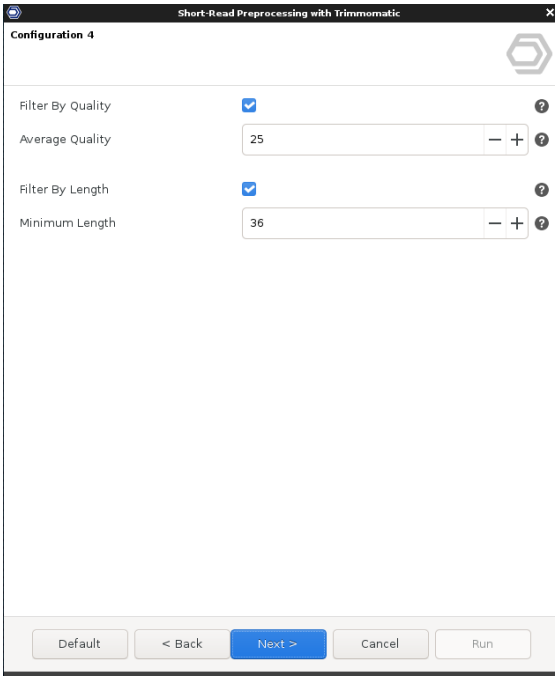


Figure 5:*Filtering Page*

Output

- **Output Prefix:**Define a prefix to establish the name of output files. The prefix will be added before each original file name.

- **Output Reads:**Select a destination folder to save the preprocessed FASTQ files.
- **Unpaired Reads:** When preprocessing paired-end data, some read pairs can lose a member as a result of trimming and filtering. Select a destination folder to save the FASTQ files containing unpaired reads. These files contain the word "unpaired" in their file names.

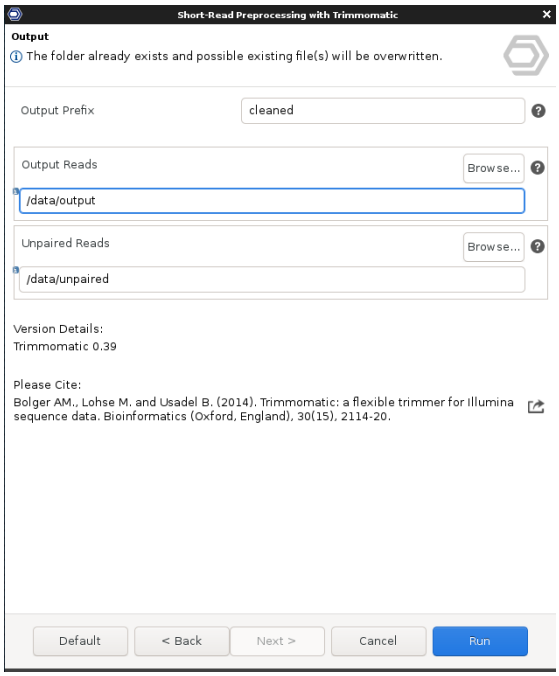


Figure 6:*Output Page*

RESULTS

Once finished, output files containing preprocessed reads are stored in the "Output Reads" folder set in the wizard (Figure 7). Files are generated in compressed format (fastq.gz).

For single-end data, one output file per input file is generated. For paired-end data, four output files per input sample (2 FASTQ files) are generated, two that contain upstream and downstream paired reads and two that contain upstream and downstream unpaired reads. The name of each output file begins with the provided prefix and continues with the original name of the file. Files with unpaired reads contain the word "unpaired" in their name so that they can be distinguished from those that contain paired reads. These files are placed in the "Unpaired Reads" folder.

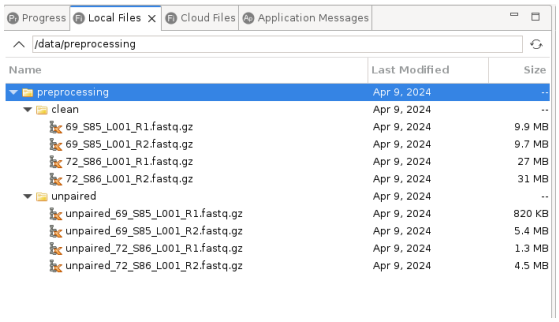


Figure 7:*FASTQ Preprocessing Output Folder*

Furthermore, a result page will show a summary of the "FASTQ Preprocessing" results (Figure 8). This page provides a table that shows how many reads have survived and how many have been dropped during the analysis.

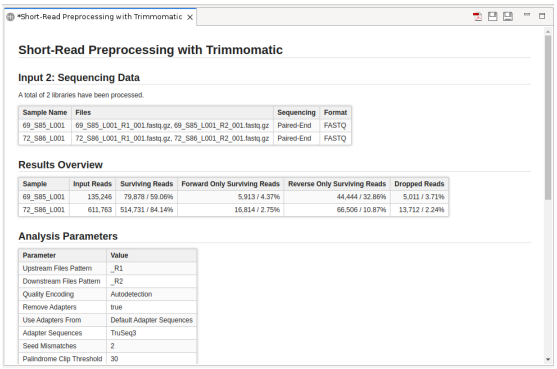


Figure 8: *FASTQ Preprocessing Report*

Long Read Quality Assessment with LongQC

Introduction

LongQC is a computationally efficient, platform-independent QC tool to spot issues before a full analysis. The tool visualizes statistics designed for erroneous long read data to highlight potential problems originated from the biological samples as well as those introduced at the sequencing stage. It supports major TGS file formats. LongQC relies on k-mer based internal overlaps and skips alignment; therefore, it operates efficiently without reference genomes.

Please cite LongQC as:

LongQC: A Quality Control Tool for Third Generation Sequencing Long Read Data.
Yoshinori Fukasawa, Luca Ermini, Hai Wang, Karen Carty, Ming-Sin Cheung.
G3: Genes, Genomes, Genetics, 10(4): 1193-1196, 2020

Quality Assessment with LongQC

LongQC can be found in the General Tools Module of OmicsBox under **FASTQ tools → Long Read Quality Assessment with LongQC**. The wizard allows the selection of several sequencing files to be assessed, the possibility to save an optional output result and some analysis parameters (Figure 1).

Input

LongQC analyses TGS reads files from either Pacbio or Oxford Nanopore sequencing technologies. These files should be formatted as the following standard types:

- **Fasta file**: Plain text file containing all the reads sequences obtained in the sequencing process.
- **FASTQ file**: Plain text file that store both the nucleotide sequences of the reads and their per base quality score. It has become the *de facto* standard for storing the output of the sequencing process.
- **PacBio BAM file**: Binary and compressed container format for storing PacBio sequencing reads. It is based on the specifications for BAM/SAM, although it does not contain information about any of alignment. More information about this format can be found in PacBio documentation.

Take into account that LongQC uses per base sequencing quality scores in some of its analyses. If these values are not provided, part of the analyses will not be performed.

Configuration

There are several parameters that can be tuned to perform a more accurate analysis. These parameters should be adjusted according the type of the input reads:

- **Sequencing Technology**: Applies the specific configuration of internal parameters that best suit with each sequencing technology. This parameter is particularly relevant in adapters analysis since it selects the concrete sequences of the adapters.
- **Transcript Mode**: This parameter enables a concrete LongQC configuration that fits the specific features of transcript (RNA, cDNA) data.
- **Short Mode**: This parameter activate a highly sensitive setting for very short and erroneous reads.

LongQC works even if the selection of the parameters does not fit the input. However, this could lead to incorrect results or warnings.

Output

LongQC provides an optional FASTQ file that contains the trimmed reads after the adapter detection analysis.

The parameters that control the output are the following:

- **Save trimmed**: Activate to save the trimmed reads. When active, destination folder selection is available.
- **Directory to save the trimmed reads**: Where the destination folder can be selected.

Results

LongQC provides the following outputs:

- **Table** with the most relevant statistics about each sample (Figure 2).
- **Report** with a information of general statistics, warnings/errors, GC composition and the Adapter detection results (Figure 3).
- **Charts**:
- Adapters detection (Line Plot/Histogram).
- Sequences Composition:
    - GC content plot (Line Plot/Histogram).
    - Masked bases plot (Line Plot/Histogram).
    - Reads Coverage:
    - Binned coverages plot (Line Plot/Histogram).
    - Coverage over length plot (Line Plot/Box Plot).
    - Reads Length (Line Plot/Histogram).
    - Reads Quality:
    - Quality over length plot (Line Plot/Box Plot).
    - Quality over coverage plot (Box Plot).

Box Plots and Histograms are only available for representing a single sample in each chart. Line Plots allow comparing several samples in the same chart.

- **File**:
- Trimmed reads (optional).

Table with general statistics

It contains the next columns (Figure 2):

- **Tags**: It indicates with a colored label if the analysis of a sample has returned a warning, an error, or none of them (correct). These warnings and errors refer to problems detected by LongQC in the samples. Minor issues are considered "warnings", while significant issues should lead to an "error".
- **Sample**: Name of the file without extensions.
- **Total Number of bp**: Number of the total base pairs from every read sequence in the file.
- **Mean Read Length**: Lengths average from all reads in the file.
- **N50**: Statistic commonly used to assess the quality of a genome assembly. Here, it represent a length-weighted median.
- **Longest Read**: Length (in number of bp) of the longest read in the sample.
- **Number of reads**: Total number of reads sequences in the file.
- **% Non-sense Reads**: Fraction (in percentage) of the reads having a coverage value of 0.

Coverage is calculated by mapping all reads between them. Therefore, non-sense reads are unique reads that cannot be mapped onto any other sequences in the same file.

- **% Q>7 Bases**: Fraction (in percentage) of bases having a quality value of 7 or higher. This column is optional since PacBio Sequel technologies do not provide any per-base quality score.

A quality value of 7 represents an error rate of 20%.

- **Warnings**: Brief description of all warnings and errors found in the sample.

LongQC Report

The report contains the next sections:

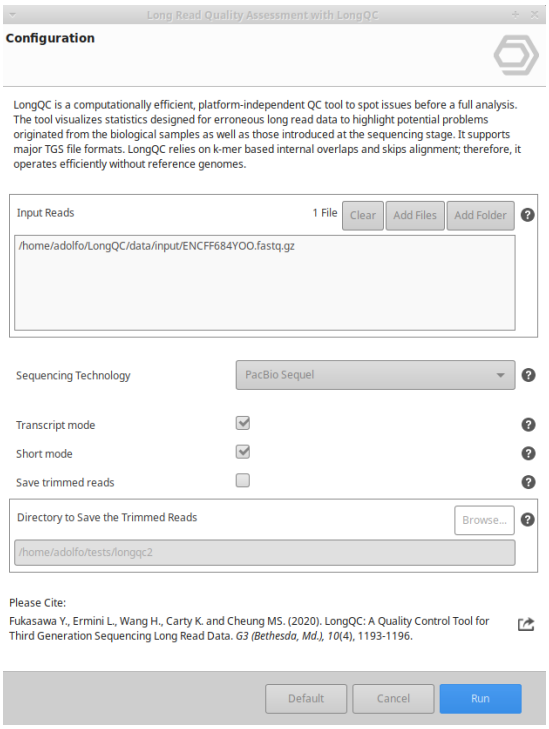
- **General Statistics**: Some of the statistics showed in the table (Figure 2).
- **GC Content Statistics**: Mean and Standard deviation of the GC content. The sample's mean should be close to the mean GC content of the organism genome.
- **Warnings/Errors**: Table with all warnings and errors detected in all samples. The table only appears if any kind of issue has been detected.
- **Adapter Statistics**: This table is shown if adapters sequences have been detected in, at least, one sample. It contains the following columns:
- **Number of trimmed reads**: The number of reads having adapter like (75% or higher identity) sequences in either terminals. If this is unexpectedly low and trimming was not conducted, it infers that adapter ligation step had some problems.
- **Max seq identity**: Maximum value of identity between adapter sequence and sequences. This value should be quite high (90%) if adapter still exists in a dataset.
- **Average trimmed length**: The average end position of aligned sequences. This should be consistent with the kit description and peak in the flanking region analysis plots.
- **Parameters**: Execution parameters of the analysis.

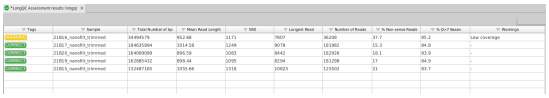
Charts

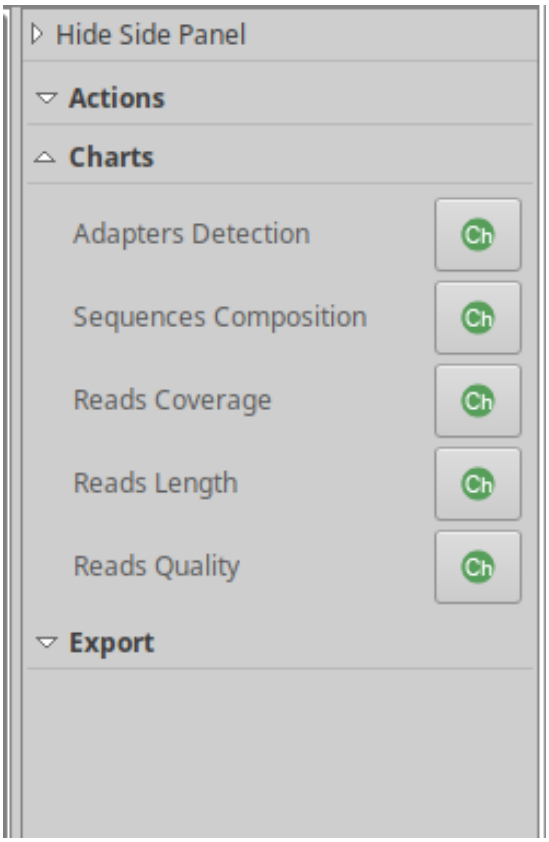
LongQC charts can be accessed through the side panel action buttons (Figure 3). All buttons display an specific wizard where several plotting options can be selected.

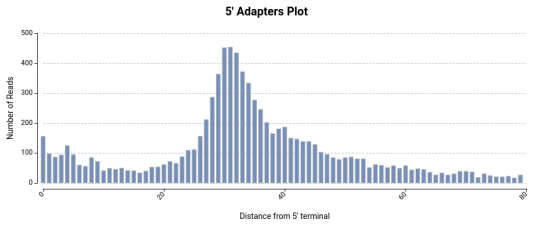
All charts are grouped in the following sections:

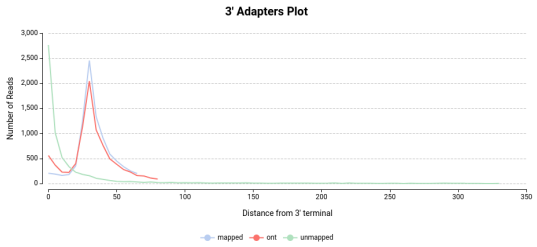
- **Adapters Detection**: These charts represent the count of specific fragment sequences that match adapter sequences and their distance to 5' and 3' terminals. If the tool detects any adapter, it should show a peak distinct from 0..
- There are two options to display the adapters detection results:
    - Histogram: Very useful since it allows to change the size of the bins. Only available for one sample (Figure 4).
    - Line Plot: Suitable for comparing two or more samples, bin sizes are fixed (Figure 5).
    - **Sequences Composition**: There are two types of charts regarding the sequence composition of the reads:
- **GC Content Plot**: It displays the GC content distribution of the samples. It should show one peak if the quality of the sample is correct. This peak should be near the average GC content of the studied genome. The presence of more than one peak could imply the existence of contaminant sequences from other organisms. However, samples from metagenomics analysis could show more than one peak. The options available to plot the GC content are the Histogram and the Line Plot (Figure 6).
- **Masked Bases Plot**: This chart shows the distribution of the per-read sequence complexity score computed by DUST algorithm. The presence of high fractions of masked bases could point to some problems in the sequencing process. The options available, in this case, are the Histogram and the Line Plot (Figure 7).
- **Reads Coverage**: These charts represent, in two different ways, the distribution of calculated coverage on every sample. The two groups of plots within this section are:
- **Binned Coverage Plot**: It displays the binned coverage distribution for all samples. It should show only one peak if the sample has no issues. However, metagenomics samples can present more than one peak and still be correct. It can be plotted as a Histogram or a Line Plot (Figure 8)**.**
- **Coverage over Length Plot**: It shows the fluctuation of coverage over the length of the reads. Significant divergences between length intervals could tell about contamination, low-quality library, or overloading in PacBio, according to the LongQC authors. Coverage over length could be plotted as:
    - Box Plot: Representing a box for each length interval. More informative and robust but only available for one sample.
    - Line Plot: Much less accurate than box plot, but it allows comparing between samples (Figure 9).
    - **Reads Length**: These charts represent the length distribution of the samples. This distribution heavily depends on the technology and the type of sequenced molecule (genomic DNA or RNA). Samples from the same experiment and organism should display highly similar distribution curves. These distributions could be represented either as a Histogram (Figure 10) or Line Plot.
    - **Reads Quality**: These charts are only available if the sample files provide any kind of per-base quality score. These scores represent the accuracy of the base calling for each base pair.
- **Quality over Length Plot**: It shows the quality fluctuation over the length of the reads. If the data is correct, quality should have a similar behavior between length intervals. Quality over length could be plotted as Box Plots (Figure 11) or Line Plots.
- **Quality over Coverage Box Plot**: This chart represents in two boxes the distribution of per-read average quality values according to their coverage value (Figure 12). Ideally, normal reads should have a better overall quality than non-sense reads. Also, the median of the non-sense reads should be in the red region. If both medians are similar, there are two possible scenarios:
    - The two medians are located in the green region: The coverage along the dataset may be low. Sequencing depth should be improved.
    - The two medians are located in the red region: There are issues with the quality of the data that could affect downstream analysis.

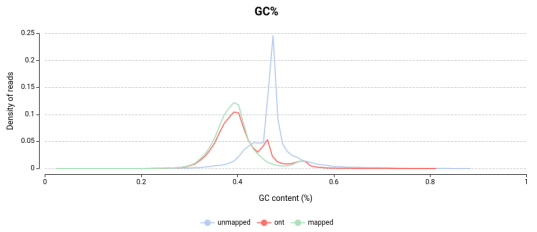


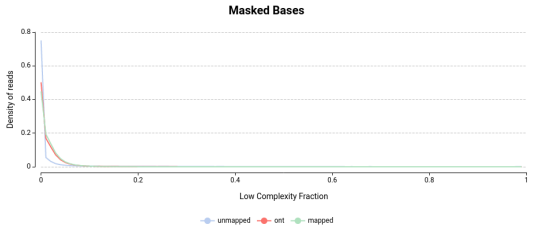


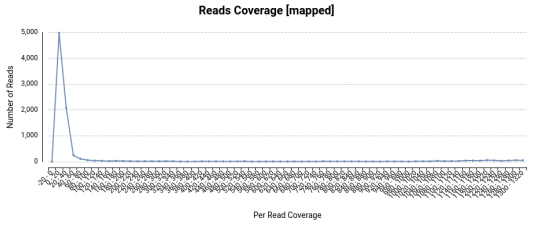


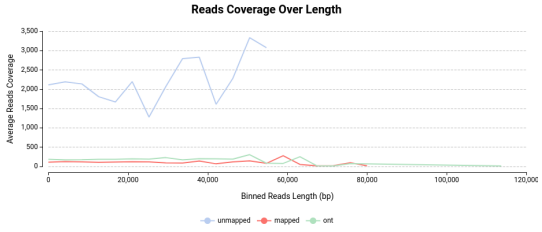


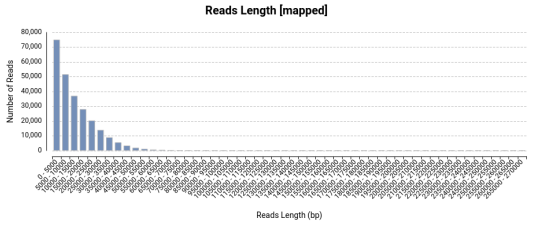


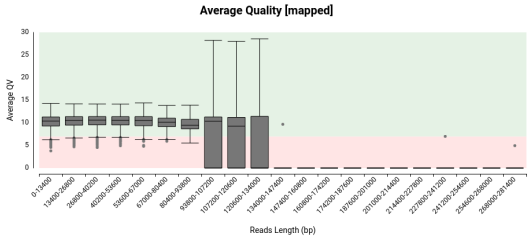


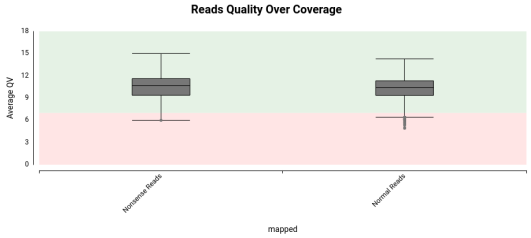












Merge FastQ and FastA Files

INTRODUCTION

This tool allows to merge multiple Fasta or FastQ files into one.

The input as well as the output files can be in compressed (.gz) format.

The tool is located in the General Module of OmicsBox under **FastQ Tools → Merge FastQ/FastA Files**.

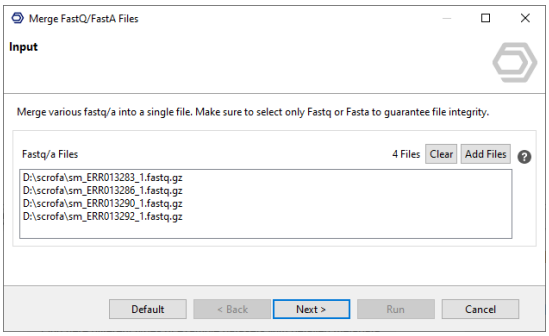
The two wizard pages allow to define the input files (fastq or fasta) as well as the output file (Figure 1 and Figure 2).

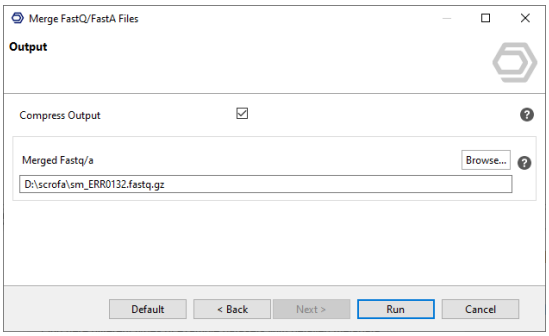
Input

FastQ or FastA formatted sequences. Please make sure that you either provide FastQ or FastA files. **Do not mix formats.**

Output

- **Compress Output**
  This option compresses the output file in .gz format.
- **Merged Fastq/a**
  The file location of the merged file.





Barcode Splitter

INTRODUCTION

Demultiplexing or barcode splitting refers to the step in processing where you would use the barcode information in order to know which sequences came from which sample after they had all been sequenced together. Barcodes refer to the unique sequences that were ligated to your each of your individual samples' genetic material before the samples got all mixed together. Depending on your sequencing facility, you may get your samples already split into individual fastq files, or they may be lumped together all in one fastq file with barcodes still attached for you to do the splitting. If this is the case, you should also have a mapping or barcode file telling you which barcodes correspond with which samples.

This tool takes FASTA/FASTQ files and splits them into several smaller files, Based on barcode matching. FastX-Toolkit is used for this task.

BARCODE SPLITTER WIZARD

Page 1 - Input

Reads- Select the FastQ/A files that contain sequences that have attached barcodes which link those sequences to the respective samples.

Barcode File - Select the mapping file that establishes the connection between each barcode and sample.

Barcode file format

Barcode files are simple text files. Each line should contain an identifier (descriptive name for the barcode), and the barcode itself (A/C/G/T), separated by a TAB character. Example:

```
#This line is a comment (starts with a 'number' sign)
BC1 GATCT
BC2 ATCGT
BC3 GTGAT
BC4 TGTCT
```

For each barcode, a new FASTQ file will be created (with the barcode's identifier as part of the file name). Sequences matching the barcode will be stored in the appropriate file.

Running the above example (assuming "mybarcodes.txt" contains the above barcodes), will create the following files:

```
/tmp/bla_BC1.txt
/tmp/bla_BC2.txt
/tmp/bla_BC3.txt
/tmp/bla_BC4.txt
/tmp/bla_unmatched.txt
```

The 'unmatched' file will contain all sequences that didn't match any barcode.

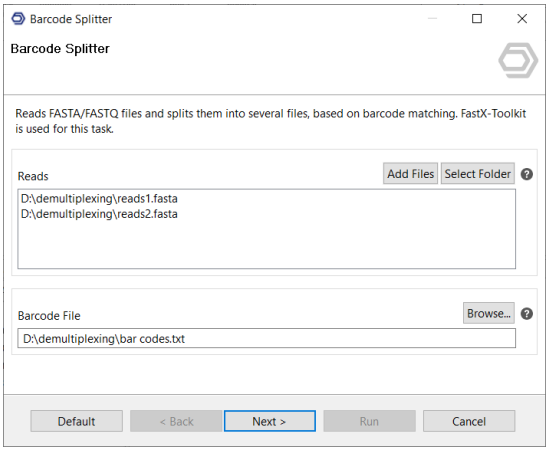


Figure 1: Wizard page 1 , Input Files

Page 2 - Configuration

Prefix - File prefix that will be added to the output files.

Suffix - File suffix that will be added to the output files.

Match Barcode - Match the barcodes at the beginning (5') or end (3') of each sequence.

Mismatches - Maximum number of allowed mismatches for barcodes.

Partial - Allow partial overlap of barcodes.

Without partial matching:

Count mismatches between the FASTA/Q sequences and the barcodes. The barcode which matched with the lowest mismatches count (providing the count is small or equal to '--mismatches N') 'gets' the sequences.

Example (using the above barcodes):
Input Sequence:
GATTTACTATGTAAAGATAGAAGGAATAAGGTGAAG

Matching at beginning of sequenecs and 1 mismatch:
GATTTACTATGTAAAGATAGAAGGAATAAGGTGAAG
GATCT (1 mismatch, BC1)
ATCGT (4 mismatches, BC2)
GTGAT (3 mismatches, BC3)
TGTCT (3 mismatches, BC4)

This sequence will be classified as 'BC1', because it has the lowest mismatch count.
If mismatches = 0 were specified, this sequence would be classified as 'unmatched', because, although BC1 had the lowest mismatch count, it is above the maximum allowed mismatches.

Matching barcodes at the end of the sequences does the same, but from the other side of the sequence.

With partial matching (very similar to indels):

Same as above, with the following addition: barcodes are also checked for partial overlap.

Example:
Input sequence is ATTTACTATGTAAAGATAGAAGGAATAAGGTGAAG
(Same as above, but note the missing 'G' at the beginning.)

Matching (without partial overlapping) against BC1 yields 4 mismatches:
ATTTACTATGTAAAGATAGAAGGAATAAGGTGAAGGATCT (4 mismatches)

Partial overlapping would also try the following match:
-ATTTACTATGTAAAGATAGAAGGAATAAGGTGAAGGATCT (1 mismatch)

Note: Scoring counts a missing base as a mismatch, so the final mismatch count is 2 (1 'real' mismatch, 1 'missing base' mismatch).
If running with mismatches = 2 (meaning allowing up to 2 mismatches), this sequence will be classified as BC1.

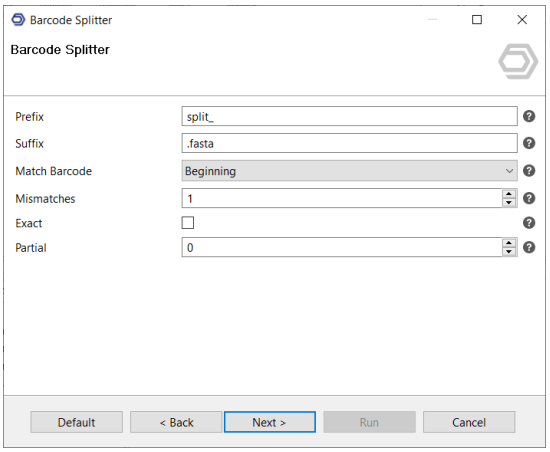


Figure 2: Wizard Page 2, Parameters

Page 3 - Output

Output Folder - Define a folder to save the results.

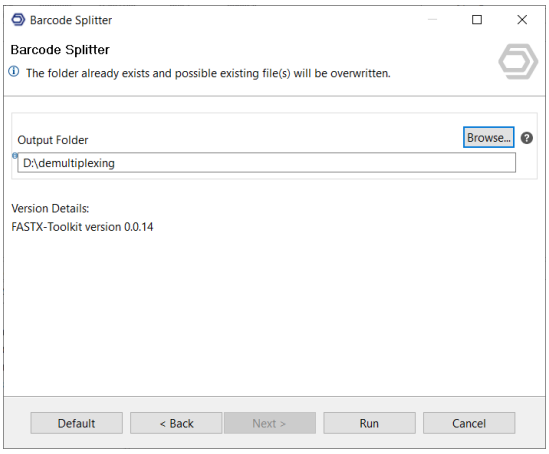


Figure 3: Wizard page 3, Output Folder

RESULTS

REFERENCES

FastX-Toolkit

Demultiplexing with Cutadapt

INTRODUCTION

Demultiplexing, or Barcode Splitting, is the step in processing where you use the barcode information to know which sequences came from which sample after they had all been sequenced together. Barcodes refer to the unique sequences ligated to each of your individual samples' genetic material before the samples got all mixed together. Depending on your sequencing facility, you may get your reads already split into individual fastq files, or they may be lumped together all in one fastq file with barcodes still attached for you to do the splitting. If this is the case, you should also have a mapping or barcode file telling you which barcodes correspond with which samples.
This tool takes FASTA/FASTQ files and splits them into several smaller files based on barcode matching. Cutadapt is used for this task.

CUTADAPT WIZARD

Page 1 - Input

Parameters

Input Reads- Select the FastQ/A files that contain sequences that have attached barcodes that link those sequences to the respective samples. Single-End and Paired-End files are allowed.

Paired-end Configuration - If Paired-End reads are provided, a pattern to distinguish upstream files from downstream files is required. The provided patterns are searched in the filenames right before the extension. The beginning of the filenames should be the same for both files of each sample.

- **Upstream Files Pattern:** Establish the pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern:** Establish the pattern to recognize downstream FASTQ files.

Barcode File - Select the mapping file that establishes the connection between each barcode and sample. In the case of Paired-End, Barcodes from this file will be matched against the Upstream Files.

Downstream Barcodes - Select to activate the barcode search in the Downstream Files. Only allowed if Paired-End data has been selected as input reads.

Downstream Barcode File - A text file containing the barcodes to be mapped against the Downstream Files.

Barcode File Format

Barcode sequences can be provided in three different formats:

1. Two-Columns TXT/CSV file: Barcode files are simple TXT files or CSV/TSV files. Each line should contain an identifier (descriptive name for the barcode), and the barcode itself (A/C/G/T), separated by a TAB character. Example:

```
BC1 GATCT
BC2 ATCGT
BC3 GTGAT
BC4 TGTCT
```

2. Three-Columns TXT/CSV file: This format is similar to the previous one but has a third column containing the names of the files where you want to look for each barcode. Example:

```
BC1 GATCT filename1
BC2 ATCGT filename1
BC3 GTGAT filename2
BC3 GTGAT filename3
BC4 TGTCT filename3
```

In this case, each barcode will be searched only in the files indicated in the third column.

3. Fasta file: Barcode sequences are contained in a fasta file preceded by its barcode IDs as fasta headers (Having ">" as a first character). Example:

```
>BC1
GATCT
>BC2
ATCGT
>BC3
GTGAT
```

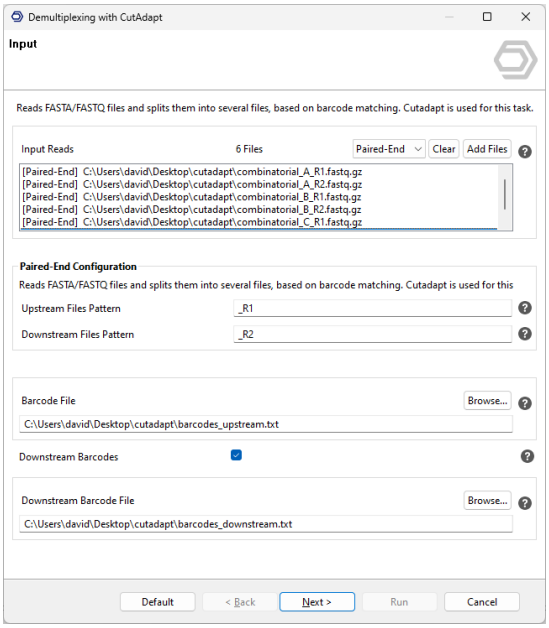
For each barcode and file, a new FASTA/FASTQ file will be created (with the barcode's identifier as part of the output file name). Sequences matching the barcode will be stored in the appropriate file. The name of the new files will contain the name of the original input file as well.

Running the above example (assuming the barcode file contains the above barcodes), will create the following files:

```
[filename]-BC1.fastq.gz
[filename]-BC2.fastq.gz
[filename]-BC3.fastq.gz
[filename]-BC4.fastq.gz
[filename]-unknown.fastq.gz
```

Take into account that, in this case, *.fastq.gz* has been chosen as the files suffix.

The 'unknown' file will contain all sequences that didn't match any barcode.



Page 2 - Configuration

Adapter Position - Match the barcodes at the beginning (5') of the sequences, at the end (3'), or anywhere along the upstream sequences.

Downstream Adapter Position -Only if the downstream adapter file has been provided. It allows indication of the position to look for the barcodes in the downstream input files.

Paired-End Adapter Strategy - If both upstream and downstream barcode files have been provided, it allows you to choose between 2 search strategies:

- *Unique Dual Indices*: Cutadapt only looks for the R1 and R2 barcodes in pairs. That is, the first R1 barcode is always used with the first R2 barcode, and so on.
- *Combinatorial Dual Indices*: Cutadapt uses all possible combinations between R1 barcodes and R2 barcodes.

Allowed Errors - Maximum number of allowed errors (mismatches and indels, if allowed) for barcodes, ranging from 0 to 10.

Allow Indels - Enable considering insertions and deletions as allowed errors.

Action - It allows us to indicate what to do with the matched sequences. It admits 4 different options:

- *Trim*: Cutadapt removes the matched sequences from the original input sequences.
- *None*: It does not modify the matched sequences.
- *Mask*: Write N characters at the positions where adapters have been found.
- *Lowercase*: Transform the matched section to lowercase. Leaves the rest of the input sequences as Uppercase.

Reverse Complement Search - Check to search the adapter sequences and their reverse complement across the input sequences. If unchecked, Cutadapt will only search barcodes in the same orientation that input sequences (from 5' to 3').

Output Configuration

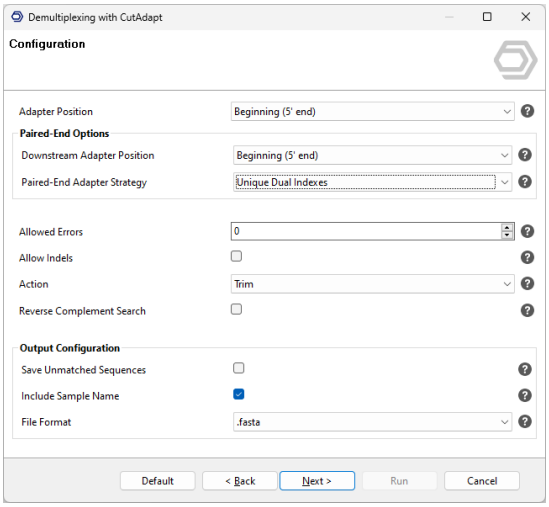
Save Unmatched Sequences - Check to save all unmatched sequences in an 'unknown' FastQ/A file.

Include Sample Name - Check to include the input file name as a prefix of the output files.

Disabling this option generates output files with the following file name structure: *[BarcodeID].fastq.gz*

In this case, all reads from all input files that match with a single barcode will be placed in the same output file.

File Format - This parameter allows the selection of the output format between *fasta* and *fastq*. Additionally, it indicates the degree of compression of the output files (*.gz* or not).



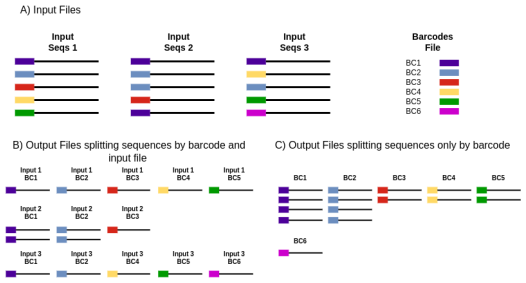
Example

In Figure 3, black lines symbolize sequencing reads, while colored boxes denote barcodes. The representation is divided into three sections:

A) Input Files: Comprising sequencing Fastq files (1-3) and barcode files.

B) Output files resulting from splitting the sequences by barcodes and input files.

C) Output files resulting from splitting the sequences only by barcodes.

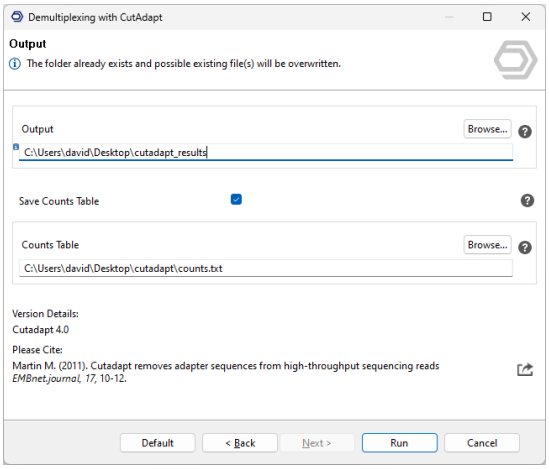


Page 3 - Output

Output Folder - Define a folder to save the results.

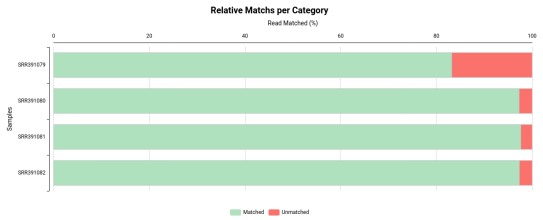
Save Counts Table - Check to save a table containing the results of matching all provided barcodes in each input file.

Counts File - Define a file name to save the barcode counts.



CUTADAPT RESULTS

- **Report** with information for each input sample regarding the proportion of reads matched with any provided barcode.
- Two **Charts**:
- Matches per Category Chart: Stacked bar plot representing the absolute number of reads in every input file and the number of them matched by any provided barcode.
- Relative Matches per Category Chart: Stacked bar plot similar to the previous chart with the relative number of reads per sample file (Figure 5). Useful when the number of reads diverges largely between input files.
- **Output FastQ/A files** containing all matched reads demultiplexed with their adapters trimmed. The demultiplexed reads can be grouped into these files by barcode and input file if the "*Include Sample Name*" parameter has been checked. Otherwise, they will be grouped only by the provided barcodes, even if they come from different input files.
- **Counts table** in a tabular TXT file. This file includes the count of all barcode matches along all the input samples. It also includes the total number of matches per sample and the number of unmatched sequences. It is formatted as a table, having the barcodes as rows and the samples as columns. Furthermore, it is compatible with any spreadsheet program.



REFERENCES

Please cite Cutadapt as:

Cutadapt removes adapter sequences from high-throughput sequencing reads.

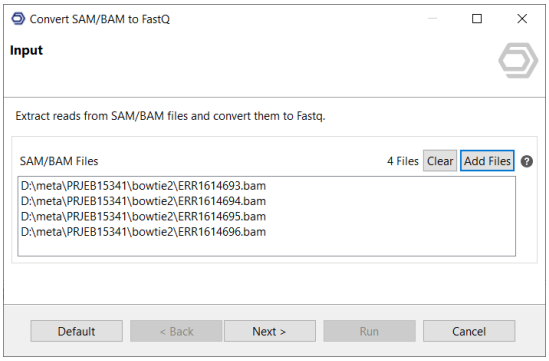
Marcel Martin. *EMBnet.journal*, 17(1):10-12, May 2011.
DOI: http://dx.doi.org/10.14806/ej.17.1.200

3.5.4 BAM Tools

Convert SAM/BAM to FastQ

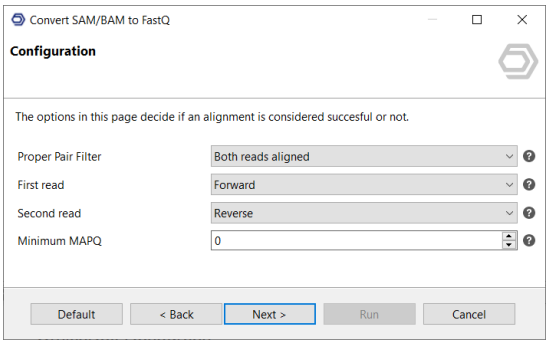
Extract reads from SAM/BAM files and convert them to FastQ.

Select the corresponding input files, single and paired-end are possible.

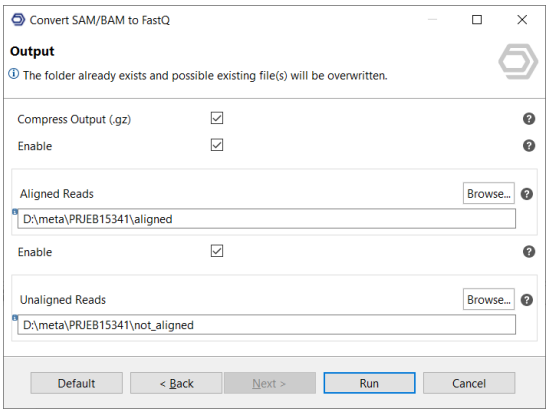


This page allows configuring under what circumstances reads are considered "aligned", i.e. they will be organized together with the aligned output fastq files.

- **Proper Pair Filter:** Decide if both reads have to be aligned or if one is enough.
- **Read Directions:** Choose in which direction the read pairs have to align.
- **Minimum MAPQ:** Filter out low-quality alignments (0 - 255).



Select where to save aligned and unaligned reads respectively and if the output should be compressed individually as in .fastq.gz.



3.5.5 Genome Browser

INTRODUCTION

The Genome Browser facilitates the visualization of various file types through a side-scrolling interface, where features are organized and displayed in sequence from left to right based on their start positions. Each file is depicted as a track within the browser, with support for multiple tracks including GFF, VCF, DNA Fasta, and BAM formats. Users have the flexibility to reorder, hide, or close tracks utilizing the provided track controls (

). The visualization process allows for one chromosome to be viewed at any given time; this includes a comprehensive representation of the entire chromosome in the top box and a detailed view of the specific region currently under examination in the smaller box.

Figure 1:*Genome Browser with all 4 tracks.*

GFF TRACK

The GFF track is able to detect types to group features by genes (genes, transcripts, exons and CDS and the relationship between them by using ID and Parent attributes, see GFF3 Specification for more details). Genes are in blue, the transcripts are the small red lines, non-coding exons are white boxes with orange border and exons with CDS are filled with orange color. Other non-grouped features appear in grey. GFF Viewer can be opened with the context menu option in the **File Manager** when selecting a GFF file and using the context menu option *Show in GFF Viewer* from the **table**when exploring a GFF file.

With the **Features button** controls you can classify the types for *Genes*, *Transcripts*, *Exons* and *CDS*, this will modify the Gene group visualization, by default these types will be set automatically. The checkboxes will hide or show the features, i.e. if you want to show only the genes, you need to uncheck the Transcripts, Exon, CDS, and Unclassified checkboxes.

A tooltip will appear on mouse hover, it will show additional information about the feature and the information is collected from the GFF file.

Figure 2: *GFF viewer.*

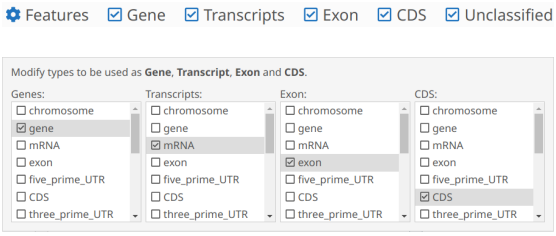


Figure 3: *Feature controls.*

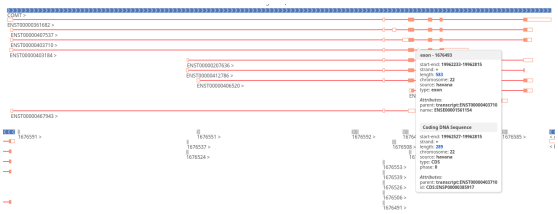


Figure 4. *Feature tooltip.*

VCF TRACK

The VCF track shows the variants in the corresponding position and if you zoom in enough, the alternative nucleotide will be shown.

Figure 5:*VCF track*

BAM TRACK

The BAM track shows the reads of a BAM file and if the sequence track is active, will also paint the differences between the read sequence and the sequence track. If you click on the

checkbox, if the BAM has paired reads, the pair will be painted one beside the other.

Figure 6:*BAM track*

DNA FASTA TRACK

The DNA Fasta or sequence track shows the nucleotides and is only shown on low region lengths.

Figure 7:*DNA Fasta Track*

NAVIGATION

Navigation is executed using the mouse. To initiate scrolling, left-click and hold the button, then move the mouse horizontally to navigate left or right. Additionally, it is possible to focus on a specific feature or gene by double-clicking on it. Zooming in and out within the current region can be achieved through the "Zoom buttons" located on the Toolbar. Should you zoom out sufficiently, the display will transition to show a histogram for broader visualization purposes. The chromosome box allows for selection of new regions either by clicking directly or selecting a new box area through click-and-drag actions.

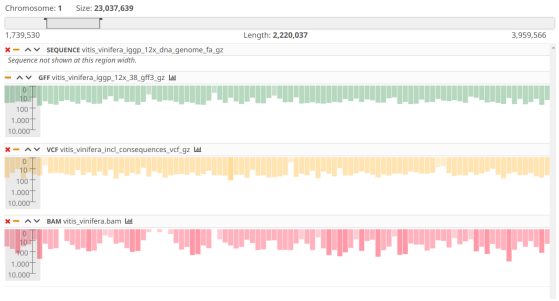


Figure 8. *Histogram, the small rectangle on the Chromosome box shows the region painted as a histogram.*

It is also possible to find features on the GFF and the VCF track using the **Search field** in the Toolbar, the search results will be shown on a panel beside the Toolbar.

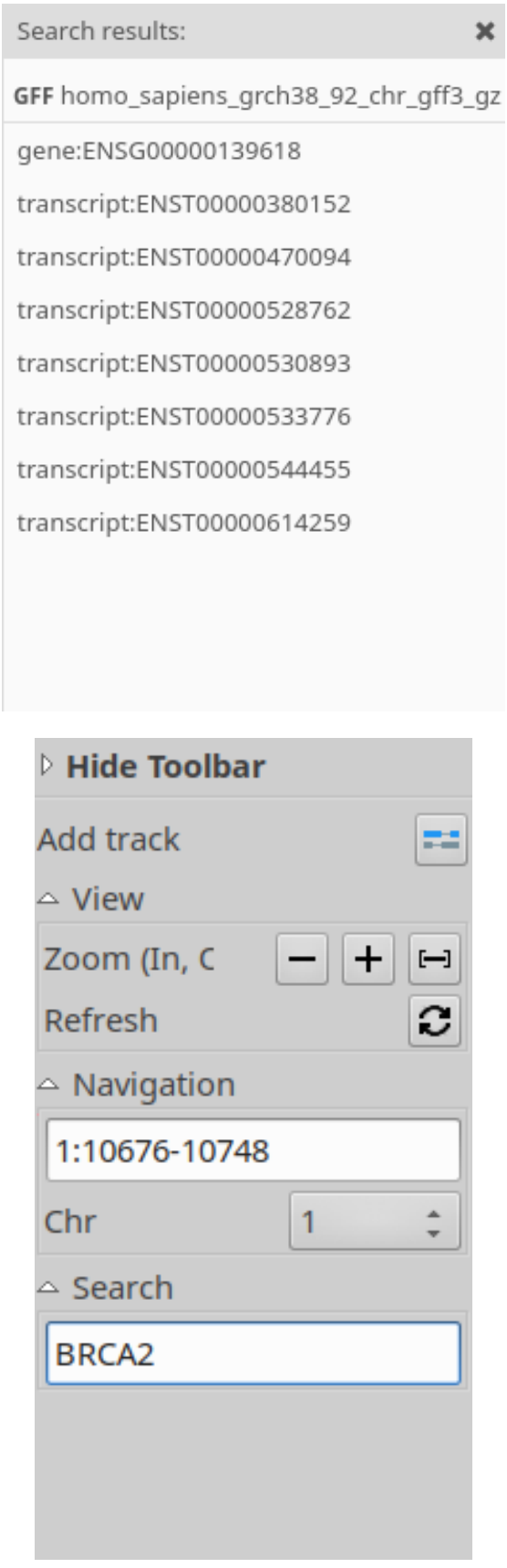


Figure 9: *Toolbar*

Also, you can add more tracks to the current Genome Browser using the **Add track** button. A window will appear the available files that can be opened with a Genome Browser as tracks.

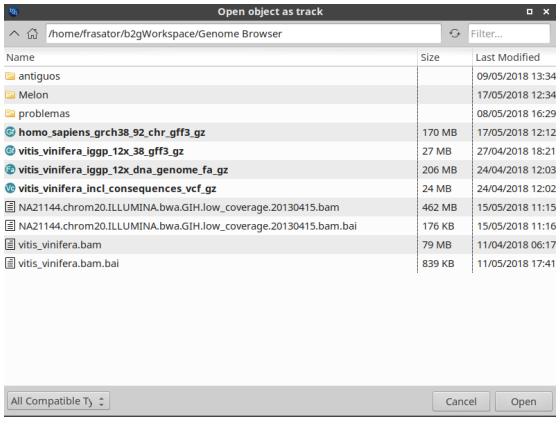


Figure 10:*Browse tracks to be added to the Genome Browser*

Load files

Files must be loaded in order to be shown on the Genome Browser to do that just click on the **File menu** and under the **Load sub-menu** you can load GFF, VCF, DNA Fasta and BAM files. These files can be loaded in either with .gff or .gz extension.

A window will appear to select the file to import, use the browse button to select the file from the file system and finally click on the **Load button** to start the Load process, the load time depends on the file size.

Genome Browser can also be opened from a **Table** view, If you have already a file open in a table, use the context menu option **Show in Genome Browser**, by right-clicking on a row, a Genome Browser will be shown on the region of that feature.

Once loaded you may save the file, once the file is saved you can visualize it using the Genome Browser.

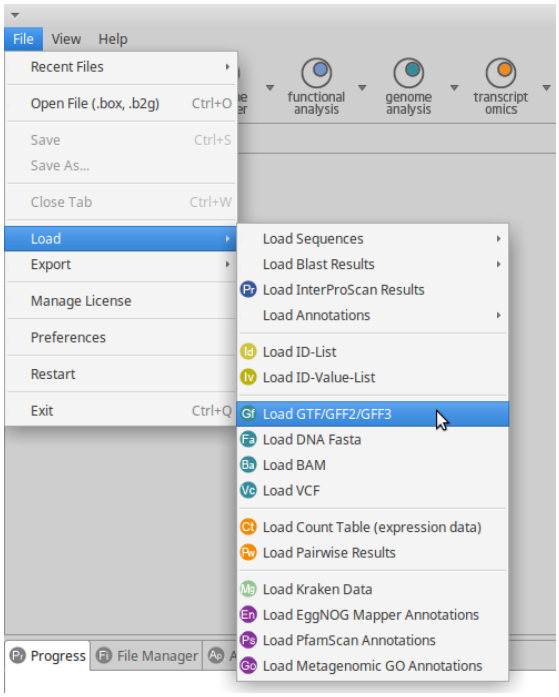


Figure 11: *Load Files to see in the browser*

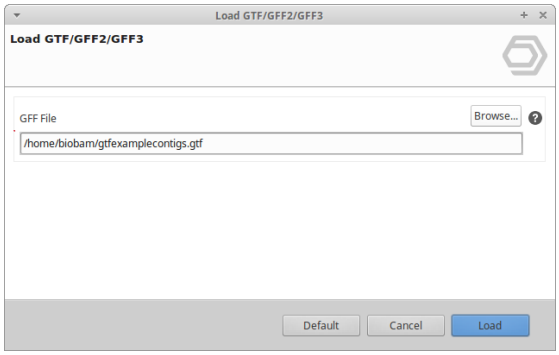


Figure 12:*Browse for GFF file*

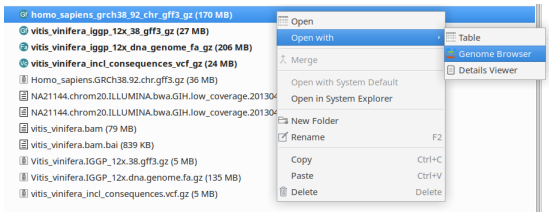


Figure 13:*Open the saved file directly with the Genome Browser*

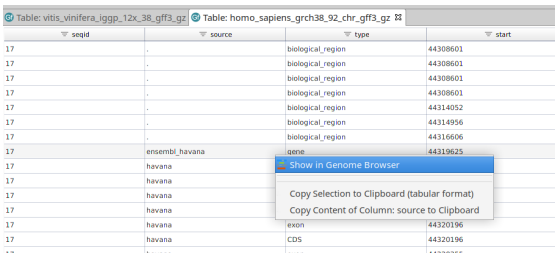


Figure 14: *Open the Genome Browser from GFF table*

Additional Resources

- Example Dataset: Download

3.5.6 Venn Diagram

Input Files

Venn Diagram tool allows you to select multiple ID List or ID value list in text or BOX/B2G format and draw the intersection of the elements of the lists.

This functionality can be found under **General Tools → Venn Diagram.** The wizard allows to select input files, you can mix the supported types (**ID List** or **ID value list**), with different formats (**Plain text** or **B2G** or **BOX**).

After selecting the files just press the **Run** button. A new tab will appear with the Venn diagram.

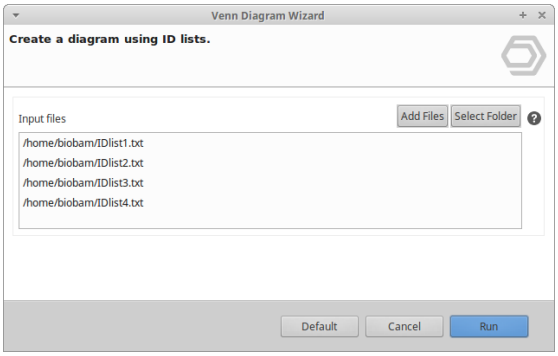


Figure 1:*Dialog to add lists to generate the Venn Diagram*

Sidebar Options

There are different options to customize Venn visualizations:

- **Proportional**. This check box allows you to change how the size of the circles is calculated, by setting this option to **true** will paint the size of the circles proportionally to the number of elements the list contains. When **false** all the circles will use the same size.
- **Grayscale**. If **true**, all circles will be painted in different shades of grey. Set to **false** to use a normal color.
- **Font size**. Use the **plus** or **minus** icon to **increase** or **decrease** the font size.
- **List control**. There will be one for each list you load with the wizard. Each control customizes each list individually.
- The **Check box** hides or shows the circle represented for this list.
- The **Color box** allows changing the color of the circle.
- The **Text field** updates the list name.
- **Table button.**This will open a table where rows are the union of all lists. The column tag indicates in which lists this element appears. You can use this table to sort and filter the elements and extract the selected rows to a new filtered Venn Diagram.

For further information this blog post might be interesting

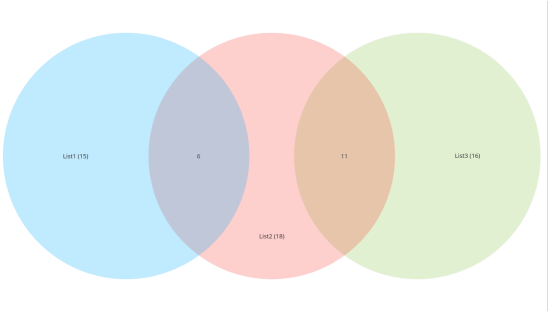


Figure 2: *Venn Diagram of 3 lists (proportional mode)*

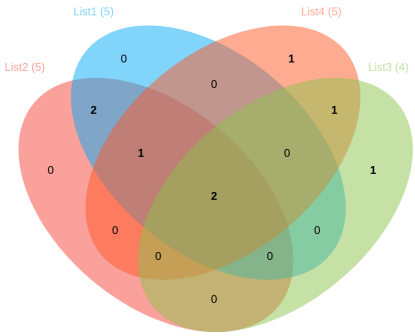


Figure 3: *Venn Diagram of 4 lists (non proportional mode)*

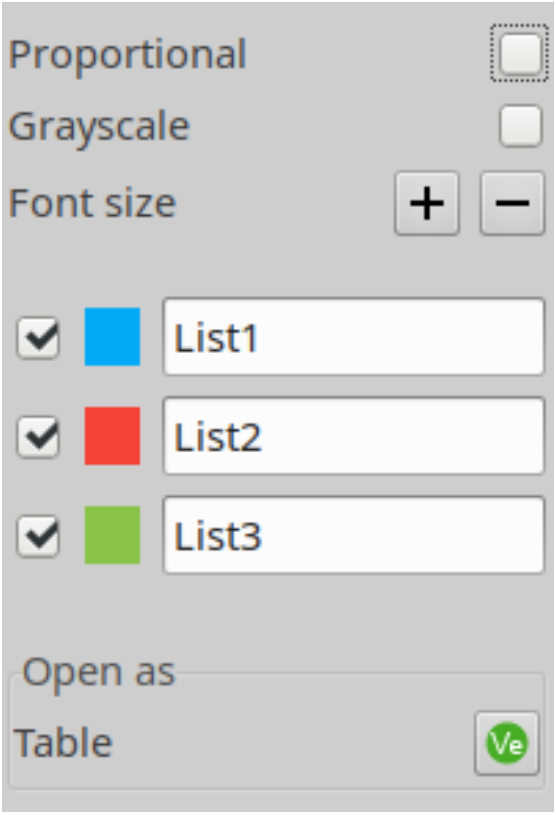


Figure 4:*Configuration panel*

Table: List1_List2_List3

Nr	Tags	
1	List3	C02006F12
2	List2 List3	C02006E12
3	List2 List3	C02006D12
4	List3	C02006F10
5	List1 List2	C02006C12
6	List2 List3	C02006E10
7	List2 List3	C02006D10
8	List1 List2	C02006C10
9	List1	C02006B10
10	List2 List3	C02006F08
11	List2 List3	C02006E08
12	List3	C02006G06
13	List2	C02006D08

Figure 5: *Table with the list and the Tags*

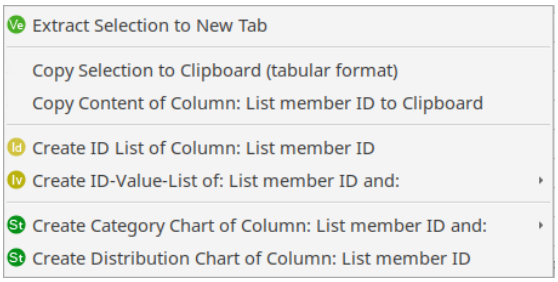


Figure 6:*Extract Selected rows from the table*

3.6 Workflows

3.6.1 Introduction

OmicsBox provides an interface to create, edit and run workflows based on the Common Workflow Language (CWL) specification. This interface allows to describe all analysis steps using the functions and tools offered by OmicsBox and connect them to perform a complete analysis in a single run. Workflows are highly customisable since users can define input data, configure the parameters of each step, save and export results, generate charts and statistics and more.

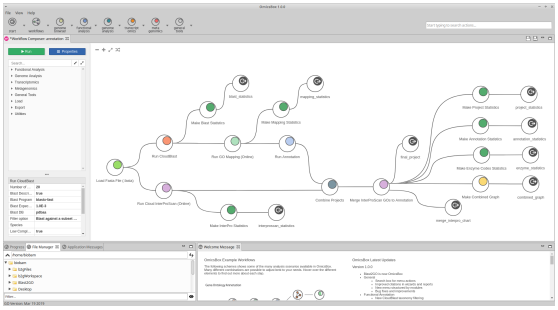


Figure 1:*Workflow Composer Interface*

3.6.2 Workflow Composer

The OmicsBox workflow composer interface offers all the necessary options to manage workflows. You can access the composer using the "Workflows" toolbar item or the "Create Workflow" menu option in the workflows toolbar menu.

Using the "Properties" button in the side panel it is possible to edit the documentation of the workflow. You can write whatever information it is useful, like the author name, author email and a description of the workflow.

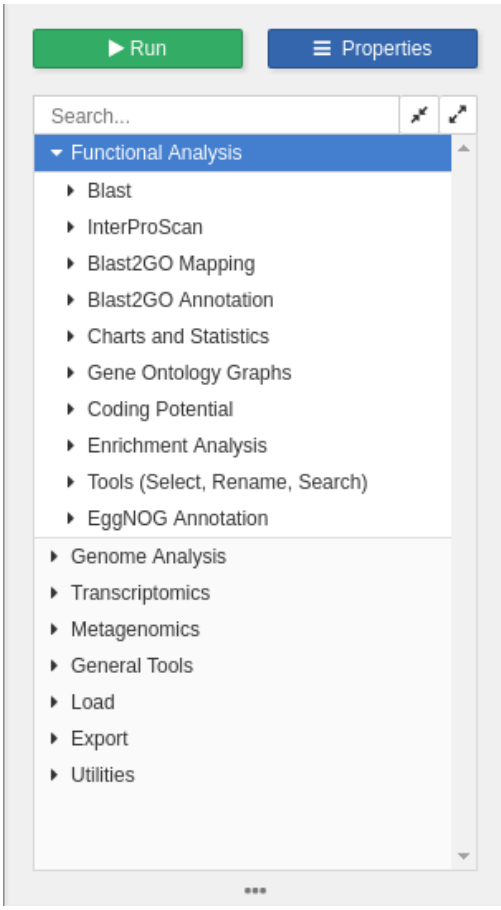


Figure 2:*List of actions that can be included in a workflow*

3.6.3 Design Workflows

To create a workflow, start adding some steps. The side panel (on the left) contains the list of actions (that may vary depending the apps installed in OmicsBox) that can be used as workflow steps (Figure 2). To add an action to the workflow click on the corresponding plus symbol next to the action's name.

Each action is represented by an icon (Figure 3). On the left side of the icon are placed the connections for every input of the action (e.g. project, count table, etc) and on the right side are placed the connections for every output it produces as result (chart, graph, etc).

To connect two steps of the workflow, click on the small circle representing the output connection of the first action and drag it to the small circle representing the input connection of the second action. If the connection is valid (i.e. both types match) the small input connection circle should turn green, and a line connecting both circles should be displayed. Otherwise, the small input connection circle should turn red, indicating the selected output can not be used as input for that action.

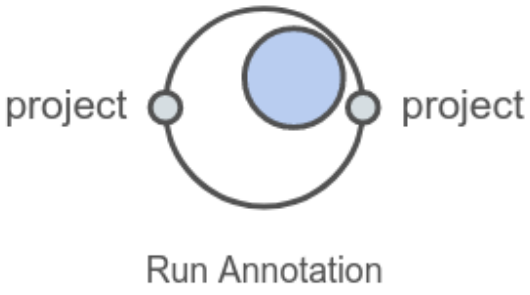


Figure 3:*Action icon*

CONFIGURE WORKFLOW STEPS

Most workflow steps can be configured. If a step needs to be configured (because its parameters are not valid) it will be highlighted in red color and it will not be possible to run the workflow (Figure 4). To configure the step right-click on the step icon and select the "Edit Parameters" option. The red color should disappear as the parameters are now valid. The parameters of each step can be consulted in the bottom region of the side panel.

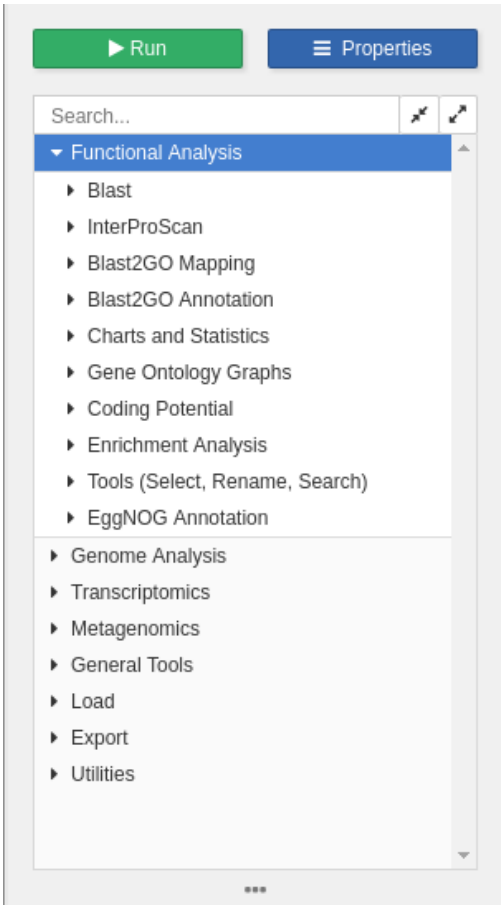


Figure 4: *Valid step (left), invalid step (right).*

DEFINE WORKFLOW INPUTS

Workflow inputs are .b2g files by definition. To use a .b2g file as input click on the input connection on the left side of the icon and drag it out. The input file can be selected by right-clicking on this step and selecting the "Select Input File" option (Figure 4 top) or in the "Run Workflow" wizard.

Some actions don't require any input and produce a result that can be used as input by other actions. These actions can be incorporated as first step in a workflow (e.g. Load Fasta, Eukaryotic and Prokaryotic Gene Finding, Create Count Table, etc).

Figure 5:*Input data definition*

DEFINE WORKFLOW OUTPUTS

Like inputs, workflow outputs are .box/.b2g files by definition. To save the results as .box/.b2g files click on the output connection on the right side of the icon and drag it out. The output name (file name in the end) can be selected by right-clicking on this step and selecting the "Change Output Name" option (Figure 6). Later in the "Run Workflow" wizard it is possible to choose the output folder of every output, or use a common output folder for all workflow's outputs.

If you want to export the output of a workflow as a regular file (e.g. .txt, .csv, .png) instead as .box/.b2g file, use the several export actions to export annotations, charts and statistics that OmicsBox offers. These actions can be incorporated as the last step in a workflow (e.g. Generic Export, Export Chart, Export Report, etc).

Figure 6:*Output data definition*

3.6.4 Run workflows

Once the workflow is ready click on the green "Run" button on the side panel to open the "Run Workflow" wizard. Here you can select the inputs files and the outputs folder(s) to save the results or use a common folder for all results (Figure 7). Click "Run" to execute the workflow. You will see a new progress bar for the workflow execution, and an additional progress bar for every step, so it is possible to cancel the whole workflow or just a certain step and continue with the others.

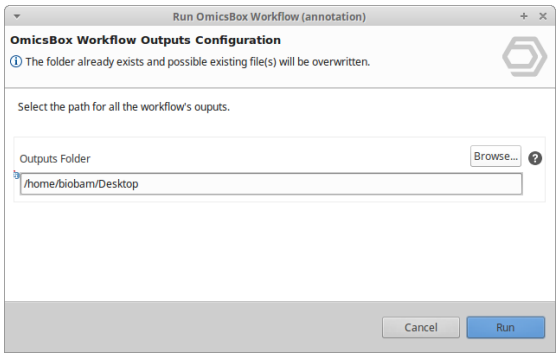


Figure 7:*Outputs folder configuration*

4. OmicsBox Modules

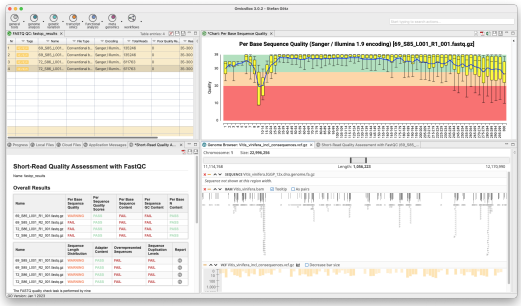
4.1 OmicsBox Modules

Five modules to easily process large and complex data sets

From raw reads to functional insights fast and easy

4.2 Module Genome Analysis

4.2.1 Module Genome Analysis



The OmicsBox Genome Analysis module allows the characterization and analysis of newly sequenced genomes, from raw reads to gene structures in an efficient and user-friendly way.

- **Quality Control and Assessment:**Use **FastQC** and **Trimmomatic** to perform the quality control of your samples, and filter reads, and to remove low-quality bases.
- ***De novo*Assembly**: The assembly feature allows the reconstruction of whole-genome sequences without a reference genome or specific hardware requirements. Assemble sequencing data from both, short and long-read technologies with 3 different algorithms: **ABySS**, **SPAdes,** and **Flye**.
- **Alignment and Polishing:**Align short sequencing reads against large sequences with **BWA**, and correct draft assemblies from long-reads with **Pilon**.
- **Repeat Masking:**Mask repeats and low-complexity DNA sequences of your eukaryotic genome assemblies with **RepeatMasker** to improve downstream gene predictions.
- **Gene Finding:**Perform prokaryotic (**Glimmer**) and eukaryotic (**Augustus**) gene predictions to characterize genome structure. The eukaryotic gene prediction offers RNA-seq intron hint support.

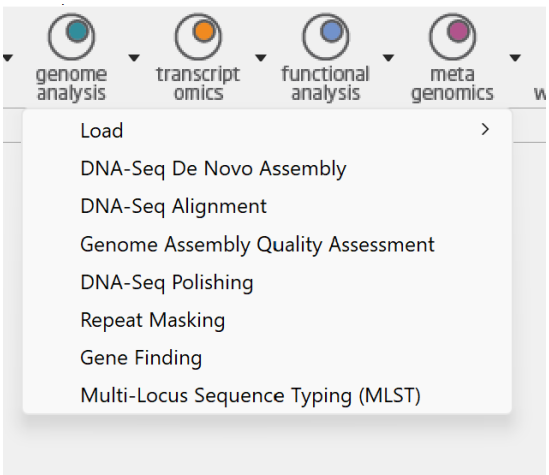


image-20240430-143626.png Additional Resources

Genome Analysis use case: Genome Assembly Annotation Sarocladium Oryzae.

Genome Analysis Example Dataset: Download.

4.2.2 DNA-Seq de Novo Assembly

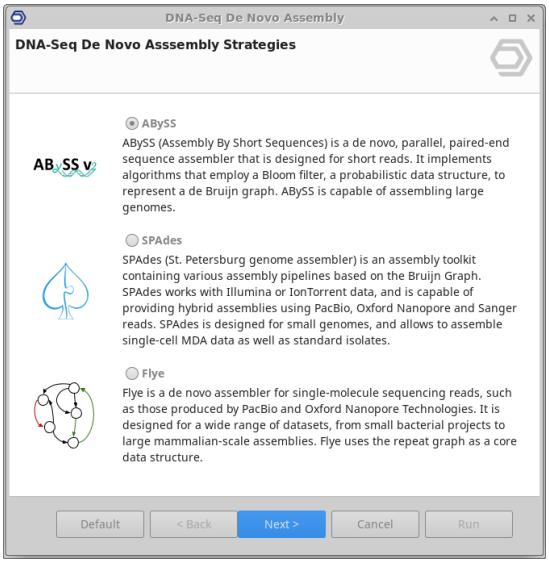
DNA-Seq de Novo Assembly

Genome assembly refers to the process of taking a large number of short DNA reads and putting them back together to create a representation of the whole genome from which the DNA originates. *De novo* genome assemblies assume no prior knowledge of the source DNA sequence length, layout or composition (i.e. no reference genome is available). The goal of an assembler is to produce long contiguous pieces of sequences (contigs) from DNA-seq reads. The contigs are then joined together to form scaffolds where possible. Short-insert paired reads provide increased information for maximizing sequencing coverage, while long-insert mate paired-end reads can pair sequence fragments across greater distances. This is especially helpful to cover highly repetitive regions.

This functionality can be found under **Genome Analysis → DNA-Seq De novo Assembly.**

Three assembly strategies are available:

- **ABySS:** ABySS (Assembly By Short Sequences) is a *de novo*, parallel, paired-end sequence assembler that is designed for short reads. It implements algorithms that employ a Bloom filter, a probabilistic data structure, to represent a de Bruijn graph. ABySS is capable of assembling large genomes.
- **SPAdes:** SPAdes (St Petersburg genome assembler) is an assembly toolkit containing various assembly pipelines based on the Bruijn Graph. SPAdes works with Illumina and IonTorrent data and is capable of providing hybrid assemblies using PacBio, Oxford Nanopore and Sanger reads. SPAdes is designed for small genomes, and allows to assemble single-cell MDA data as well as standard isolates.
- **Flye:** Flye is a de novo assembler for single-molecule sequencing reads, such as those produced by PacBio and Oxford Nanopore Technologies. It is designed for a wide range of datasets, from small bacterial projects to large mammalian-scale assemblies. Flye uses the repeat graph as a core data structure.



ABySS

INTRODUCTION

ABySS 2.0 is a multistage *de novo* assembly pipeline consisting of unitig, contig, and scaffold stages.

- At the **unitig stage**, the program performs the initial assembly of sequences according to the De Bruijn graph assembly algorithm. The unitig stage loads the full set of k-mers from the input sequencing reads into a hash table and stores auxiliary data for each k-mer such as the number of k-mer occurrences in the reads and the presence/absence of possible neighbor k-mers in the De Bruijn graph.
- At the **contig stage**, the paired-end reads are aligned to the unitigs and the pairing information is used to orient and merge overlapping unitigs.
- At the **scaffold stage**, the mate-pair reads are aligned to the contigs to orient and join them into scaffolds, inserting runs of "N" characters at gaps in coverage and for unresolved repeats.

The main innovation of ABySS 2.0 is a Bloom filter-based implementation of the unitig assembly stage. It reduces the overall memory requirements, enabling assembly of large genomes. A Bloom filter is a compact data structure for representing a set of elements that supports operations of inserting elements and querying the presence of elements. The Bloom filter data structure consists of a bit vector and one or more hash functions, where the hash functions map each k-mer to a corresponding set of positions within the bit vector (bit signature for the k-mer).

During unitig assembly, two passes are made through the input sequencing reads:

1. In the first pass, k-mers are extracted from the reads and are loaded into a Bloom filter. The program discards all k-mers with an occurrence count below a user-specified threshold (typically in the range of two to four). In this way, k-mers caused by sequencing errors are filtered out. The retained k-mers are known as solid k-mers.
2. In the second pass, the program identifies reads that consist entirely of solid k-mers, and extend them left and right within the De Bruijn graph to create unitigs.

Please cite ABySS 2.0 as:

Jackman SD, Vandervalk BP, Mohamadi H, et al (2017). "ABySS 2.0: resource-efficient assembly of large genomes using a Bloom filter". Genome Res. 2017;27(5): 768-777.

RUN ABYSS ASSEMBLY

This functionality can be found under **Genome Analysis → DNA-Seq *de novo* Assembly → ABySS**. The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, and Figure 3).

Input

- **Input Reads:** First, choose the type of sequencing data. Then, select the files of this type of data for the assembly. Both paired-end and single-end short reads can be provided, and both types of data can be combined in the same run.
- **Additional Data:** ABySS supports additional data types as supplementary information:
- Additional Paired-end Libraries: Paired-end libraries that will be used only for merging unitigs into contigs and will not contribute toward the consensus sequence.
- Mate-pair Libraries: Mate-Pair libraries that will be used for scaffolding. Mate-Pair libraries that will be used for scaffolding. Mate-pair libraries do not contribute toward the consensus sequence.
- Linked Reads: Linked reads from 10x Genomics Chromium. The linked reads are used to correct assembly errors and scaffolding.
- Long Sequences Libraries: Provide long sequence libraries (such as RNA-Seq contigs) that will be used for rescaffolding. Long sequence libraries do not contribute toward the consensus sequence.
- **Paired-end Configuration:** If paired-end reads are provided, a pattern to distinguish upstream files from downstream files is required. The provided patterns are searched in the filenames right before the extension. The beginning of the filenames should be the same for both files of each sample.
- **Upstream Files Pattern**: Establish the pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern**: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is SRR037717_1.fastq and the downstream SRR037717_2.fastq, "_1" should be established as the upstream pattern and "_2" as the downstream pattern.

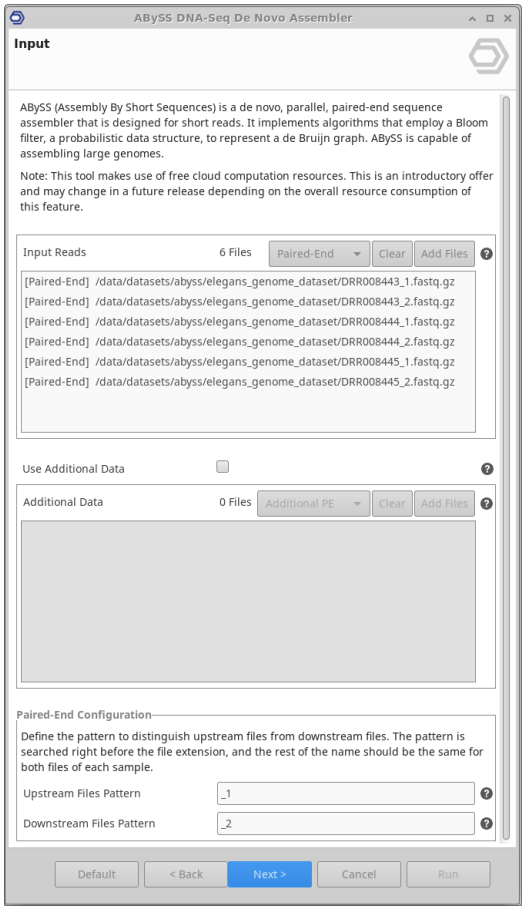


Figure 1: *Input Page*

Configuration

- **K-mer Size:** The term k-mer refers to all possible subsequences of the given length that are contained in a read. In sequence assembly, k-mers are used during the construction of De Bruijn graphs. The choice of the k-mer size has many different effects on the sequence assembly, it is advisable to try different values and check the results to choose the best one. It is recommended to use odd values of at least half the length of the reads.
- **Use paired De Bruijn graph:** Assembly will be performed using a paired De Bruijn graph. In this mode, k-mer pairs are used, which consist of two equal-sized k-mers separated by a fixed distance. To assemble using the paired De Bruijn graph mode, specify the k-mer pair span (distance between k-mers).
- **K-mer Pair Span:** Set the span of a k-mer pair (distance between k-mers).
- **Minimum Alignment Length:** Establish the minimum alignment length of a read (bp). This means that there must be a perfect match of the established length between each read and its target contig.
- **Hash Functions:** Set the number of Bloom filter hash functions. K-mers from each input sequencing read are loaded into the Bloom filter by computing the hash values of each k-mer sequence and setting the corresponding bit.

- **K-mer Count Threshold:** Set the k-mer count threshold for Bloom filter assembly. Optimal values are typically in the range of 2-4. K-mers with an occurrence count below the threshold will be discarded.

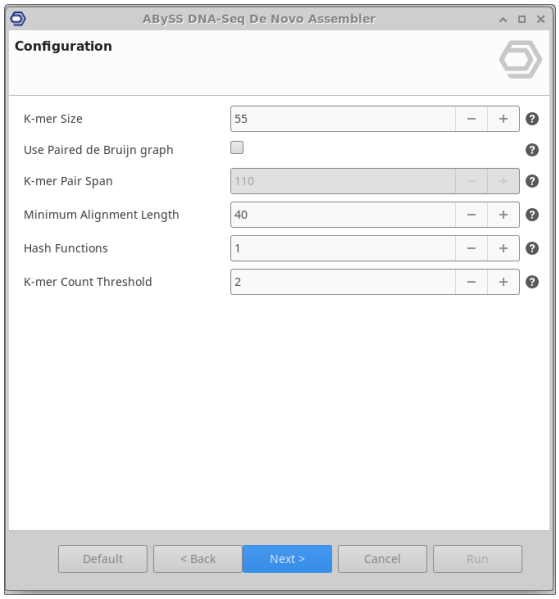


Figure 2: *Configuration Page*

Output

- **Unitigs Fasta:** Where to store the Fasta file containing the assembled unitigs.
- **Contigs Fasta:** Where to store the Fasta file containing the assembled contigs.
- **Scaffolds Fasta:** Where to store the Fasta file containing the assembled scaffolds.
- **Long Scaffolds Fasta:** Where to store the Fasta file containing the assembled long scaffolds.

Note that this file is only generated if long sequence data were provided.

- **Save Graph Files:** Save the final repeat graphs in a .dot file. Graph files are generated for contigs and scaffolds.
- **Graph Files:** Select a folder to store the graph files (.dot).

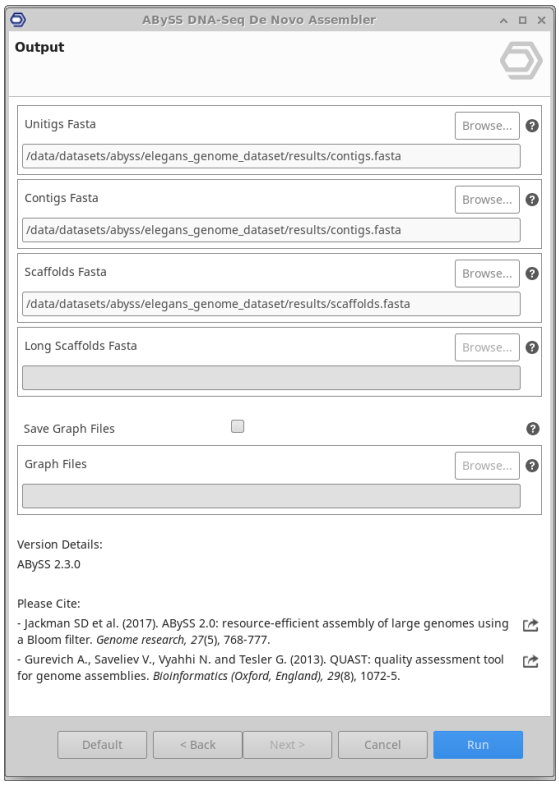


Figure 3: *Output Page*

RESULTS

ABySS returns the assembled sequences in three FASTA files (four if long sequence libraries were provided). Each one corresponds to a different stage of the assembly procedure:

- **Unitigs**: Contains sequences assembled without using paired-end information. In case you provide only single-end data, this will be the only result file, since pairing information is required to assemble contigs.
- **Contigs**: Contains sequences assembled with paired information, scaffolding over sequencing coverage gaps, but no repeats.
- **Scaffolds**: Contains sequences assembled with paired information, scaffolding over sequencing coverage gaps and repeats.
- **Long Scaffolds:** Contains sequences that were obtained by rescaffolding using long sequences libraries.

Unitigs/contigs/scaffolds names in ABySS output FASTA files have the following format:

> 4 678 16718

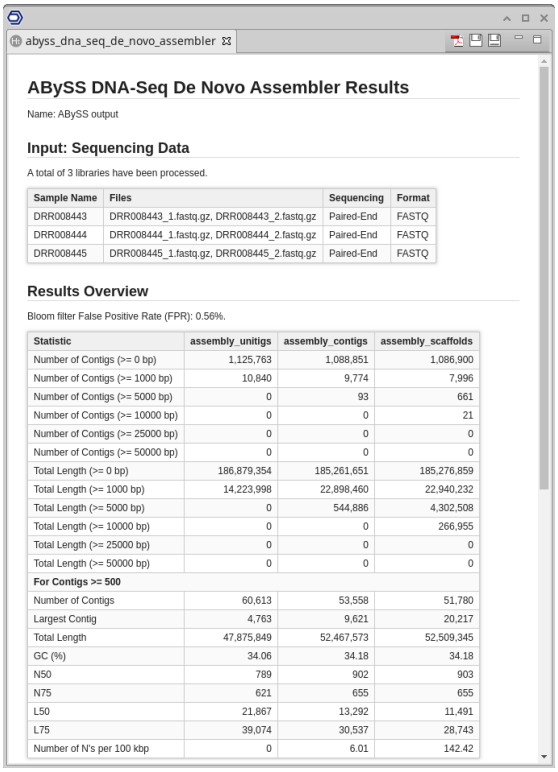
- 4 is the name of the sequence.
- 678 is the sequence length in nucleotides.
- 16718 is the number of kmers that mapped to the sequence during assembly.

If the "Save Graph Files" option is checked, ABySS returns the sequence overlap graphs in Graphviz dot format. The GraphViz DOT syntax is well defined and implemented by a number of existing graph tools.

For further information about how ABySS represents a sequence overlap graph, please visit the ABySS File Formats page.

In addition to the resulting FASTA files, a report and a chart are generated. The report shows a summary of the DNA-Seq *De Novo* Assembly results (Figure 4). This page contains information about the input sequencing data and a results overview. The Results Overview table shows a number of common statistics used to describe the quality of a sequence assembly:

- **N50:** This statistic defines the assembly quality in terms of contiguity. N50 is calculated by first ordering every unitig, contig or scaffold from longest to shortest. Next, starting from the longest sequence, the lengths of each sequence are summed up, until this running sum equals one-half of the total length of all sequences in the assembly. The N50 of the assembly is the length of the shortest contig in this list. Higher values of N50 indicate a better assembly. Note that any Nx statistic is calculated in the same way, e.g. N75 is calculated summing up all the lengths until the sum equals 75% of the total length.
- **L50:** Defined as the smallest number of contigs whose lengths sum makes up half of the total assembly length.
- **Bloom filter False Positive Rate (FPR):** The Bloom filter can generate *false positives* when the bit signatures of different k-mers overlap by chance. This means that a certain fraction of k-mer queries will return true even though the k-mers do not exist in the input sequencing data. Users are recommended to target a Bloom filter false positive rate (FPR) smaller than 5%. Parameters such as the k-mer size, hash functions or k-mer count threshold can influence the false positive rate.



abyss_dna_seq_de_novo_assembler

ABySS DNA-Seq De Novo Assembler Results

Name: ABySS output

Input: Sequencing Data

A total of 3 libraries have been processed.

Sample Name	Files	Sequencing	Format
DRR008443	DRR008443_1.fastq.gz, DRR008443_2.fastq.gz	Paired-End	FASTQ
DRR008444	DRR008444_1.fastq.gz, DRR008444_2.fastq.gz	Paired-End	FASTQ
DRR008445	DRR008445_1.fastq.gz, DRR008445_2.fastq.gz	Paired-End	FASTQ

Results Overview

Bloom filter False Positive Rate (FPR): 0.56%.

Statistic	assembly_unitigs	assembly_contigs	assembly_scaffolds
Number of Contigs (>= 0 bp)	1,125,763	1,088,851	1,086,900
Number of Contigs (>= 1000 bp)	10,840	9,774	7,996
Number of Contigs (>= 5000 bp)	0	93	661
Number of Contigs (>= 10000 bp)	0	0	21
Number of Contigs (>= 25000 bp)	0	0	0
Number of Contigs (>= 50000 bp)	0	0	0
Total Length (>= 0 bp)	186,879,354	185,261,651	185,276,859
Total Length (>= 1000 bp)	14,223,998	22,898,460	22,940,232
Total Length (>= 5000 bp)	0	544,886	4,302,508
Total Length (>= 10000 bp)	0	0	266,955
Total Length (>= 25000 bp)	0	0	0
Total Length (>= 50000 bp)	0	0	0
For Contigs >= 500			
Number of Contigs	60,613	53,558	51,780
Largest Contig	4,763	9,621	20,217
Total Length	47,875,849	52,467,573	52,509,345
GC (%)	34.06	34.18	34.18
N50	789	902	903
N75	621	655	655
L50	21,867	13,292	11,491
L75	39,074	30,537	28,743
Number of N's per 100 kbp	0	6.01	142.42

Figure 4: *Summary Report*

The Nx plot (Figure 5) shows Nx values as x varies from 0 to 100 %. The Nx values are displayed for unitigs, contigs and scaffolds.

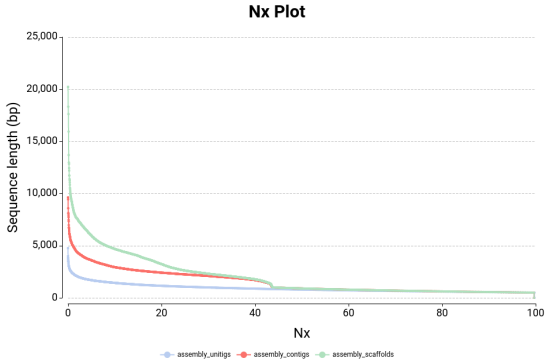


Figure 5: *Nx Plot*

SPAdes

INTRODUCTION

SPAdes is a *de novo* genome assembly pipeline that can deal with data coming from several sequencing technologies and supports hybrid and single-cell assemblies. The SPAdes assembly pipeline consists of four stages:

1. Assembly graph construction. SPAdes uses the *multisized de Bruijn graph*, implements new bulge/tip removal algorithms, detects and removes chimeric reads, aggregates biread information into distance histograms, and allows to backtrack the performed graph operations.
2. *k-bimer* adjustment: SPAdes derives accurate distance estimates between k-mers in the genome using joint analysis of distance histograms and paths in the assembly graph.
3. Constructs the paired assembly graph: Inspired by *Paired de Bruijn graphs* (PDBG) approach.
4. Contig construction: SPAde constructs DNA sequences of contigs and the mapping of reads to contigs by backtracking graph simplifications.

SPAdes uses a modification of Hammer for error correction and quality trimming prior to assembly.

In general, SPAdes uses two techniques for scaffolding.

- SPAdes tries to estimate the size of the gap separating contigs using read pairs.
- SPAdes, using the assembly graph, joins contigs that are separated by a complex tandem repeat, that cannot be resolved exactly, with a fixed gap size of 100 bp.

Contigs produced by SPAdes do not contain N symbols.

Please, cite SPAdes as:

- Nurk, Bankevich et al., 2013.
- Bankevich, Nurk et al., 2012.
- Antipov et al., 2015 (in case you perform hybrid assembly using PacBio or Nanopore reads).
- Prjibelski et al., 2014 (if you use multiple paired-end and/or mate-pair libraries).
- Vasilinetc et al., 2015 (if you use multiple paired-end and/or mate-pair libraries).

RUN SPADES ASSEMBLY

This functionality can be found under **Genome Analysis → DNA-Seq*de novo*Assembly → SPAdes**. The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

Input

- **Input Reads:** Select the files containing the sequencing libraries (reads). The assembly strategy requires at least one of these types of sequencing libraries.
- Illumina single-end, paired-end, or high-quality mate-pairs.
- IonTorrent single-end, paired-end, or high-quality mate-pairs.
- PacBio CCS reads (should be provided as single-end data).

These files are assumed to be in FASTQ format. For IonTorrent data, SPAdes supports unpaired reads in unmapped BAM format.

- **IonTorrent Data:** This option is required when assembling IonTorrent data. Illumina and IonTorrent libraries should not be assembled together. For IonTorrent data, SPAdes also supports unpaired reads in unmapped BAM format (like the one produced by the Torrent Server).
- **Single-cell Data:** This option is required for Multiple Displacement Amplification (MDA) single-cell data assembly.
- **Paired-end Configuration:** If paired-end reads are provided, a pattern to distinguish upstream files from downstream files is required. The provided patterns are searched in the filenames right before the extension. The beginning of the filenames should be the same for both files of each sample.
- **Upstream Files Pattern**: Establish the pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern**: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is SRR037717_1.fastq and the downstream SRR037717_2.fastq, "_1" should be established as the upstream pattern and "_2" as the downstream pattern.

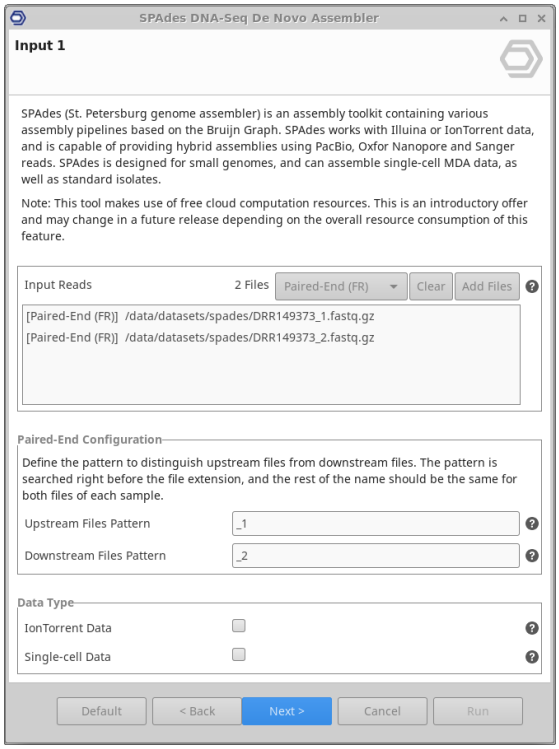


Figure 1: *Input Page*

Input 2

- **Use Additional Mate-Pair Data:** SPAdes supports mate-pair only assembly. However, high-quality mate-pair libraries are recommended in these cases. Here, regular mate-pair libraries can be provided as supplementary information. Upstream and downstream files will be distinguished using the pattern established on the previous page (Paired-end Configuration).
- **Use Data for Hybrid Assembly:**
- PacBio (CLR), Oxford Nanopore, and Sanger reads can be provided for hybrid assemblies (e.g. with Illumina or IonTorrent data). SPAdes uses this data for gap closure and repeat resolution.
- Contigs of the same genome (trusted) generated by other assemblers can be specified to merge them into SPAdes assembly.
- Less reliable contigs (untrusted) can be used only for gap closure and repeat resolution.

Only contigs of the same genome should be specified since SPAdes does not work with genomes of closely related species.

Figure 2: *Input Page 2*

Configuration

- **Automatic K-mer Sizes:** K-mer sizes are selected automatically based on the read length and data set type:
- If single-cell data is provided, the default values are 21, 33 and 55.
- For multicell datasets, K values are automatically selected using maximum read length.
- **K-mer Sizes:** Define a comma-separated list of k-mer sizes to be used. These must be odd and less than 128. You can find recommendations about K-mer sizes in the SPAdes documentation.
- **Read Error Correction:** Performs a read error correction before assembly. Depending on the sequencing platform, the BayesHammer (Illumina) or the IonHammer (IonTorrent) tools are used for this task. This procedure is recommended to obtain high-quality assemblies but can be turned off if read error correction has been done previously.
- **Mismatch Careful Mode:** Tries to reduce the number of mismatches and short indels. It also runs MismactCorrector, a post-processing tool that uses BWA.

This option is recommended only for the assembly of small genomes. For large and medium-size eukaryotic genomes is not recommended.

- **Read Coverage Cutoff:** Configure the read coverage cutoff value that SPAdes will use to obtain the most reliable assembled sequences. Must be a positive decimal number, or automatic, or off. When set to "Automatic" SPAdes automatically computes coverage threshold using conservative strategy.
- **Read Coverage Cutoff Value:** If the "Defined by User" option is selected above, set a positive float value.

Figure 3: *Configuration Page*

Output

- **Contigs Fasta:** Where to store the Fasta file containing the assembled contigs.
- **Scaffolds Fasta:** Where to store the Fasta file containing the assembled scaffold. Recommended for use as resulting sequences.
- **Save Graph Files:** Save the final graphs in a .gfa file. Two files are generated: assembly_graph_after_simplification.gfa and assembly_graph_with-scaffolds.gfa.
- **Graph Files:** Select a folder to store the graph files (.gfa).

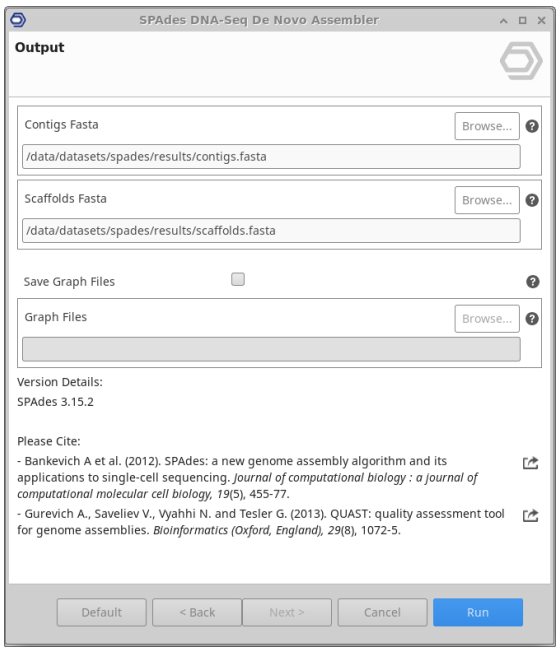


Figure 4: *Output Page*

RESULTS

SPAdes returns the assembled sequences in two FASTA files:

- **Contigs**: Contains resulting contigs.
- **Scaffolds**: Contains resulting scaffolds (recommended for use as resulting sequences).

Contigs/scaffolds names in SPAdes output FASTA files have the following format:

> NODE_3_length_237403_cov_243.207

- 3 is the number of the contig/scaffold.
- 237403 is the sequence length in nucleotides.
- 243.207 is the k-mer coverage for the last (largest) k value used. Note that the k-mer coverage is always lower than the read (per-base) coverage.

If the "Save Graph Files" option is checked, SPAdes returns the assembly graph and scaffolds paths in GFA 1.0 format. The "assembly_graph_after_simplification.gfa" file correspond to contigs before repeat resolution (edges of the assembly graph). Paths corresponding to contigs after repeat resolution (scaffolding) are stored in "assembly_graph_with-scaffolds.gfa".

To view GFA files, the Bandage visualization tool is recommended.

In addition to the resulting FASTA files, a report and a chart are generated. The report shows a summary of the DNA-Seq *De Novo* Assembly results (Figure 5). This page contains information about the input sequencing data and a results overview. The Results Overview table shows a number of common statistics used to describe the quality of a sequence assembly (see the explanation in the previous section).

- **N50:** This statistic defines the assembly quality in terms of contiguity. N50 is calculated by first ordering every contig or scaffold from the longest to the shortest. Next, starting from the longest sequence, the lengths of each sequence are summed up, until this running sum equals one-half of the total length of all sequences in the assembly. The N50 of the assembly is the length of the shortest contig in this list. Higher values of N50 indicate a better assembly. Note that any Nx statistic is calculated in the same way, e.g. N75 is calculated summing up all the lengths until the sum equals 75% of the total length.
- **L50:** Defined as the smallest number of contigs whose lengths sum makes up half of the total assembly length.

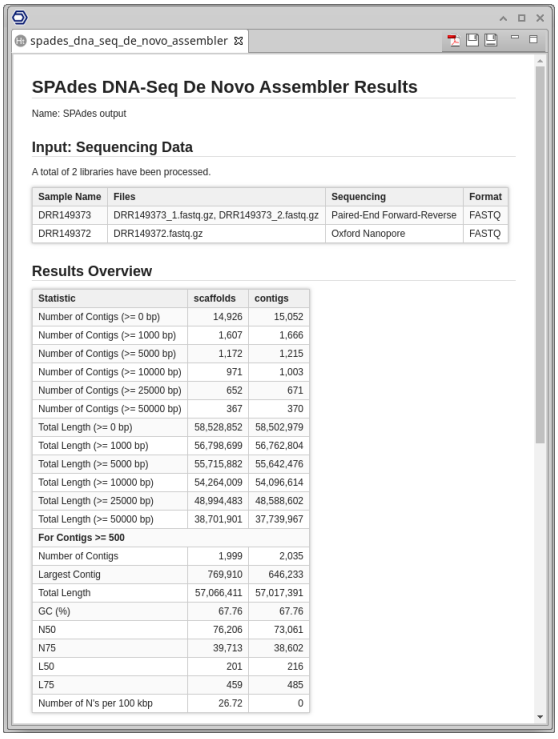


Figure 5: *Summary Report*

The Nx plot (Figure 6) shows Nx values as x varies from 0 to 100 %. The Nx values are displayed for contigs and scaffolds.

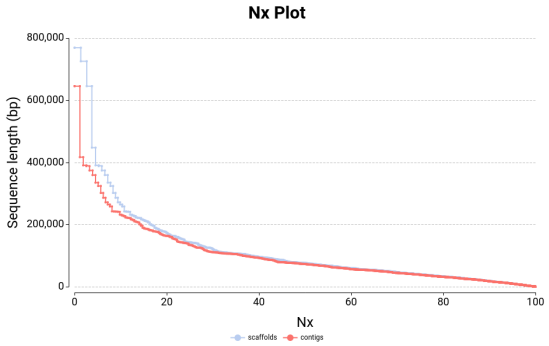


Figure 6: *Nx Plot*

Flye

INTRODUCTION

Flye is a long-read assembly algorithm that generates arbitrary paths in an unknown repeat graph, called disjointigs, and constructs an accurate repeat graph from these error-riddled disjointigs:

1. Flye initially generates disjointigs that represent concatenations of multiple disjoint genomic segments
2. Concatenates all error-prone disjointigs into a single string (in arbitrary order).
3. Constructs an accurate assembly graph from the resulting concatenate.
4. Uses reads to untangle this graph and resolves bridged repeats.
5. Resolves bridged repeats (which are bridged by some reads in the repeat graph).
6. Uses the repeat graph to resolve unbridged repeats (which are not bridged by any reads) using small differences between repeat copies.
7. Output accurate contigs formed by paths in this graph.

Please, cite Flye as:

- Kolmogorov M., Yuan J., Lin Y. and Pevzner PA. (2019). Assembly of long, error-prone reads using repeat graphs. Nature Biotechnology, 37(5), 540-546.

RUN FLYE ASSEMBLY

This functionality can be found under **Genome Analysis → DNA-Seq*de novo*Assembly → Flye**. The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, and Figure 3).

Input

- **Input Reads:** Select the files containing the sequencing libraries (long reads). Currently, PacBio (raw, corrected, HiFi) and ONT reads (raw, corrected) are supported. Expected error rates are <30% for raw, <3% for corrected reads, and <1% for HiFi.

Mixing different read types is not yet supported.

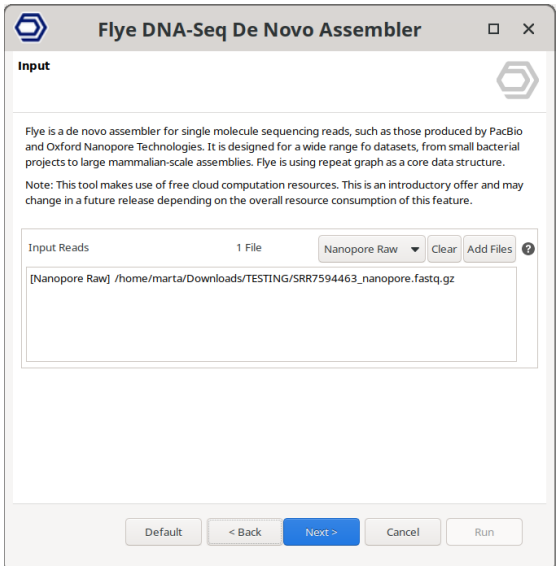


Figure 1: *Input Page*

Configuration

- **Reduce RAM Consumption:** For high coverage datasets, reduce the memory usage by using only a subset of longest reads for initial disjointig extension stage (usually the memory bottleneck). All reads will be used at the later pipeline stages (e.g. for repeat resolution). Enabling this option requires specifying the following parameters:
  - **Estimated Genome Size:** Specify the estimated genome size. The letters 'k', 'm', or 'g' could be included to represent kilobases, megabases, and gigabases. For example, 5m or 2.6g.
  - **Target Coverage:** Specify the target coverage for initial disjointig assembly. The longest reads will be used until matching the specified coverage. Tipically, a coverage of 40 is enough to produce good disjointigs.
- **Automatic Minimum Overlap:** The minimum overlap length for two reads to be considered overlapping is chosen automatically based on the read length distribution (reads N90) and does not require a manual setting.
- **Manual Minimap Overlap:** This sets a minimum overlap length for two reads to be considered overlapping. The typical value is 3k-5k. Intuitively, we want to set this parameter as high as possible, so the repeat graph is less tangled. However, higher values might lead to assembly gaps. In some rare cases (for example in the case of biased read length distribution) it makes sense to set this parameter manually.

- **Polishing:** Polishing is performed as the final assembly stage, with the aim of correcting errors. By default, Flye runs one polishing iteration.
- **Number of Polishing Iterations:** Additional iterations might correct a small number of extra errors (due to improvements on how reads may align to the correct assembly).
- **Plasmids:** This option allows to rescue short unassembled plasmids.
- **Keep Haplotypes:** Do not collapse alternative haplotypes.

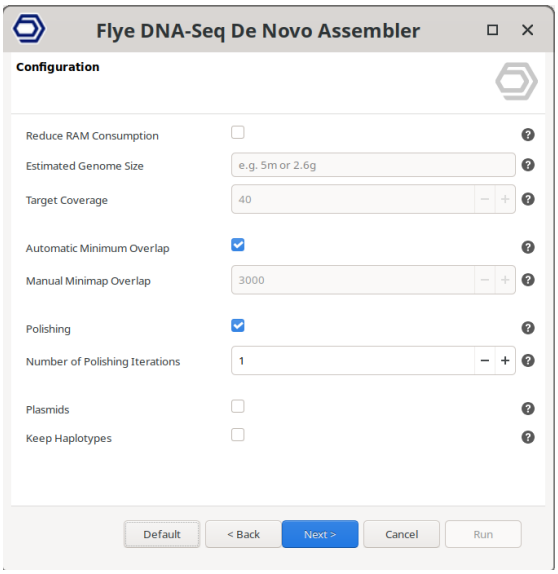


Figure 2: *Configuration Page*

Output

- **Assembly Fasta:** Select where to store the Fasta file containing the assembled genomic sequences.
- **Save Graph File:** Save the final repeat graph in a .gfa file.
- **Graph File:** Where to store the Gfa file containing the final repeat graph created by Flye.

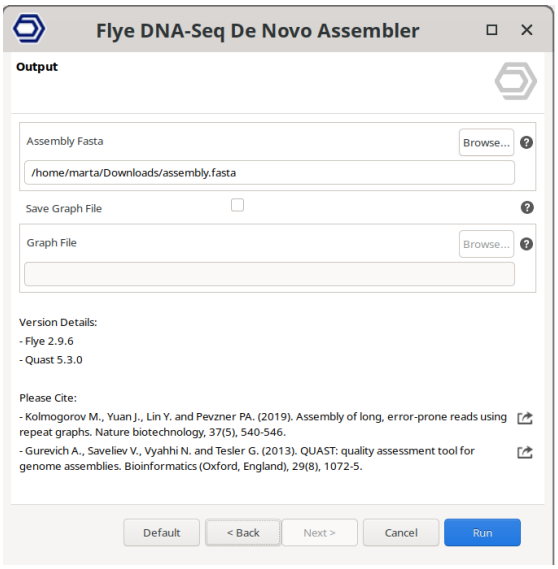


Figure 3: *Output Page*

RESULTS

Flye returns the results in two different files:

- **Assembly (Fasta)**: Contains resulting contigs/scaffolds.
- **Assembly Graph (Gfa):** Final repeat graph. Note that the edge sequences might be different (shorter) than contig sequences because contigs might include multiple graph edges.

Repeat graphs produced by Flye could be visualized using AGB or Bandage.

In addition to the resulting files, a report and a chart are generated. The report shows a summary of the DNA-Seq *De Novo* Assembly results (Figure 4). This page contains information about the input sequencing data and a results overview. The Results Overview table shows a number of common statistics used to describe the quality of a sequence assembly (see the explanation in the previous section).

- **N50:** This statistic defines the assembly quality in terms of contiguity. N50 is calculated by first ordering every contig or scaffold from the longest to the shortest. Next, starting from the longest sequence, the lengths of each sequence are summed up, until this running sum equals one-half of the total length of all sequences in the assembly. The N50 of the assembly is the length of the shortest contig in this list. Higher values of N50 indicate a better assembly. Note that any Nx statistic is calculated in the same way, e.g. N75 is calculated summing up all the lengths until the sum equals 75% of the total length.
- **L50:** Defined as the smallest number of contigs whose lengths sum makes up half of the total assembly length.

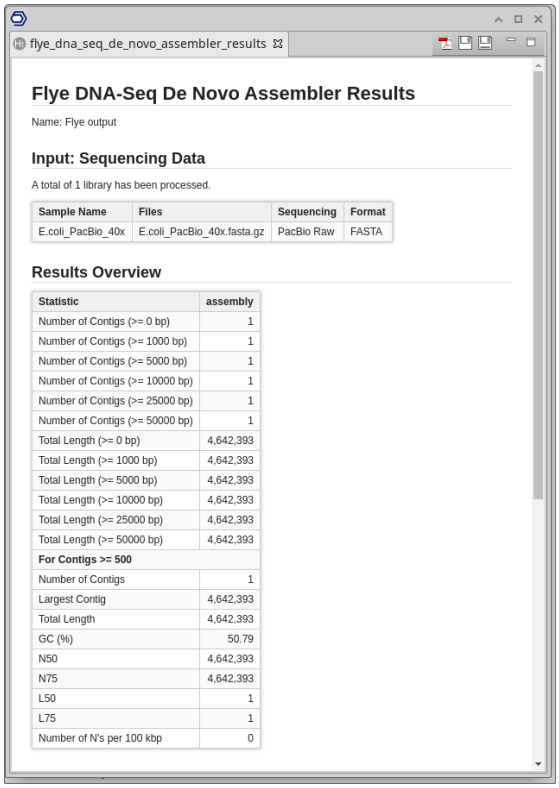


Figure 4: *Summary Report*

The Nx plot (Figure 5) shows Nx values as x varies from 0 to 100 %. The Nx values are displayed for contigs and scaffolds.

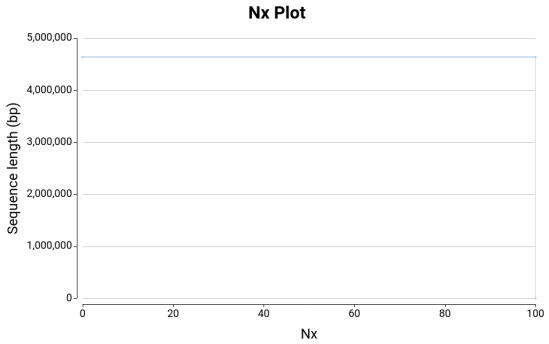


Figure 5: *Nx Plot*

4.2.3 DNA-Seq Alignment

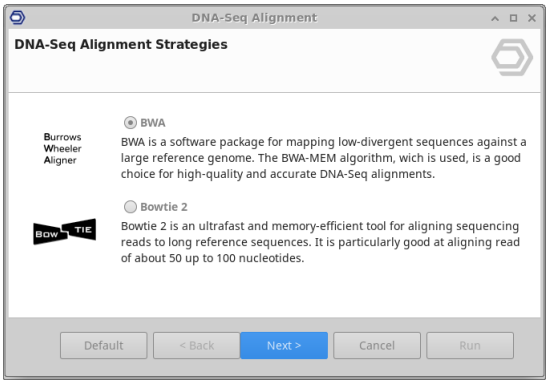
DNA-Seq Alignment

Read alignment is a common process applied to high-throughput sequencing data, being one of the first stages required for many different types of analysis. In the DNA-Seq scenario, this process is applied for variant calling and before the polishing procedure. The goal of the read alignment is to map short sequencing reads efficiently to a large reference genome to identify the 'correct' genomic loci from which the read originated whilst taking into account errors in the sequence reads.

This functionality can be found under **Genome Analysis → DNA-Seq Alignment**.

Two alignment strategies are available:

- **BWA**: BWA is a software package for mapping log-divergent sequences against a large reference genome. The BWA-MEM algorithm, which is used, is a good choice for high-quality and accurate DNA-Seq alignments.
- **Bowtie 2**: Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning readings of about 50 up to 100 nucleotides.



DNA-Seq BWA

INTRODUCTION

The Burrows-Wheeler Alignment algorithm (BWA) is a read alignment package that is based on a backward search with Burrows-Wheeler Transform (BWT), to efficiently align short sequencing reads against a large reference sequence such as the human genome, allowing mismatches and gaps. BWA supports both base space reads, e.g. from Illumina sequencing machines, and color space reads from AB SOLiD machines. The BWA-MEM algorithm is used, which performs local alignment. It may produce multiple primary alignments for different parts of a query sequence.

Please cite BWA as:

Li H. and Durbin R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics (Oxford, England), 25(14), 1754-60.

RUN DNA/RNA-SEQ ALIGNMENT (BWA)

This functionality can be found under **Genome Analysis → DNA-Seq Alignment → BWA**. The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

Input

- **Input Reads:** Select the files containing sequencing reads. These files are assumed to be in FASTQ/FASTA format. Both, single and paired-end data are accepted.
- **Paired-end Configuration:** If paired-end reads are provided, a pattern to distinguish upstream files from downstream files is required. The provided patterns are searched in the filenames right before the extension. The beginning of the filenames should be the same for both files of each sample.
- **Upstream Files Pattern**: Establish the pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern**: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is SRR037717_1.fastq and the downstream SRR037717_2.fastq, "_1" should be established as the upstream pattern and "_2" as the downstream pattern.

- **Reference Genome:** Specify a FASTA file with the genome reference sequences. Multiple reference sequences (e.g. chromosomes or scaffolds) are allowed.

It is not recommended to provide masked genome sequences since the algorithm will force those reads that originate in repeats to map (falsely) somewhere else in the genome.

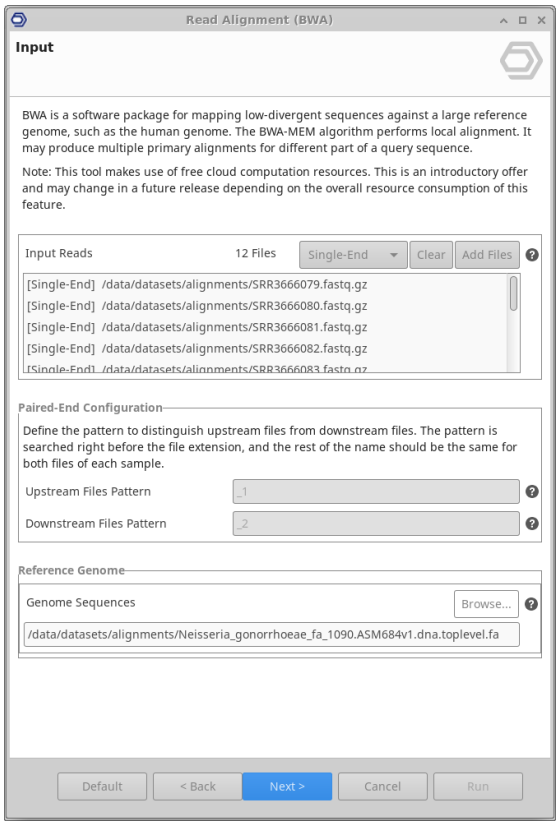


Figure 1: *Input Page*

Algorithm Options

- **Minimum Seed Length:** Matches shorter than this value will be missed. The alignment speed is usually intensive to this value unless it significantly deviates 20.

- **Band Width:** Essentially, baps longer than this value will not be found. Note that the maximum gap length is also affected by the scoring matrix and the hit length, not solely determined by this option.
- **Z-dropoff:** Also known as Off-diagonal X-dropoff. Stop extension when the difference between the best and the current extension score is above |i-j|*A+Z-dropoff, where *i* and *j* are the current positions of the query and reference, respectively, and A is the matching score. Z-dropoff is similar to BLAST's X-dropoff except that it does not penalize gaps in one of the sequences in the alignment. Z-dropoff not only avoids unnecessary extension but also reduces poor alignments inside a long good alignment.
- **Trigger Re-seeding:** Look for internal seeds inside a seed longer than {Minimum Seed Length} * Trigger Re-seeding. This is a key heuristic parameter for tuning the performance. Larger values yield fewer seeds, which leads to faster alignment speed but lower accuracy.
- **Seed Occurrence:** Seed occurrence for the 3rd round seeding.
- **Skip Seeds:** Discard a seed if it has more than this number of occurrences in the genome.
- **Drop Chains:** Drop chains shorter than this fraction of the longest overlapping chain.
- **Discard Chains:** Discard a chain if seeded bases shorter than this value.
- **Mate Rescue Rounds:** Perform at most this number of rounds of mate rescues for each read.
- **Skip Mate Rescue:** Skip the mate rescue procedure.
- **Skip Pairing:** In the paired-end mode, perform SW to rescue missing hits only but do not try to find hits that fit a proper pair. The mate rescue is performed, unless the Skip Mate Rescue option is also in use.

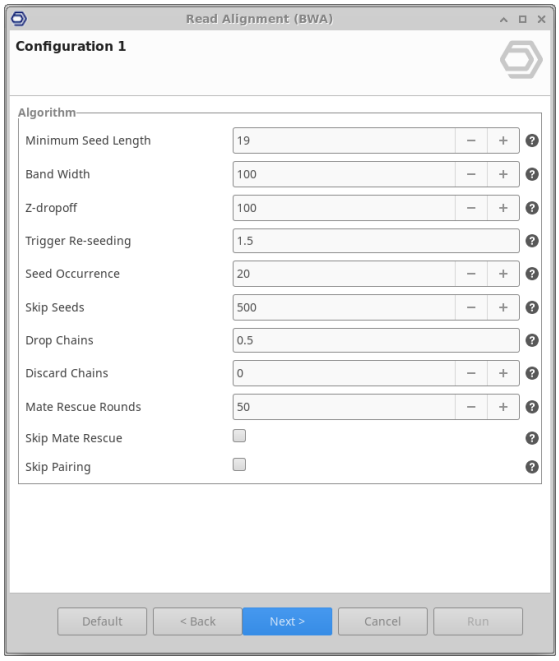


Figure 2: *Configuration Page 1*

Scoring Options

- **Matching Score:** Score for a sequence match.
- **Mismatch Penalty:** Penalty for a mismatch. The sequence error rate is approximately: {.75 * exp[-log(4) * Mismatch Penalty/Matching Score]}.
- **Gap Open Penalty for Deletions:** Gap open penalty for deletions.
- **Gap Open Penalty for Insertions:** Gap open penalty for insertions.
- **Gap Extension Penalty for Deletions:** Gap extension penalty for deletions. A gap of size k cost '{-O} + {-'Gap Extension Penalty'}*k'.
- **Gap Extension Penalty for Insertions:** Gap extension penalty for insertions. A gap of size k cost '{-O} + {-'Gap Extension Penalty'}*k'.
- **5'-end Clipping Penalty:** Penalty for 5'-end clipping. When performing SW extension, BWA-MEM keeps track of the best score reaching the end of the query. If this score is larger than the best SW score minus the clipping penalty, clipping will not be applied. Note that in this case, the SAM AS tag reports the best SW score; the clipping penalty is not deducted.
- **3'-end Clipping Penalty:** Penalty for 3'-end clipping.
- **Unpaired Read Penalty:** Penalty for an unpaired read pair.

Output Options

- **Minimum Score:** Minimum score to output.
- **Mark Split Alignments as Primary:** For split alignment, take the alignment with the smallest coordinate as primary.
- **Not Modify mapQ of Supp. Alignments:** Do not modify the mapping quality of supplementary alignments.

- **Output All SE/Unpaired PE Alignments:** Output all alignments for Single-End or unpaired Paired-End reads.
- **Soft Clipping for Supplementary:** Use soft clipping for supplementary alignments.
- **Mark Shorter Split Hits as Secondary:** Mark shorter split hits as secondary.
- **Sort BAM File:** Establish how output BAM files should be sorted.
- **Add Read Group Information:** Include the 'Read Group' header (@RG) in output BAM files. This information may be required for downstream analysis of third-party tools. If this option is checked, the following read group tags will be included for each sample:
- Identifier (ID), automatically generated.
- The name of the sample (SM), inferred from file names.
- Sequencing Platform (PL), provided by the user.
- **Sequencing Platform:** Choose the sequencing platform which was used to obtain the input data. Consider that if this option is provided, all output BAMs will be tagged with the same platform.

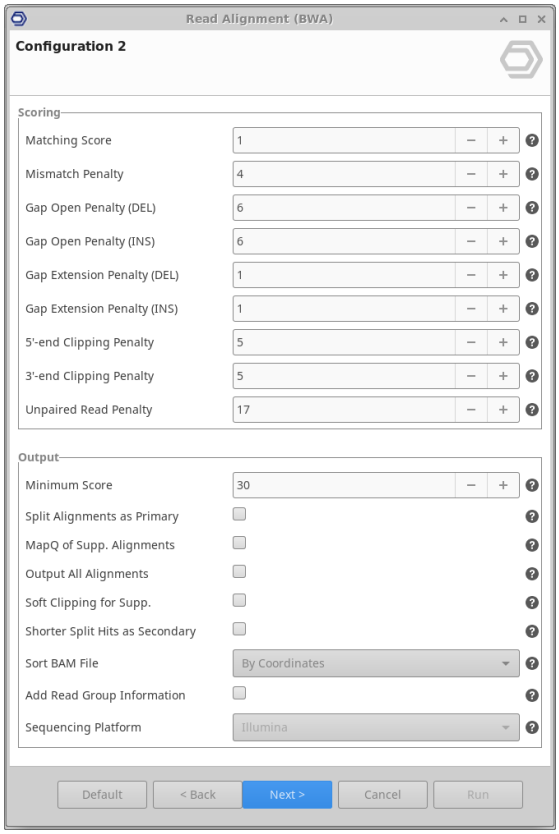


Figure 3: *Configuration Page 2*

Output

- **Alignment Files:** Select a destination folder to save output BAM files.

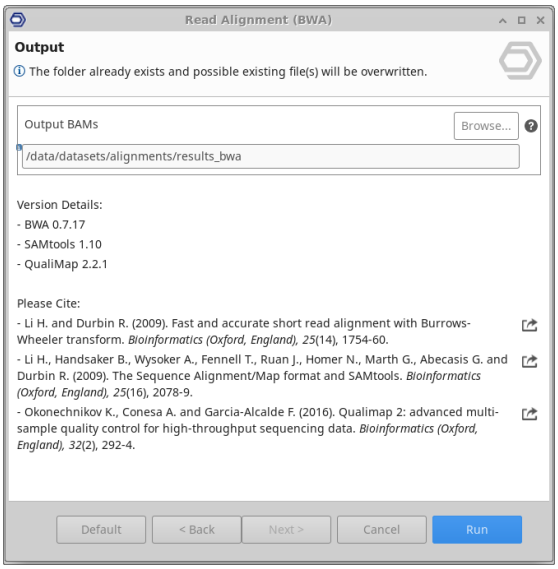


Figure 4: *Output Page*

RESULTS

The main outputs are the BAM files. A BAM file (*.bam) is a compressed binary version (BGZF format) of a SAM file that is used to represent aligned sequences. SAM is a TAB-delimited text format consisting of a header section and an alignment section. Header lines start with '@', while alignment lines do not. Each alignment line has

11 mandatory fields for essential alignment information such as the mapping position, and a variable number of optional fields for flexible or aligner-specific information.

SAM Format Description

1. **QNAME**: Query template (read) name.In a SAM file, a read may occupy multiple alignment lines, when its alignment is chimeric or when multiple mappings are given.
2. **FLAG**: SAM flags summarize many properties of reads, represented by flag bits, into a single number:
3. Read is paired.
4. Read is mapped in a proper pair.
5. Read is unmapped.
6. Mate is unmapped.
7. Read reverse strand.
8. Mate reverse strand.
9. Read is from the first pair.
10. Read is from the second pair.
11. Alignment isn't primary.
12. Read fails platform/vendor quality checks.
13. Read is PCR or optical duplicate.
14. **RNAME**: Reference sequence name. If @SQ header lines are present, RNAME must be present in one of the SQ-SN tag.
15. **POS**: 1-based leftmost mapping position of the first CIGAR operation. The first base in a reference sequence has coordinate 1.
16. **MAPQ**: Mapping quality. It equals −10 log10 Pr{mapping position is wrong}, rounded to the nearest integer. A value 255 indicates that the mapping quality is not available.
17. **CIGAR**: A string describing how the read aligns with the reference. It consists of one or more components. Each component comprises an operator and the number of bases which the operator applies to. Operators are:
18. M: Align match.
19. I: Insertion to the reference.
20. D: Deletion from the reference.
21. N: Skipped region from the reference.
22. S: Soft clipping.
23. H: Hard clipping.
24. P: Padding (silent deletion from padded reference).
25. =: Sequence match
26. X: Sequence mismatch
27. **RNEXT**: Reference sequence name of the primary alignment of the next read in the template. If all segments are mapped to the same reference, the unsigned observed template length equals the number of bases from the leftmost mapped base to the rightmost mapped base.
28. **PNEXT**: a 1-based position of the primary alignment of the next read in the template.
29. **TLEN**: Signed observed template length.
30. **SEQ**: Segment sequence.
31. **QUAL**: ASCII of base QUALity plus 33 (same as the quality string in the Sanger FASTQ format).

In addition to these 11 obligatory fields, optional fields may be included. All optional fields follow the TAG:TYPE:VALUE format where TAG is a two-character string.

For more information about the SAM format, visit the SAM Format Specification Page.

You can check the meaning of a FLAG number using the SAM Flag Translator.

In addition, a report and two charts are generated with complementary information. The report (Figure 5) shows a summary of the DNA-Seq Alignment results. This page contains information about the reference genome sequences, the input FASTQ files, and a results overview. The last section is divided into several subsections: globals, paired information, ACTG content, coverage, mapping quality, insert size, mismatches, and indels.

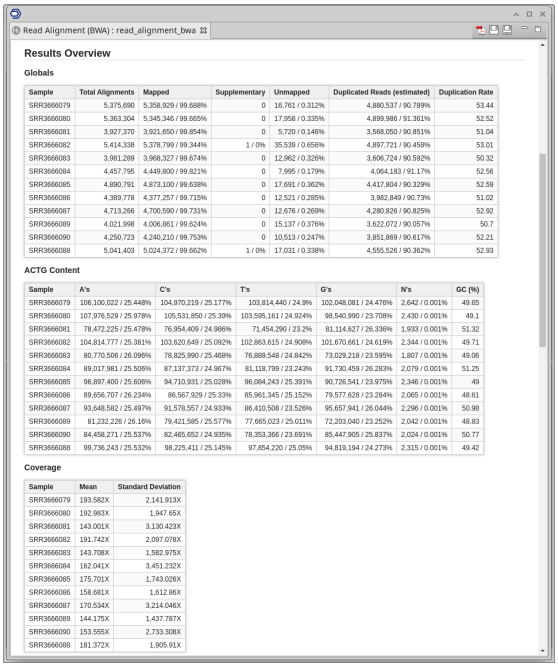


Figure 5: *Summary Report*

The bar charts (Figure 6) show the number of mapped and unmapped reads of each input file.

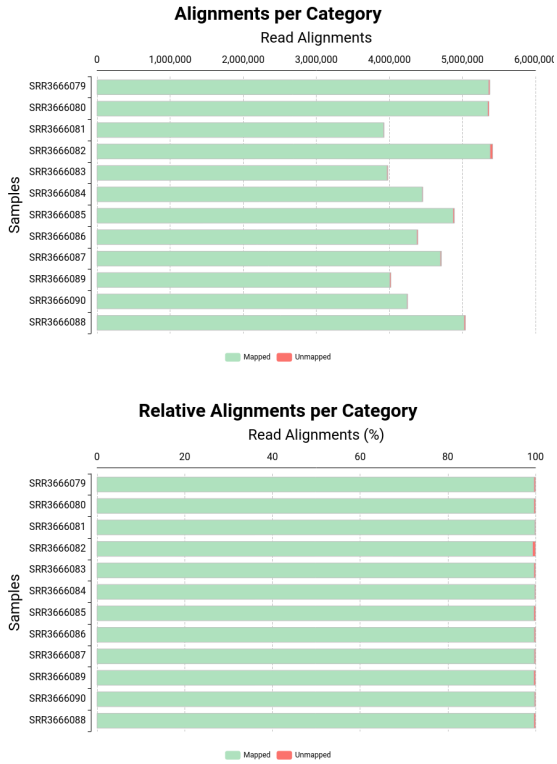


Figure 6: *Alignments per Category Charts*

Finally, the Genome Browser allows you to visualize genomic coordinates (GFF/GTF) in a side-scrolling way. Several tracks can be added to the browser, the currently supported tracks are VCF, DNA Fasta, and BAM. The BAM track (Figure 7) shows the reads of a BAM file and if the sequence track is active, it will also highlight the differences between the read sequence and the sequence track.

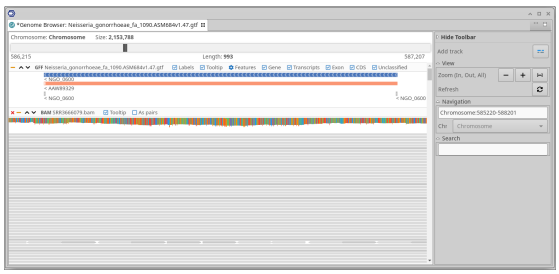


Figure 7: *Genome Browser*

DNA-Seq Bowtie 2

INTRODUCTION

Bowtie 2 is an ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. It is particularly good at aligning reads of about 50 up to 100s of characters to relatively long (e.g. mammalian) genomes. Bowtie 2 indexes the genome with an FM Index (based on the Burrows-Wheeler Transform or BWT) to keep its memory footprint small. This algorithm has some interesting features:

- Bowtie 2 supports gapped alignment with affine gap penalties.
- Bowtie 2 supports local alignment, which doesn't require reads to align end-to-end. Local alignments might be "trimmed" ("soft clipped") at one or both extremes in a way that optimizes alignment score.
- Bowtie 2 allows alignments to overlap ambiguous characters (e.g. Ns) in the reference.
- Bowtie 2's paired-end alignment is more flexible. E.g. for pairs that do not align in a paired fashion, Bowtie 2 attempts to find unpaired alignments for each mate.

Please cite Bowtie 2 as:

Langmead B. and Salzberg SL. (2012). Fast gapped-read alignment with Bowtie 2. Nature methods, 9(4), 357-9.

RUN DNA-SEQ ALIGNMENT (BOWTIE 2)

This functionality can be found under **Genome Analysis → DNA-Seq Alignment → Bowtie 2**. The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, Figure 3, Figure 4, and Figure 5).

Input

- **Input Reads:** Select the files containing sequencing reads. These files are assumed to be in FASTQ/FASTA format. Both, single and paired-end data are accepted.
- **Paired-end Configuration:** If paired-end reads are provided, a pattern to distinguish upstream files from downstream files is required. The provided patterns are searched in the filenames right before the extension. The beginning of the filenames should be the same for both files of each sample.
- **Upstream Files Pattern**: Establish the pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern**: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is SRR037717_1.fastq and the downstream SRR037717_2.fastq, "_1" should be established as the upstream pattern and "_2" as the downstream pattern.

- **Reference Genome:** Specify a FASTA file with the genome reference sequences. Multiple reference sequences (e.g. chromosomes or scaffolds) are allowed.

It is not recommended to provide masked genome sequences since the algorithm will force those reads that originate in repeats to map (falsely) somewhere else in the genome.

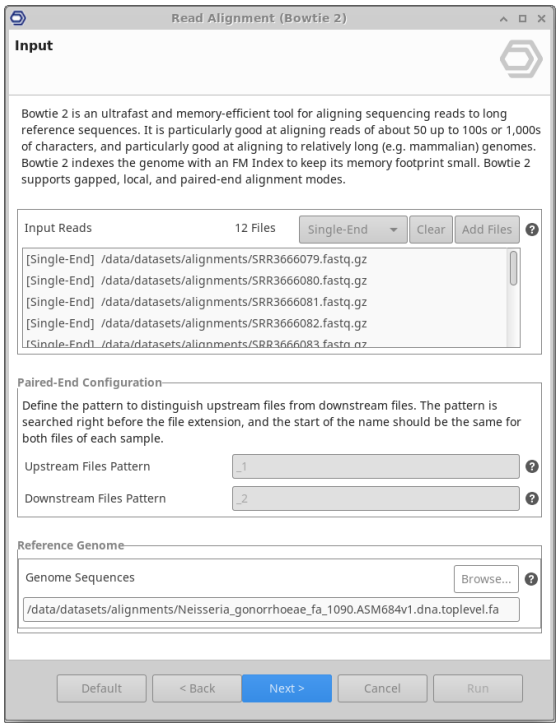


Figure 1: *Input Page*

General Options

- **Parameter Preset:** Bowtie 2 comes with some useful combinations of parameters packages into shorter 'preset' parameters. The preset options are designed to cover a wide area of the speed/sensitivity/accuracy trade-off space, with the presets ending in 'fast' generally being faster but less sensitive and less accurate, and the presets ending in sensitive generally being slower but more sensitive and more accurate.

Selecting a preset will overwrite most of the parameters. These parameters can be readjusted, even if a preset has been selected.

- **Alignment:**
- **End to End:** By default, Bowtie 2 performs end-to-end read alignments. It searches for alignments involving all of the read characters. It is also known as 'untrimmed' or 'unclipped' alignment.
- **Local:** In this mode, Bowtie 2 might 'trim' or 'clip' read characters from one or both ends of the alignment if doing so maximizes the alignment score.

Alignment Options

- **Max # Mismatches:** Sets the number of mismatches allowed in a seed alignment during multiseed alignment. It can be set to 0 or 1. Setting this higher makes alignment slower (often much slower) but increases sensitivity.
- **Length of Seed Substrings:** Sets the length of the seed substrings to align during multiseed alignment. Smaller values make alignment slower but more sensitive.
- **Interval Between Seed Substrings:** Sets a function governing the interval between seed substrings to use during multiseed alignment. Since it's best to use longer intervals for longer reads, this parameter sets the interval as a function of the read length, rather than a single one-size-fits-all number. For instance, specifying "S, 1,1.15" sets the interval function f to f(x) = 1 + 1.15* sqrt(x), where x is the read length.

To rapidly narrow the number of possible alignments that must be considered, Bowtie 2 begins by extracting substrings ("seeds") from the read and its reverse complement and aligning them in an ungapped fashion with the help of the FM Index. This is 'multiseed alignment'.

- **Max # of non-A/C/G/T:** Sets a function governing the maximum number of ambiguous characters (usually Ns and/or .s) allowed in a read as a function of reading length. Reads exceeding this ceiling are filtered out. For instance, specifying -L,0,0.15 sets the N-ceiling function f to f(x) = 0 + 0.15 * x, where x is the read length.
- **Include DP:** 'Pads' dynamic programming problems by this number of columns on either side to allow gaps.
- **Disallow Gaps at Tips:** Disallows gaps within this number of positions at the beginning or end of the read.
- **Ignore Qualities:** When calculating a mismatch penalty, always consider the quality value at the mismatched position to be the highest possible, regardless of the actual value.
- **Do not Align Forward:** If specified, Bowtie 2 will not attempt to align reads to the forward (Watson) reference strand.
- **Do not Align Reverse-Complement:** If specified, Bowtie 2 will not attempt to align reads against the reverse-complement (Crick) reference strand.
- **Do not Allow 1 Upfront Mismatch:** By default, Bowtie 2 will attempt to find either an exact or a 1-mismatch end-to-end alignment for the read before trying the multiseed heuristic. This option prevents Bowtie 2 from searching for 1-mismatch end-to-end alignments before using the multiseed heuristic.

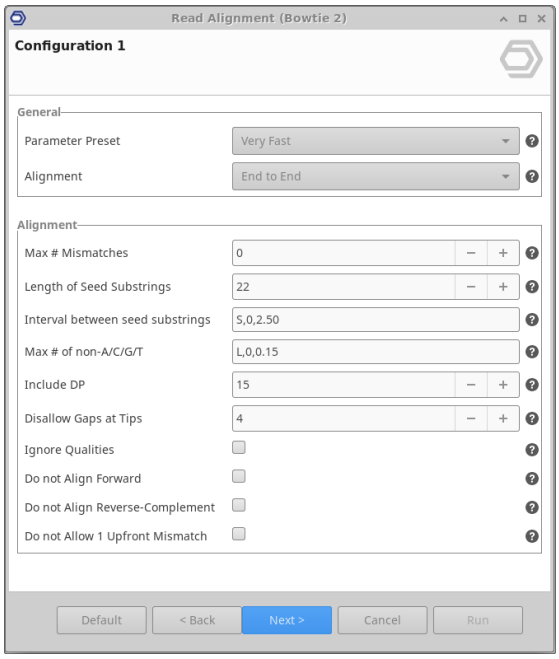


Figure 2: *Configuration Page 1*

Scoring Options

- **Match Bonus:** Sets the match bonus. In local mode, this value is added to the alignment score for each position where a read aligns to a reference character and the characters match. This parameter is not used in the 'End to End' mode.
- **Max/Min Penalty:** Sets the maximum and minimum mismatch penalties, both integers.

- **Penalty for non-A/C/G/Ts in Ref:** Sets penalty for positions where the read, reference, or both, contain an ambiguous character such as N.
- **Read Gap Open:** Sets the read gap open (first value) and extend (second value) penalties.
- **Reference Gap Open:** Sets the reference gap open (first value) and extend (second value) penalties.
- **Min Acceptable Align. Score:** Sets a function governing the minimum alignment score needed for an alignment to be considered valid. This is a function of read length. For instance, specifying L, 0, -0.6 sets the minimum-score function f to f(x)=0+-0.6*x, where x is the read length.

Reporting Options

- **Reporting:** Bowtie 2 searches for distinct, valid alignments for each read. The way in which they are reported can be adjusted:
- **Default**: The best alignment found is reported (the best mapping quality).
- **Report up to X**: Report up to X alignments per read. For reads that have more than X distinct, valid alignments, Bowtie 2 does not guarantee that the X alignments reported are the best possible in terms of alignment score.
- **Report all alignments**: There is no upper limit on the number of alignments to search for. This mode can be very slow in repetitive genomes.
- **Report up to:** When the 'Report up to X' mode is used, Bowtie 2 behaves differently. It searches for at most X distinct, valid alignments for each read. The search terminates when it can't find more distinct valid alignments, or when it finds X, whichever happens first. All alignments found are reported in descending order by alignment score. Each reported alignment beyond the first has the SAM 'secondary' bit (which equals 256) set in its FLAGS field.

For reads that have more than X distinct, valid alignments, Bowtie 2 does not guarantee that the X alignments reported are the best possible in terms of alignment score.

Effort Options

- **Give Up Extending After:** Set a value up to which consecutive seed extension attempts can 'fail' before Bowtie 2 moves on, using the alignments found so far. A seed extension 'fails' if it does not yield a new best or a new second-best alignment.
- **Try Sets of Seeds:** Set the maximum number of times Bowtie 2 will 're-seed' reads with repetitive seeds. When 're-seeding', Bowtie 2 simply chooses a new set of reads (same length, same number of mismatches allowed) at different offsets and searches for more alignments. A read is considered to have repetitive seeds if the total number of seed hits divided by the number of seeds that aligned at least once is greater than 300.

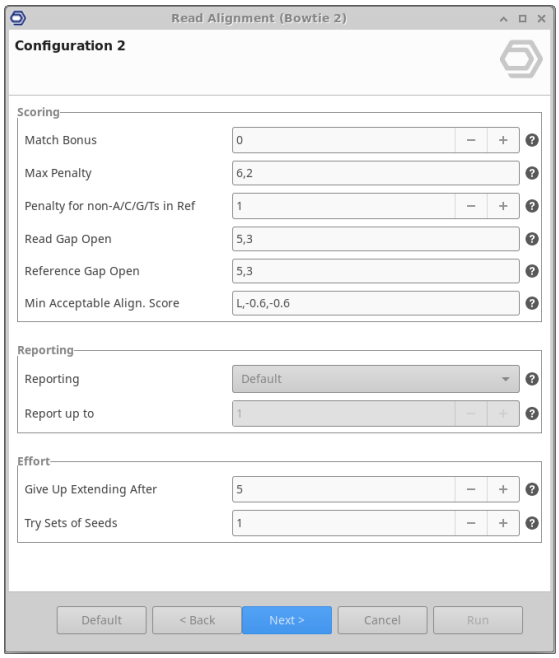


Figure 3: *Configuration Page 2*

Paired-end Specific Options

- **Minimum Fragment Length:** The minimum fragment length for paired-end alignments. E.g. if 60 is specified and a paired-end alignment consists of two 20 bp alignments in the appropriate orientation with a 20-bp gap between them, is considered valid. A 19-bp gap would not be valid in that case.
- **Maximum Fragment Length:** The maximum fragment length for valid paired-end alignments. E.g., if 100 is specified and a paired-end alignment consists of two 20-bp alignments in the proper orientation with a 60-bp gap between them, is considered valid. A 61-bp gap would not be valid in that case.
- **Read Order:** The upstream/downstream mate orientations for a valid paired-end alignment against the forward reference strand.
- **Forward / Reverse:** If there is a candidate paired-end alignment where mate 1 appears upstream of the reverse complement of mate 2, that alignment is valid. Also, if mate 2 appears upstream of the reverse complement of mate 1, is that too valid.
- **Reverse / Forward:** This mode requires that an upstream mate 1 be reverse-complemented and a downstream mate 2 be forward-oriented.
- **Forward / Forward:** This mode requires both, an upstream mate 1 and a downstream mate 2 to be forward-oriented.

- **No Mixed:** When Bowtie 2 cannot find a concordant or discordant alignment for a pair, it then tries to find alignments for the individual mates. This option disables that default behavior.
- **No Discordant:** Bowtie 2 looks for discordant alignments if it cannot find any concordant alignments. A discordant alignment is an alignment where both mates align uniquely, but that does not satisfy the paired-end constraints (Min and Max Fragment Length, and Read Order). This option disables that default behavior.
- **Dovetail:** If one mate alignment extends past the beginning of the other such that the wrong mate begins upstream, consider that to be concordant.
- **No Contain:** If one mate alignment contains the other, consider that to be non-concordant. By default, a mate can contain the other in a concordant alignment.
- **No Overlap:** If one mate alignment overlaps the other at all, consider that to be non-concordant. By default, mates can overlap in a concordant alignment.

Output Options

- **Add Read Group Information:** Include the 'Read Group' header (@RG) in output BAM files. This information may be required for downstream analysis of third-party tools. If this option is checked, the following read group tags will be included for each sample:
- Identifier (ID), automatically generated.
- The name of the sample (SM), inferred from file names.
- Sequencing Platform (PL), provided by the user.
- **Sequencing Platform:** Choose the sequencing platform which was used to obtain the input data. Consider that if this option is provided, all output BAMs will be tagged with the same platform.

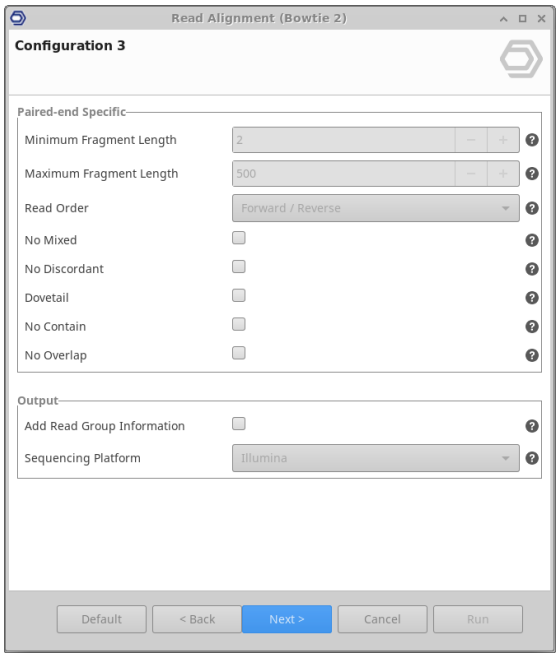


Figure 4: *Configuration Page 3*

Output

- **Alignment Files:** Select a destination folder to save output BAM files.

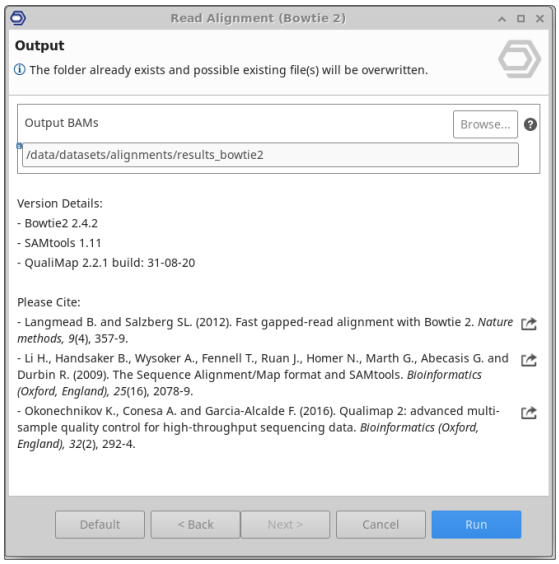


Figure 5: *Output Page*

RESULTS

The main outputs are the BAM files. A BAM file (*.bam) is a compressed binary version (BGZF format) of a SAM file that is used to represent aligned sequences. SAM is a TAB-delimited text format consisting of a header section and an alignment section. Header lines start with '@', while alignment lines do not. Each alignment line has

11 mandatory fields for essential alignment information such as the mapping position, and a variable number of optional fields for flexible or aligner-specific information.

SAM Format Description

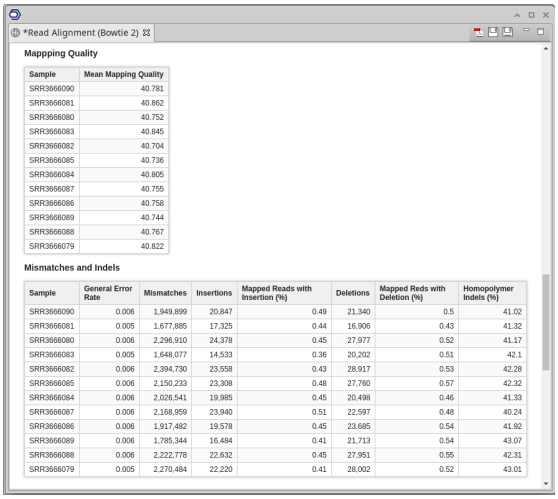
1. **QNAME**: Query template (read) name.In a SAM file, a read may occupy multiple alignment lines, when its alignment is chimeric or when multiple mappings are given.
2. **FLAG**: SAM flags summarize many properties of reads, represented by flag bits, into a single number:
3. Read is paired.
4. Read is mapped in a proper pair.
5. Read is unmapped.
6. Mate is unmapped.
7. Read reverse strand.
8. Mate reverse strand.
9. Read is from the first pair.
10. Read is from the second pair.
11. Alignment isn't primary.
12. Read fails platform/vendor quality checks.
13. Read is PCR or optical duplicate.
14. **RNAME**: Reference sequence name. If @SQ header lines are present, RNAME must be present in one of the SQ-SN tag.
15. **POS**: 1-based leftmost mapping position of the first CIGAR operation. The first base in a reference sequence has coordinate 1.
16. **MAPQ**: Mapping quality. It equals −10 log10 Pr{mapping position is wrong}, rounded to the nearest integer. A value 255 indicates that the mapping quality is not available.
17. **CIGAR**: A string describing how the read aligns with the reference. It consists of one or more components. Each component comprises an operator and the number of bases which the operator applies to. Operators are:
18. M: Align match.
19. I: Insertion to the reference.
20. D: Deletion from the reference.
21. N: Skipped region from the reference.
22. S: Soft clipping.
23. H: Hard clipping.
24. P: Padding (silent deletion from padded reference).
25. =: Sequence match
26. X: Sequence mismatch
27. **RNEXT**: Reference sequence name of the primary alignment of the next read in the template. If all segments are mapped to the same reference, the unsigned observed template length equals the number of bases from the leftmost mapped base to the rightmost mapped base.
28. **PNEXT**: a 1-based position of the primary alignment of the next read in the template.
29. **TLEN**: Signed observed template length.
30. **SEQ**: Segment sequence.
31. **QUAL**: ASCII of base QUALity plus 33 (same as the quality string in the Sanger FASTQ format).

In addition to these 11 obligatory fields, optional fields may be included. All optional fields follow the TAG:TYPE:VALUE format where TAG is a two-character string.

For more information about the SAM format, visit the SAM Format Specification Page.

You can check the meaning of a FLAG number using the SAM Flag Translator.

In addition, a report and two charts are generated with complementary information. The report (Figure 6) shows a summary of the DNA-Seq Alignment results. This page contains information about the reference genome sequences, the input FASTQ files, and a results overview. The last section is divided into several subsections: globals, paired information, ACTG content, coverage, mapping quality, insert size, mismatches, and indels.


*Read Alignment (Bowtie 2)

Mappping Quality

Sample	Mean Mapping Quality
SRR3666090	40.781
SRR3666081	40.862
SRR3666080	40.752
SRR3666083	40.845
SRR3666082	40.704
SRR3666085	40.736
SRR3666084	40.805
SRR3666087	40.755
SRR3666086	40.758
SRR3666089	40.744
SRR3666088	40.767
SRR3666079	40.822

Mismatches and Indels

Sample	General Error Rate	Mismatches	Insertions	Mapped Reads with Insertion (%)	Deletions	Mapped Reds with Deletion (%)	Homopolymer Indels (%)
SRR3666090	0.006	1,949,899	20,847	0.49	21,340	0.5	41.02
SRR3666081	0.005	1,677,885	17,325	0.44	16,906	0.43	41.32
SRR3666080	0.006	2,296,910	24,378	0.45	27,977	0.52	41.17
SRR3666083	0.005	1,648,077	14,533	0.36	20,202	0.51	42.1
SRR3666082	0.006	2,394,730	23,558	0.43	28,917	0.53	42.28
SRR3666085	0.006	2,150,233	23,308	0.48	27,760	0.57	42.32
SRR3666084	0.006	2,026,541	19,985	0.45	20,498	0.46	41.33
SRR3666087	0.006	2,168,959	23,940	0.51	22,597	0.48	40.24
SRR3666086	0.006	1,917,482	19,578	0.45	23,685	0.54	41.92
SRR3666089	0.006	1,785,344	16,484	0.41	21,713	0.54	43.07
SRR3666088	0.006	2,222,778	22,632	0.45	27,951	0.55	42.31
SRR3666079	0.005	2,270,484	22,220	0.41	28,002	0.52	43.01

Figure 6: *Summary Report*

The bar charts (Figure 7) show the number of mapped and unmapped reads of each input file.

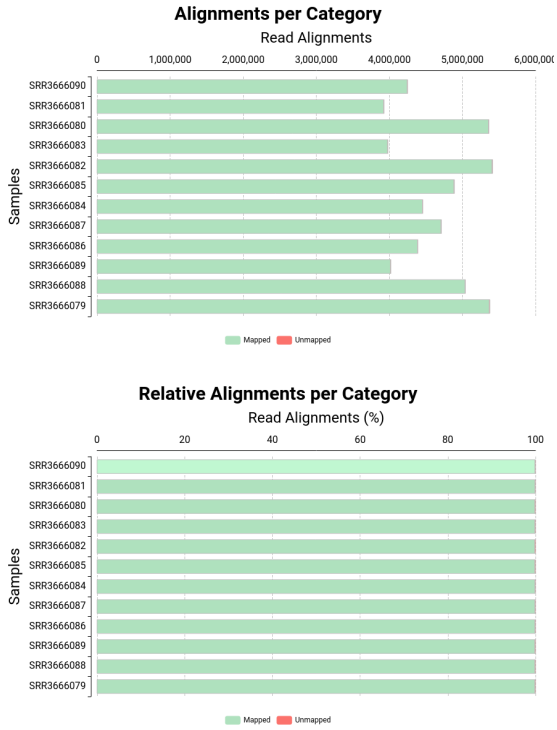


Figure 7: *Alignments per Category Charts*

Finally, the Genome Browser allows you to visualize genomic coordinates (GFF/GTF) in a side-scrolling way. Several tracks can be added to the browser, the currently supported tracks are VCF, DNA Fasta, and BAM. The BAM track (Figure 8) shows the reads of a BAM file and if the sequence track is active, it will also highlight the differences between the read sequence and the sequence track.

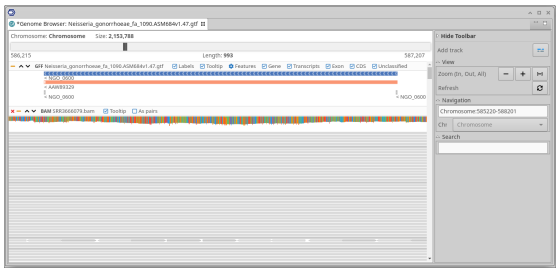


Figure 8: *Genome Browser*

4.2.4 DNA-Seq Polishing

Introduction

Third-generation DNA sequencing technologies allows scientist to generate longer sequence reads, which can be used in whole-genome sequencing projects to yield better repeat resolution and more contiguous genome assemblies. However, although long-read sequencing technologies can produce genomes with long contiguity, the relatively high error rate of long reads has made it challenging to generate a highly accurate final sequence. An effective strategy to generate highly contiguous assemblies with a very low overall error rate is to combine long reads with short-read data. This strategy can be pursued by using short reads to "polish" the consensus built from long reads.

The DNA-Seq Polishing application is based on Pilon. Pilon is a fully automated, all-in-one tool for correcting draft assemblies and calling sequence variants of multiple sizes, including very large insertions and deletions. Pilon works with many types of sequence data but is particularly strong when supplied with paired-end data from short-read libraries (e.g. Illumina). Pilon significantly improves draft genome assemblies by correcting bases, fixing misassemblies and filling gaps. For both, haploid and diploid genomes, Pilon produces more contiguous genomes with fewer errors, enabling the identification of more biologically relevant genes.

Pilon requires as input a FASTA file of the genome along with one or more BAM files of reads aligned to the input FASTA file. Pilon uses read alignment analysis to identify inconsistencies between the input genome and the evidence in the reads. It then attempts to make improvements to the input genome, including:

- Single base differences.
- Small indels.
- Larger indel or block substitution events.
- Gap filling.
- Identification of local misassemblies, including the optional opening of new gaps.

Pilon then outputs a FASTA file containing an improved representation of the genome from the read data.

Please cite Pilon as:

Walker BJ et al. (2014). Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. PloS one, 9(11), e112963.

Run DNA-Seq Polishing

This functionality can be found under **Genome Analysis → DNA-Seq Polishing.**The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

INPUT

- **Input Sequences:** Specify a FASTA file with the genome draft sequences to be polished. Multiple reference sequences are allowed (e.g. chromosomes or scaffolds).
- **Input BAMs:** Provide one or more BAM files of reads aligned to the input draft genome. BAM files can be obtained using the DNA-Seq Alignment functionality, to align short-reads to the draft genome.

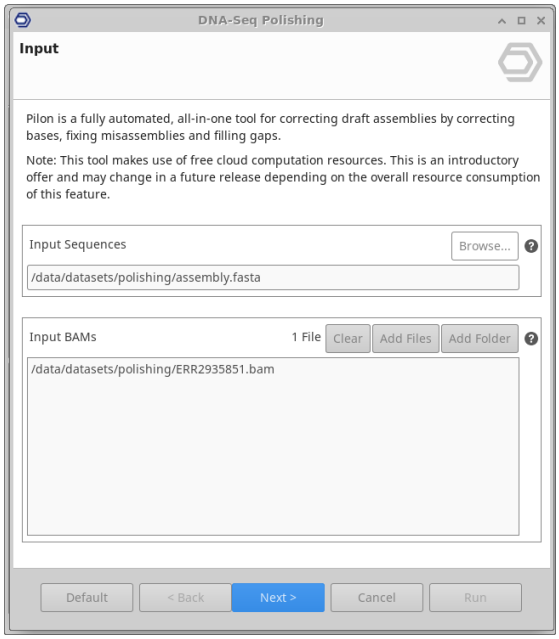


Figure 1: *Input Page*

CONTROL OPTIONS

- **Diploid:** Check this option if the sample is from a diploid organism. This will eventually affect the calling of heterozygous SNPs.
- **Issues to Fix:** Select the categories o issues to try to fix:
- SNPs: Try to fix individual base errors.
- Indels: Try to fix small indels.
- Gaps: Try to fill gaps.
- Local Misassemblies: Try to detect and fix local misassemblies.
- Ambiguous Bases*: Fix ambiguous bases to the most likely alternative.
- Breaks*: Allow local reassembly to open new gaps. Works with the "Local Missasemblies" category.
- Circular Elements*: Try to close circular elements when used with long corrected reads.
- Novel Sequence*: Assemble novel sequence from unaligned non-jump reads.

*Experimental fix types. By default, Pilon corrects for SNPs, indels, gaps and local misassemblies.

- **Duplicates:** Use reads marked as duplicates in the input BAMs (ignored by default).
- **IUPAC:** Pilon will use IUPAC nucleotide codes in the output FASTA file to represent ambiguous bases and/or heterozygous SNPs.
- **Failed Sequencer Quality:** Use reads which failed sequencer quality filtering (ignored by default).

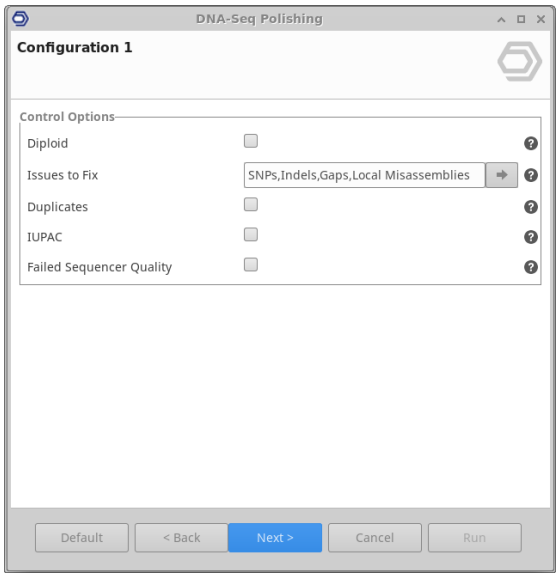


Figure 2: *Configuration Page 1*

HEURISTICS OPTIONS

- **Default Quality:** Assumes bases are of this quality if qualities are no present in input BAMs.
- **Flank:** Controls how much of the well-aligned reads will be used, this many bases at each end of the good reads will be ignored.
- **Gap Margin:** Closed gaps must be within this number of baes of true size to be closed.
- **K-mer Size:** K-mer size used by internal assembler.
- **Minimum Depth:** Variants (SNPs and indels) will only be called if there is coverage of good pairs at this depth or more. If this value is >= 1, it is an absolute depth. If it is a fraction < º, then minimum depth is computed by multiplying this value by the mean coverage for the region, with a minimum value of 5.

The default value is 0.1. This means that the depth to call is 10% of mean coverage or 5, whichever is greater.

- **Unclosed Gaps:** Minimum size of unclosed gaps.
- **Minimum Mapping Quality:** Minimum alignment mapping quality for a read to count in pileups.
- **Minimum Base Quality:** Minimum base quality to consider for pileups.
- **Skip Stray Pairs Identification:** Skip marking a pass through the input BAM files to identify stray pairs, that is, those pairs in which both reads are aligned but not marked valid because they have inconsistent orientation or separation. Identifying stray pairs can help fill gaps and assemble larger insertions, especially of repeat content.

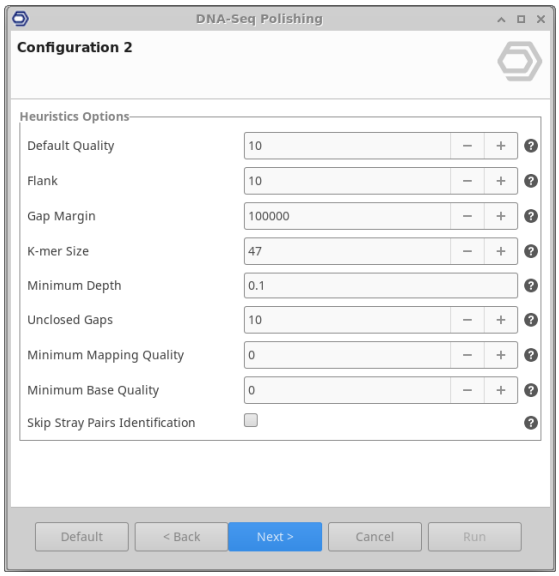


Figure 3: *Configuration Page 2*

OUTPUT

- **Output FASTA:** Select a file where the polished sequences will be placed.
- **Save Changes:** Pilon produces a file containing a space-delimited record of every change made in the assembly. Check this option to obtain this file.
- **Output Changes:** Select a file where the "changes" file will be placed. The format for the file is a follows: .

To improve performance, both input sequences and alignments are divided into 100MB batches.

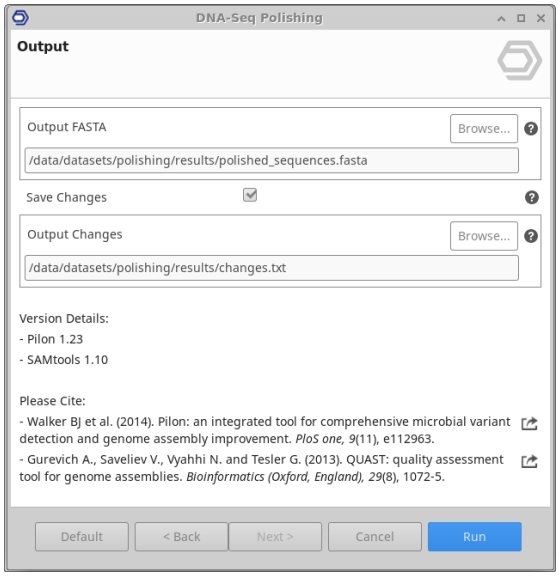


Figure 4: *Output Page*

Results

Pilon generates a FASTA file (polished_sequences.fasta), containing the improved genomic sequences. Pilon renames the sequence headers by appending "_pilon" to each FASTA element name. If the "Save Changes" option is checked, Pilon returns a text file reporting all changes applied to the input sequences. The format for this space-delimited file is a follows: .

```
## Deletion
contig_103:1825 contig_103_pilon:1825 T .
## Insertion
contig_103:233958 contig_103_pilon:233948 . C
## SNP
contig_103:364767 contig_103_pilon:364756 A G
## Segmental
contig_103:1054454-1054491 contig_103_pilon:1054403-1054440 CTAAATGGTAGTTGAGAATAGTGGCTACAAGAATTATA GTAAATGGTAGTTGAGAATAGTGGCTAACAGAATCATT
```

In addition to the resulting files, a report and 2 charts are generated. The report shows a summary of the DNA-Seq Polishing results (Figure 5). This page contains information about the input sequencing data and a results overview. The Results Overview section shows the number of each type of change that has been applied to the input sequences.

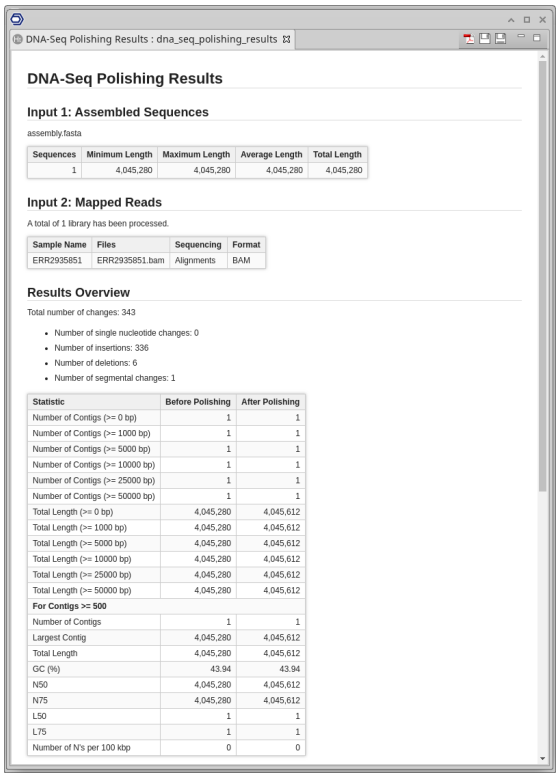


Figure 5: *Summary Report*

The Nx plot (Figure 6) shows Nx values as x varies from 0 to 100 %. The Nx values are displayed for contigs/scaffolds before and after polishing.

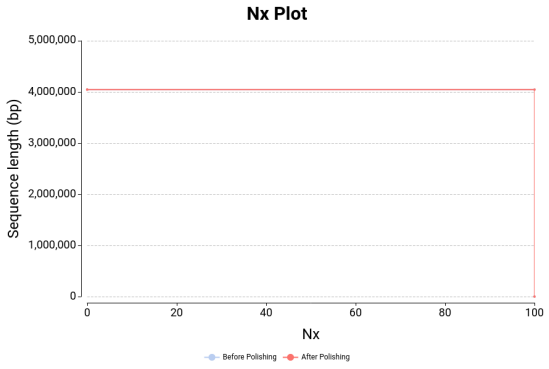


Figure 6: *Nx Plot*

The Fix Type Distribution Chart (Figure 7) displays the proportion of each type of change that has been applied to the input sequences.

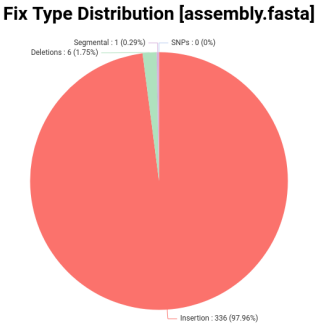


Figure 7: *Fix Type Distribution Chart*

4.2.5 Genome Completeness Assessment

Introduction

The Completeness Assessment functionality provides quantitative measures for the assessment of genome assembly completeness, based on evolutionarily-informed expectations of gene content from Benchmarking Universal Single-Copy Orthologs (BUSCO) selected from OrthoDB.

The Benchmarking Universal Single-Copy Orthologs are ideal for such quantifications of completeness, as the expectations for these genes to be found in a genome in single-copy are evolutionarily strong.

The application offers predefined BUSCO sets for six major phylogenetic clades. Sampling hundreds of genomes, orthologous groups with single-copy orthologs in >90% of species were selected. Importantly, this threshold accommodates the fact that even well-conserved genes can be lost in some lineages, as well as allowing for incomplete gene annotations and rare gene duplications.

OmicsBox offers predefined BUSCO datasets for six major phylogenetic clades:

- Bacteria
- Archaea
- Eukaryota
- Protists
- Fungi
- Plants

Please cite BUSCO and OrthoDB as:

Seppey M., Manni M. and Zdobnov EM. (2019). BUSCO: Assessing Genome Assembly and Annotation Completeness. Methods in molecular biology (Clifton, N.J.), 1962, 227-245.

Kriventseva EV., Kuznetsov D., Tegenfeldt F., Manni M., Dias R., Simao FA. and Zdobnov EM. (2019). OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial, and viral genomes for evolutionary and functional annotations of orthologs. Nucleic acids research, 47(D1), D807-D811.

Run Completeness Assessment

This functionality can be found under **genome analysis→ Completeness Assessment with Busco.**The wizard allows to select the input files and adjust the analysis parameters (Figure 1 and Figure 2).

INPUT

- **Input Sequences**: Select the input file to be analyzed. Either a nucleotide FASTA file or a protein FASTA file (depending on the mode selected on the next page).

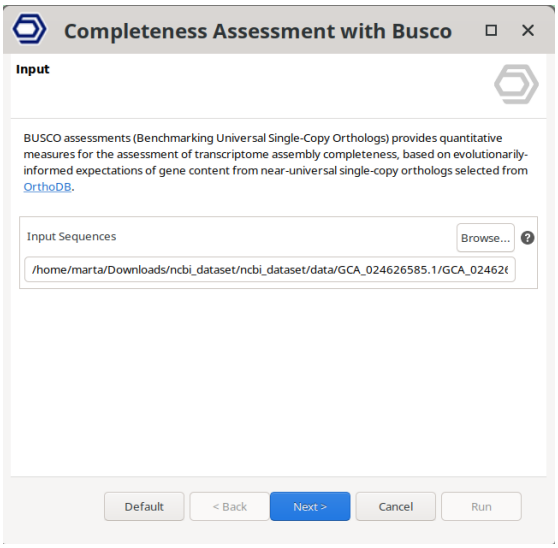


***Figure 1**. Input Wizard Page.*

CONFIGURATION

- **Lineage:**Choose the appropriate lineage-specific profile to classify matches, depending on the species to be assessed. Genes that make up the BUSCO sets for each major lineage are selected from orthologous groups with genes present as single-copy orthologs in at least 90% of species.
- **Mode:** Set the assessment mode according to the type of sequences to be analyzed.

- Genome: nucleotide sequences (e.g. transcriptome *de novo*assembly).
- Proteome: Protein amino acid sequences.
- **Blast e-Value:**The statistical significance threshold for reporting matches against a sequence database. If the statistical significance of alignment is greater than the e-Value threshold, this hit will not be reported. Lower e-Value thresholds are more stringent, leading to fewer results. Increasing the threshold shows less stringent matches. The default e-Value used by BUSCO is 1e-03.

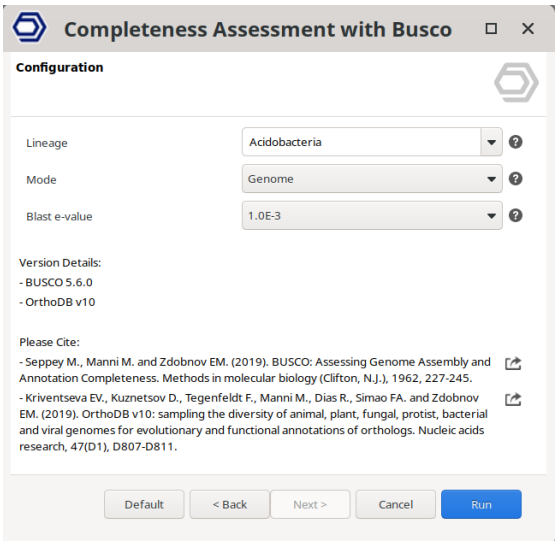


Figure 2.*Configuration Wizard Page*

Results

Once finished, a new tab is opened containing the results of the completeness assessment procedure (Figure 3). Each row corresponds to a BUSCO from the lineage database selected, and columns show the following information:

- BUSCO ID: Name of the BUSCO.
- Sequence ID: Name of the transcript/protein sequence matching the BUSCO.
- Score: Score of the alignment.
- Length: Length of the transcript/protein sequence matching the BUSCO.
- Tag: Result category.

The results are simplified into categories of Complete and single-copy, Complete and duplicated, Fragmented, or Missing BUSCOs:

- **Complete (single and duplicated):** The BUSCO matches have scored within the expected range of scores and within the expected range of length alignments to the BUSCO profile.
- **Fragmented:** The BUSCO matches have scored within the range of scores but not within the range of length alignments to the BUSCO profile. For transcriptomes or annotated gene sets, this indicates incomplete transcripts or gene models.
- **Missing:** There were either no significant matches at all, or the BUSCO matches scored below the range of scores for the BUSCO profile. For transcriptomes or annotated gene sets this indicates that these orthologous are indeed missing or the transcripts or gene models are so incomplete/fragmented that they could not even meet the criteria to be considered as fragmented.

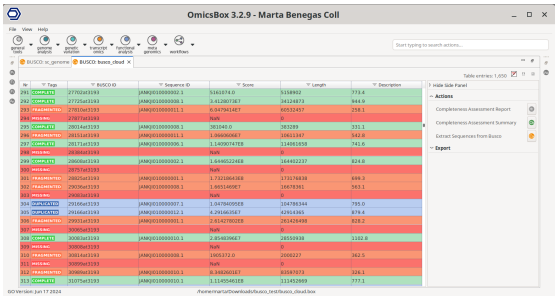


Figure 3.*BUSCO Project*

A result page will show a summary of the "Completeness Assessment" results (Figure 4). This page provides a quick evaluation of the results and provides ID lists containing BUSCO or transcript/protein identifiers assigned to the different categories. The result summary can be generated via **Side Panel → Actions → Completeness Assessment Report.**

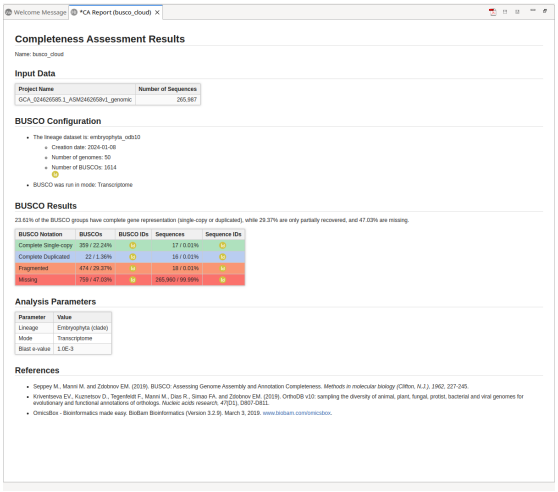


Figure 4.*Completeness Assessment Report*

Furthermore, the Completeness Assessment Summary chart (Figure 5) shows the percentage of lineage-specific BUSCOs assigned to each category. The pie chart can be generated via **Side Panel → Actions → Completeness Assessment Summary.**

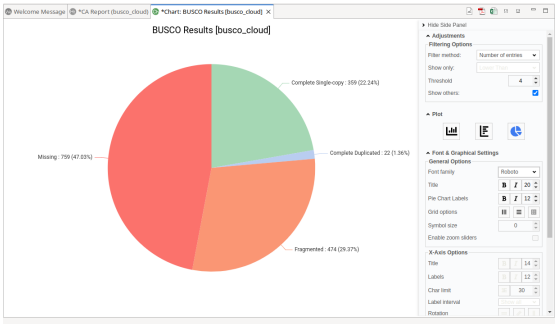


Figure 5.*Completeness Assessment Summary Chart*

Finally, the **Extract Original Sequences** utility (sidebar) allows extracting sequences from the original project based on its analysis status (Figure 6). For this, the original project containing the sequences that were assessed should be provided.

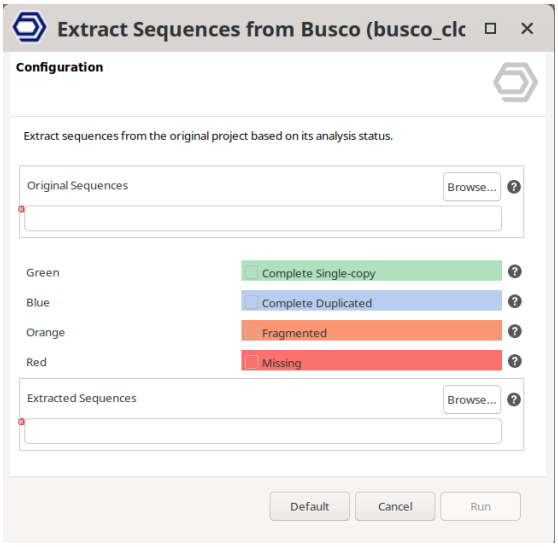


Figure 6.*Extract Original Sequences*

4.2.6 Genome Assembly Quality Assessment

The Genome Assembly Quality Assessment tool is designed to assess the quality of the *de novo* genome assemblies. It does so by comparing the assembly against the reference genome. Multiple assemblies can be specified, thus allowing easy comparison between them.

This tool can be useful in different scenarios. For example, it makes it easier to compare *de novo* assembled genomes obtained with different assembly algorithms. Usually, there's more than one option to perform the assembly, so it is difficult to decide at a first glimpse what will be the best for our dataset. Moreover, it may be interesting to try the same algorithm with different configurations. In all these cases, the Genome Assembly Quality Assessment tool allows the comparison of assemblies obtained with different strategies in order to try to decide the best configuration. Furthermore, once decided the best assembly strategy for our data, it may be transferrable to assemble data of similar characteristics (sequencing platform and related species, for example).

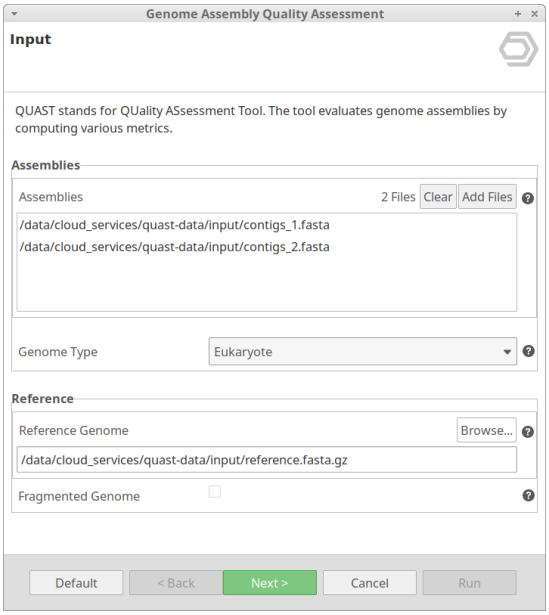
This tool is based on QUAST. Please cite QUAST as: Gurevich A, Saveliev V, Vyahhi N, Tesler G. QUAST: quality assessment tool for genome assemblies. Bioinformatics. 2013 Apr 15;29(8):1072-5. doi: 10.1093/bioinformatics/btt086. Epub 2013 Feb 19. PMID: 23422339; PMCID: PMC3624806.

RUN GENOME ASSEMBLY QUALITY ASSESSMENT

The tool is available in genome analysis > Genome Assembly Quality Assessment.

Input

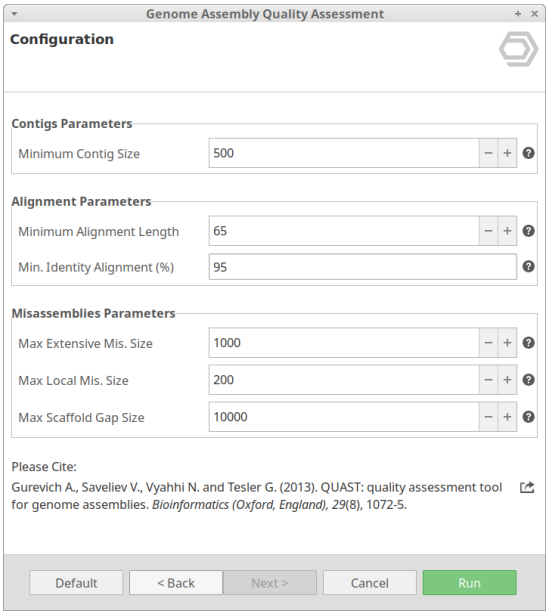
- **Assemblies**. One or more Genome Assembly(ies) in FASTA format. They can be the output of OmicsBox's DNA-Seq de Novo Assembly tool.
- **Genome Type**. Select how is the genome of the species under study:
- Prokaryote implies a circular genome, so QUAST takes this feature into account and correctly processes its linear representation.
- Eukaryote indicates QUAST that the genome is not circular.
- Eukaryote + Large. Besides eukaryote, the genome is large (typically > 100 Mbp). Thus, QUAST uses optimal parameters for the evaluation of large genomes. It modifies the default values of the Minimum Contig Size, Minimum Alignment Length, and Max Extensive Misassemblies Size. They can be overridden manually in the corresponding parameters. In addition, this mode tries to identify misassemblies caused by transposable elements and exclude them from the number of misassemblies. See Mikheenko et al., 2018 for more details.
- **Reference Genome.** Reference genome to compare the assemblies with. The assemblies are aligned to the reference using Minimap2 in order to obtain different quality metrics.
- **Fragmented Genome**. The reference genome is fragmented (e.g. a scaffold reference). QUAST will try to detect misassemblies caused by fragmentation and mark them fake.



Configuration

- **Minimum Contig Size** (in bp). Contigs shorter than this minimum size won't be taken into account to compute QUAST metrics (unless specified).
- **Minimum Alignment Length** (in bp). Assembly(ies) are aligned against the reference genome. Alignments shorter than this value will be filtered and won't be taken into account to compute QUAST metrics. Note that alignments shorter than 65 bp will be filtered regardless of this threshold.
- **Min. Identity Alignment (%)**. Minimum percentage of identity to consider as proper alignment. Alignments with an identity (%) lower than this value will be filtered and won't be taken into account to compute QUAST metrics. Note that alignments with an identity (%) lower than 80% will be filtered regardless of this threshold.
- **Extensive Misassemblies Size**. Gap or overlap size between the left and right flanking sequence of the aligned contig to be considered as a relocation. Greater gap or overlaps than this value are counted as Extensive Misassemblies, whereas lower gaps or overlaps are counted as Local Misassemblies.

- **Local Misassemblies Size**. Gap or overlap size between the left and right flanking sequence of the aligned contig to be considered as a Local Misassembly. Shorter inconsistencies are considered as (long) indels. Note that this value must be lower than the Extensive Misassemblies Size.
- **Max Scaffold Gap Size**. Maximum gap size between scaffolds to be detected as such. Longer inconsistencies are considered as relocations and thus, counted as extensive misassemblies. Note that this value must be greater than the extensive misassembly size.



RESULTS

QUAST Table

The main result is an OmicsBox table containing a set of summary statistics for each analyzed assembly (Figure 4). Those statistics are:

- **Total Length**. The total length in bp of the assembly.
- **Largest Contig**. The length in bp of the largest contig in the assembly.
- **N50.** It measures the contiguity of the assembly. The greatest the number, the more contiguous the assembly is. It is calculated by sorting the contigs from longest to shortest, then summing the length of the contigs until 50% of the total assembly size is reached. The N50 is the length of the contig in which this threshold is reached.
- **NG50**. It's similar to N50 but it is calculated taking into account the reference genome size instead of the total assembly size. Since the same reference size has been used to calculate the NG50 of all the assemblies, this metric is more comparable between assemblies.
- **L50**. It's another statistic that measures the contiguity of the assembly. The lower the number, the more contiguous the assembly is. It is calculated with the same procedure as the N50. However, the L50 is the number of contigs needed to reach 50% of the total assembly size. It is also the number of contigs with lengths equal to or greater than N50.
- **LG50**. Similar to L50 but it is calculated taking into account the reference genome size instead of the total assembly size. It is also more suitable to compare assemblies. It's only shown if QUAST has been able to calculate it.
- **Num. Misassemblies**. The total number of detected misassemblies. This includes relocations, translocations, and inversions. Please visit QUAST's manual for more information about how QUAST detects each type of misassemblies.

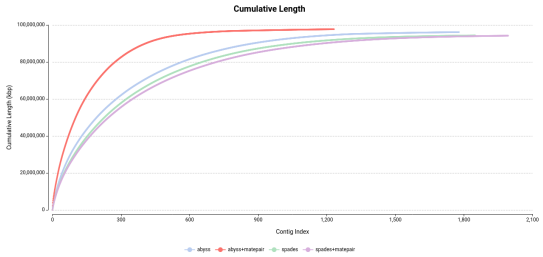
Summary Report

The Summary Report contains all the statistics calculated by QUAST (Figure 5). In addition, the parameters used during the analysis are specified as well at the end of the report. Please visit QUAST's manual for more details about each statistic.

Charts

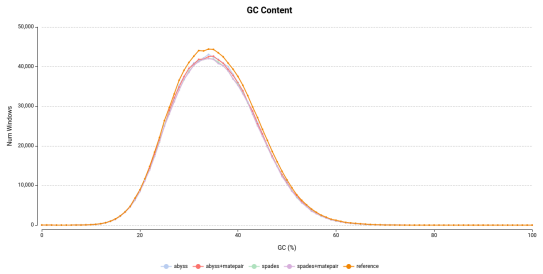
General Charts Cumulative Length

Shows the accumulation of contig sizes (Figure 3). The x-axis orders contigs from largest to smallest, while the y-axis represents the total size of the x-largest contigs in the assembly.



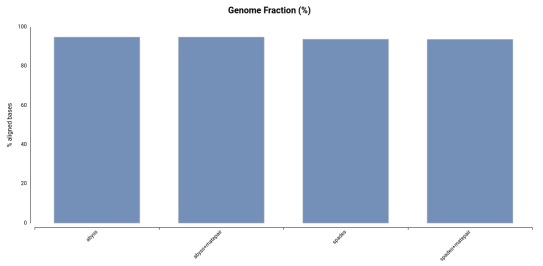
GC Content

Shows the distribution of GC content in the contigs (Figure 4). The x value is the GC percentage (0 to 100%). The y value is the number of non-overlapping 100 bp windows in which GC content equals x%. For a single genome, the distribution is typically Gaussian. However, for assemblies with contaminants, the GC distribution appears to be a superposition of Gaussian distributions, giving a plot with multiple peaks.



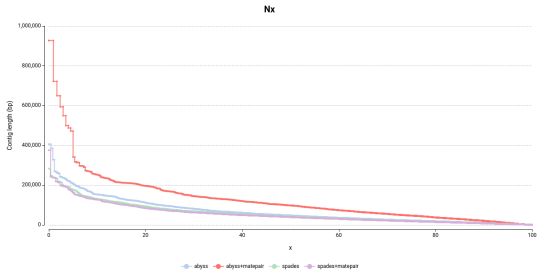
Genome Fraction

The genome fraction (%) is a measure of the number of bases in the reference genome that have been aligned to at least one contig in the assembly (Figure 5). This percentage is calculated by dividing the number of aligned bases by the total number of bases in the reference genome. It's important to note that contigs from repetitive regions may align to multiple locations in the genome, potentially leading to an overestimation of the genome fraction.



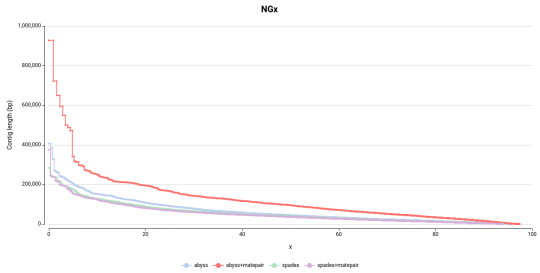
Nx Plot

Plots the values of N as a function of x, with x ranging from 0 to 100% (Figure 6). In order to calculate e.g. N50, contigs are first sorted from the longest to the shortest. Then, the contig's lengths are summed until 50% of the total assembly length is reached. N50 is the length of the contig in which that threshold is achieved.



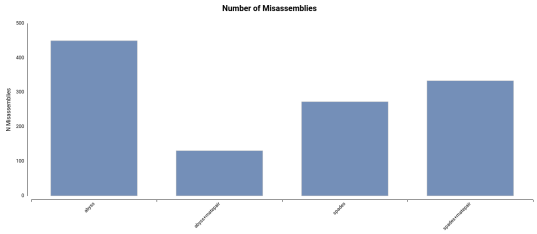
NGx Plot

Plots the values of NG as a function of x, with x ranging from 0 to 100% (Figure 7). NGx is calculated in the same way as the Nx, but taking as a reference the genome size instead of the assembly size. This makes NGx more comparable between assemblies. In addition, this statistic is more robust against changes in the assembly (e.g. filtering of the shortest contigs).



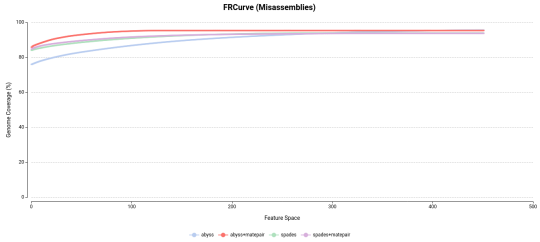
Misassemblies Charts Number of Misassemblies

Shows the total number of misassemblies found in each assembly (Figure 8).



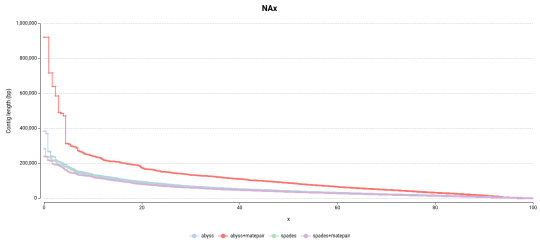
Feature-Response Curve

The x value (Feature space) is the total maximum number of misassemblies allowed in the contigs (Figure 9). The y value (Genome coverage %) is the total number of aligned bases in the contigs, divided by the reference length. The response (quality) of the assembler output is analyzed as a function of the maximum number of possible misassemblies allowed in the contigs.



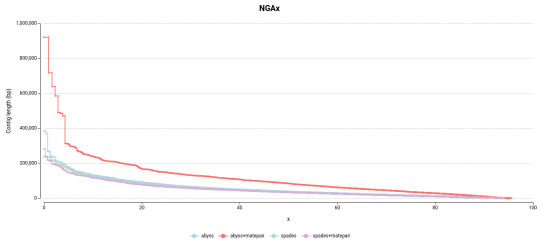
Alignment Charts NAx Plot

Plots the values of NA as a function of x, with x ranging from 0 to 100% (Figure 8). NAx is calculated in the same way as Nx, but taking into account only the part of the assembly that is aligned to the reference genome.



NGAx Plot

Plots the values of NGA as a function of x, with x ranging from 0 to 100% (Figure 9). NGAx is calculated in the same way as NGx, but taking into account only the part of the assembly that is aligned to the reference genome.



4.2.7 Repeat Masking

Introduction

Repetitive DNA sequences are abundant in a broad range of species. The term repeat is used to describe two different types of sequences: **low complexity** sequences, such as homopolymeric runs of nucleotides, and **transposable elements,**such as viruses, long interspersed nuclear elements (LINEs), and short interspersed nuclear elements (SINEs). Eukaryotic genomes can be very repeat-rich: for example, 47% of the human genome is thought to consists of repeats. Adequate repeat annotation should be a part of every genome annotation project.

Repeat identification and masking is usually a previous step to the gene prediction and annotation phase. The term 'masking' means transforming every nucleotide identified as a repeat to an 'N', 'X' or to a lower case a, t, g, or c (the latter is known as soft masking). The masking step signals to downstream sequence alignment and gene prediction tools that these regions are repeats. Identifying repeats is complicated by the fact that repeats are often poorly conserved; thus, accurate repeat detection usually requires a repeat library for the species of interest. Also, the borders of these repeats are usually ill-defined; repeats often insert within other repeats, and only fragments within fragments are present, which means that complete elements are found quite rarely.

Users must carefully post-process the outputs of this process since that failure to mask genome sequences can be catastrophic. Left unmasked repeats can seed millions of spurious BLAST alignments, producing false evidence for gene annotation. Worse still, many transposon open reading frames (ORFs) look like true host genes to gene predictors, causing portions of transposon ORFs to be added as additional exons to gene predictions, completely corrupting the final gene annotations. Good repeat masking is thus crucial for the accurate annotation of protein-coding genes.

This application is based on **RepeatMasker**. RepeatMasker is a program that screens DNA sequences and detects transposable elements, satellites, and low-complexity DNA sequences. The output of the program is a detailed annotation of the repeats that are present in the query sequence as well as a modified version of the query sequence in which all the annotated repeats have been masked. RepeatMasker uses a sequence search engine to perform its search for repeats. In OmicsBox, RMBLast and HMMER are supported. RepeatMasker also uses the Tandem Repeat Finder to detect tandem repeats.

RepeatMasker comes with the **Dfam Database.** The Dfam database is an open collection of DNA Transposable Element sequence alignments, Hidden Markov Models (HMMs), consensus sequences, and genome annotations. Dfam represents a collection of multiple sequence alignments, each containing a set of representative members of a specific transposable element family. These alignments (seed alignments) are used to generate HMMs and consensus sequences for each family. The Dfam website gives information about each family and provides genome annotations for a collection of core genomes. The current release (Dfam 3.0) contains 6,235 TE families spanning five organisms: human, mouse, zebrafish, fruit fly, nematode, and a growing number of additional species.

To supplement these databases, OmicsBox allows providing custom libraries, as well as the RepeatMasker edition of **RepBase**. RepBase is a database of representative repetitive sequences from eukaryotic species. Users can download the RepeatMasker library file from the Genetic Information Research Institute (GIRI) website after requesting an account opening.

Please cite RepeatMasker as:

Smit, AFA, Hubley, R & Green, P. RepeatMasker Open-4.0. 2013-2015 <http://www.repeatmasker.org >.

Run Repeat Masking

This functionality can be found under **Genome Analysis → Repeat Masking.**The wizard allows to provide input files and adjust analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

INPUT

- **Input Sequences:**Select the file that contains the DNA sequences to be masked. Input sequences must be in FASTA format.

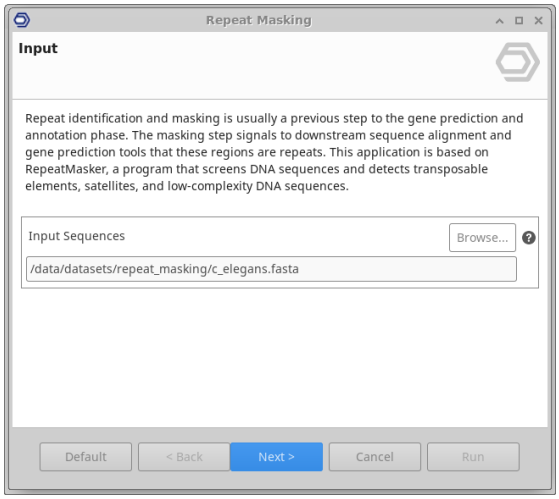


Figure 1:*Input Page*

SEARCH CONFIGURATION

- **Search Configuration:**Select the search engine to perform the search for repeats.
- **RMBlast:**This is a RepeatMasker compatible version of the NCBI Blast tool suite.
- **HMMER**: It uses the *nhmmer* program to search sequences against the Dfam database.
- **Repeat Database:**RepeatMasker works with these databases:
- **Dfam:** It is a database of transposable elements included in the application, so it is not necessary to provide any additional file.
- **Custom:**Allows providing a custom library of sequences to be masked in the query. The library file needs to contain repetitive elements in FASTA format. The recommended format for IDs in a custom library is ">repeatname#class/subclass".
- **RepBase:**We highly recommend obtaining the RepeatMasker edition of RepBase. Searches are optimized to use this database and can produce higher-quality annotations. To obtain the RepBase RepeatMasker edition go to the Genetic Information Institute website. This option expects an EMBL file as a database file.

This functionality is compatible with the RepBase RepeatMasker edition 20181026 and 20170127. Make sure you are providing the proper database.

- **Database File:** If it is necessary, select the file containing the database to perform the search.
- **Custom:**The library file needs to contain sequences in FASTA format. The recommended format for IDs in a custom library is ">repeatname#class/subclass".
- **Repbase:**EMBL file downloaded from the Genetic Information Institute website.
- **Species:**Specify the species or clade of the input sequence. The species name must be a valid NCBI Taxonomy Database species name and be contained in the RepeatMasker repeat database. Take into account that if HMMER is selected as the search engine, the Dfam database only contains information about human, mouse, zebrafish, fruit fly, and nematode.

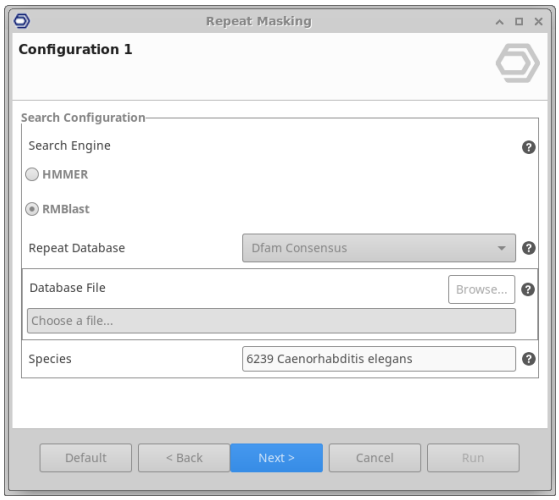


Figure 2:*Configuration Page 1*

RMBLAST OPTIONS

- **Speed/Sensitivity:**Select the sensitivity of the search. The more sensitive the longer the processing time:
- **Rush:**About 10% less sensitive and 4-10 times faster than the default option (quick searches are fine under most circumstances).
- **Quick:**5-10% less sensitive, 2-5 times faster than the default.
- **Slow:**0-5% more sensitive, 2-3 times slower than the default.
- **Apply Divergence Cutoff:**This option masks only those repeats that are less divergent from the consensus than a specific percentage.
- **Divergence Cutoff:** Set the divergence cutoff.

OUTPUT OPTIONS

- **Masking Options:**Select how sequences will be masked. Repetitive elements can be replaced by N, by X, or by lower case. Note that some downstream applications require a specific type of masking.
- **Only Alu elements:**Only masks Alus and 7SLRNA, SVA, and LTR5. This option only works for primate DNA.
- **Type of repeat:**Select the type of repeats that the algorithm will detect and mask: Interspersed repeats, simple repeats, and low complexity DNA, or both.

- **Not mask RNA genes:**RepeatMasker by default screens for matches to small pol III transcribed RNAs (mostly tRNAs and snRNAs) due to their close similarity to SINEs and the abundance of some of their pseudogenes. Check this option if you are interested in leaving the small RNA genes sequences unmasked.

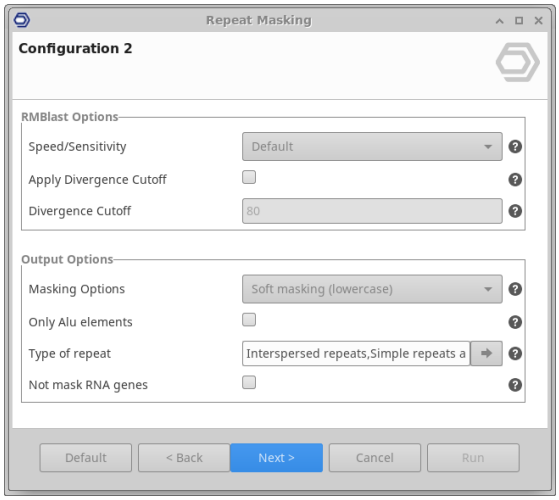


Figure 3:*Configuration Page 2*

OUTPUT

- **Output FASTA:**Select a file where the masked sequences will be placed.

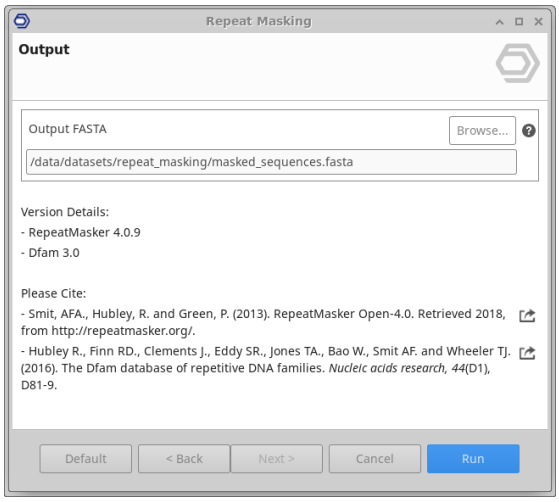


Figure 4:*Output Page*

Results

The Repeat Masking process returns the masked sequences in FASTA format and the location of the detected repeats in GFF format (Figure 5 and Figure 6). The repeat sequences found during the procedure are replaced by X, N, or lowercase (according to the selected mask option), so the output FASTA will contain the same

sequences as the input FASTA but with the nucleotides corresponding to a repetitive element masked. The coordinates and strand, as well as the class and subclass of each repetitive element, are annotated in the output GFF project.

```
>LK927563.1 Caenorhabditis elegans genome assembly
Ggaaagttctggaatgttccagaactttctagaaaaatcgagaaaattct
ggaatgttccagaactttctagaacaattgggaaagttctggaatgttcc
agaactttctagaaaaatcgagaaaattctggaatgttccagaactttct
agaaaaaattggaaaattctggagaagcagaaaaattgagcttagagctt
tagaagaggtagttatttgggagttgatgggggatcaagtcaaggtactg
tagtggtactatagggg tactgtaggtatacggtagggttactgtagttt
tgggaaaaattggcattttgtcctttgaagagatattgggttaggagttg
gtggaggataatgtcaaggtactgtagtggtattgtaaggttactgtctt
ggccaaaaagtaacagaaagttttcatactgtctgtgaatttttgaaaca
tgcatgtcgtagaaaaatacatacatgtaacagtgccagtaaaccgcgtt
ttaagtgtttcaagttttttttcatgatgagaaatttttgagtgattgaa
ggtatgtgtgtcaaatactttttaatggtgccagtcgttgcccgcgccgta
ggcgcagtcagcggctggTATATATATAATTTTTTTCAGGGAGACTTCCA
```

Figure 5: *Soft-masked Sequences (FASTA)*

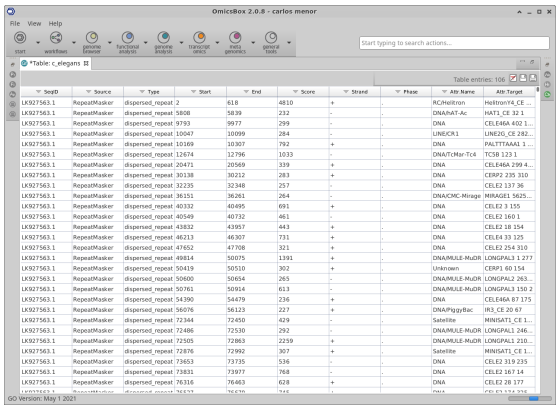
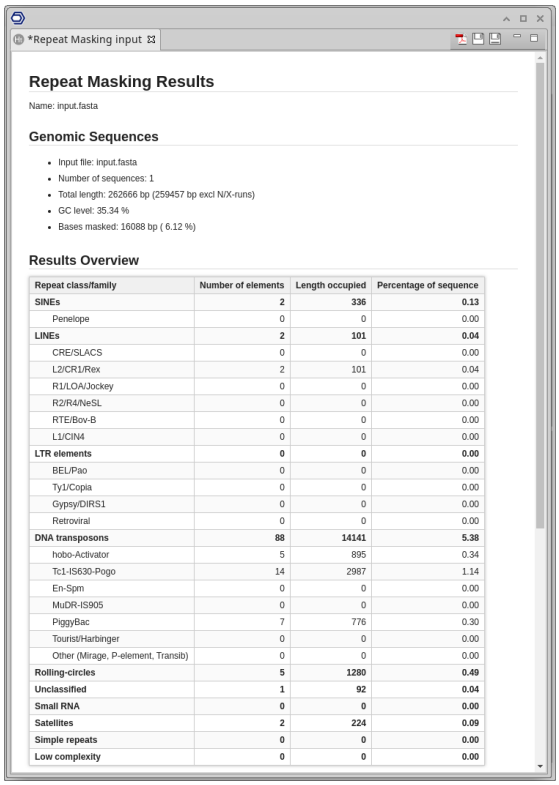


Figure 6:*Output GFF*

In addition to the resulting FASTA and GFF files, a report and a chart are generated. The report shows a summary of the Repeat Masking results (Figure 7). This page contains information about the input sequencing data and a results overview. The Results Overview table shows the number of elements, the length occupied, and the percentage of sequence that each repeat class and subclass covers.



*Repeat Masking input

Repeat Masking Results

Name: input.fasta

Genomic Sequences

- Input file: input.fasta
- Number of sequences: 1
- Total length: 262666 bp (259457 bp excl N/X-runs)
- GC level: 35.34 %
- Bases masked: 16088 bp (6.12 %)

Results Overview

Repeat class/family	Number of elements	Length occupied	Percentage of sequence
SINEs	**2**	**336**	**0.13**
Penelope	0	0	0.00
LINEs	**2**	**101**	**0.04**
CRE/SLACS	0	0	0.00
L2/CR1/Rex	2	101	0.04
R1/LOA/Jockey	0	0	0.00
R2/R4/NeSL	0	0	0.00
RTE/Bov-B	0	0	0.00
L1/CIN4	0	0	0.00
LTR elements	**0**	**0**	**0.00**
BEL/Pao	0	0	0.00
Ty1/Copia	0	0	0.00
Gypsy/DIRS1	0	0	0.00
Retroviral	0	0	0.00
DNA transposons	**88**	**14141**	**5.38**
hobo-Activator	5	895	0.34
Tc1-IS630-Pogo	14	2987	1.14
En-Spm	0	0	0.00
MuDR-IS905	0	0	0.00
PiggyBac	7	776	0.30
Tourist/Harbinger	0	0	0.00
Other (Mirage, P-element, Transib)	0	0	0.00
Rolling-circles	**5**	**1280**	**0.49**
Unclassified	**1**	**92**	**0.04**
Small RNA	**0**	**0**	**0.00**
Satellites	**2**	**224**	**0.09**
Simple repeats	**0**	**0**	**0.00**
Low complexity	**0**	**0**	**0.00**

Figure 7:*Summary Report*

The Repeat Distribution chart (Figure 8) shows the percentage of sequence covered by each repeat class.

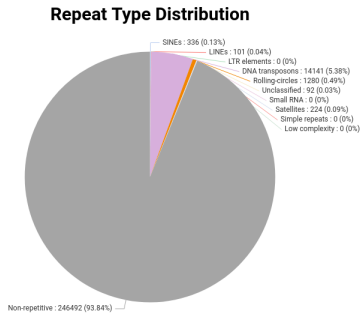


Figure 8:*Repeat Distribution Pie Chart*

4.2.8 Gene Finding

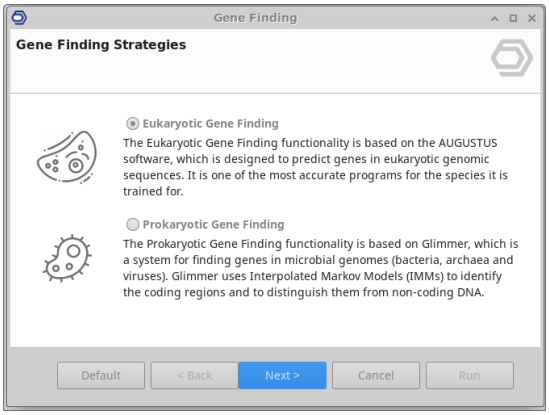
Gene Finding

With advancements in high throughput sequencing technologies, complete genomic sequences of novel species are becoming more and more abundant. Given a new genome, one of the most important tasks is determining the structure of its protein-coding genes. This procedure is known as gene prediction or gene finding, which is essential for genome characterization and allows downstream bioinformatics applications, such as functional annotation.

This functionality can be found under **Genome Analysis → Gene Finding**.

Two gene finding strategies are available:

- **Eukaryotic Gene Finding:** The Eukaryotic Gene Finding functionality is based on the AUGUSTUS software, which is designed to predict genes in eukaryotic genomic sequences. It is one of the most accurate programs for the species it is trained for.
- **Prokaryotic Gene Finding:** The Prokaryotic Gene Finding functionality is based on Glimmer, which is a system for finding genes in microbial genomes (bacteria, archaea, and viruses). Glimmer uses Interpolated Markov Models (IMMs) to identify the coding regions and to distinguish them from noncoding DNA.



Eukaryotic Gene Finding by AUGUSTUS

INTRODUCTION

The Eukaryotic Gene Finding functionality is intended to predict gene structures in genomic sequences, such as genomes, chromosomes, or scaffolds. It is based on the AUGUSTUS software which is designed to predict genes in genomic sequences, especially for those from eukaryotic organisms, and it is one of the most accurate programs for the species for which it is trained.

AUGUSTUS can be used as an *ab initio* program, which means it bases its prediction purely on the sequence. Includes pre-trained models for over 100 species. AUGUSTUS may also incorporate hints on the gene structure coming from extrinsic sources such as RNA-Seq, proteins, EST/cDNA, and IsoSeq data. Hints are extrinsic evidence about the location and structure of genes. Each hint is local information, associated with a particular genome region. When predicting genes, AUGUSTUS can incorporate these hints, which will change the likelihood of gene structure candidates. It will tend to predict gene structures that are in agreement with the hints.

Species

Archaea

- *Sulfolobus solfataricus*

Bacteria

- *Staphylococcus aureus*
- *Streptococcus pneumoniae*
- *Thermoanaerobacter tengcongensis*
- *Burkholderia pseudomallei*
- *Escherichia coli K-12*

Alveolata & Protozoan

- *Vitrella brassicaformis*
- *Plasmodium falciparum*
- *Toxoplasma gondii*
- *Tetrahymena thermophila*
- *Leishmania tarentolae*

Diatom

- *Fragilariopsis cylindrus*
- *Phaeodactylum tricornutum*
- *Pseudo-nitzschia multistriata*
- *Thalassiosira pseudonana*

Alga

- *Ectocarpus siliculosus*
- *Galdieria sulphuraria*

Fungi

- *Sphaceloma murrayae*
- *Aspergillus fumigatus*
- *Aspergillus nidulans*
- *Aspergillus oryzae*
- *Aspergillus terreus*
- *Coccidioides immitis*
- *Histoplasma capsulatum*
- *Botrytis cinerea*
- *Pneumocystis jirovecii*
- *Candida albicans*
- *Candida guilliermondii*
- *Candida tropicalis*
- *Eremothecium gossypii*
- *Kluyveromyces lactis*
- *Lodderomyces elongisporus*
- *Pichia stipitis (Scheffersomyces stipitis)*
- *Saccharomyces cerevisiae (RM11-1a_1)*
- *Saccharomyces cerevisiae (S288C)*
- *Yarrowia lipolytica*
- *Schizosaccharomyces pombe*
- *Chaetomium globosum*
- *Fusarium graminearum*
- *Magnaporthe grisea*
- *Neurospora crassa*
- *Sordaria macrospora*
- *Verticillium albo-atrum*
- *Verticillium longisporum*
- *Coprinopsis cinerea*
- *Laccaria bicolor*
- *Phanerochaete chrysosporium*
- *Cryptococcus gattii*
- *Cryptococcus neoformans*
- *Ustilago maydis*
- *Gonapodya prolifera*
- *Encephalitozoon cuniculi*
- *Rhizopus oryzae*
- *Conidiobolus coronatus*

Nematoda & Nemertea (Roundworms & Ribbon worms)

- *Ancylostoma ceylanicum*
- *Brugia malayi*
- *Caenorhabditis elegans*
- *Trichinella spiralis*
- *Notospermus geniculatus*

Platyhelminthes (**Flatworms)**

- *Schistosoma mansoni*

Arthropoda (Insecta & Arachnida)

- *Parasteatoda sp.*
- *Acyrthosiphon pisum*
- *Aedes aegypti*
- *Apis dorsata*
- *Apis mellifera*
- *Bombus impatiens*
- *Bombus terrestris*
- *Camponotus floridanus*
- *Culex pipiens*
- *Drosophila melanogaster*
- *Heliconius melpomene*
- *Nasonia vitripennis*
- *Rhodnius prolixus*
- *Tribolium castaneum*

Chordata (Fish, Bird & Mammal)

- *Danio rerio*
- *Xiphophorus maculatus*
- *Ciona intestinalis*
- *Callorhinchus milii*
- *Rhincodon typus*
- *Scyliorhinus torazame*
- *Lethenteron camtschaticum*
- *Petromyzon marinus*
- *Gallus gallus*
- *Homo sapiens*

Cnidaria & Ctenophora (Jellyfish & Anemone)

- *Nematostella vectensis*
- *Aurelia aurita*
- *Cassiopea xamachana*
- *Chrysaora chesapeakeij*
- *Nemopilema nomurai*
- *Rhopilema esculentum*
- *Mnemiopsis leidyi*

Echinodermata (Starfish & Sea Urchin)

- *Pisaster ochraceus*
- *Strongylocentrotus purpuratus*

Hemichordata & Mollusca (Acorn worm & Mollusk)

- *Ptychodera flava*
- *Argopecten irradians*

Placozoa (Marine free-living organism)

- *Trichoplax adhaerens*

Porifera (Sponge)

- *Amphimedon queenslandica*

Viridiplantae (Plant)

- *Chlamydomonas eustigma*
- *Chlamydomonas reinhardtii*
- *Dunaliella salina*
- *Monoraphidium neglectum*
- *Raphidocelis subcapitata*
- *Volvox sp.*
- *Chloropicon primus*
- *Bathycoccus prasinos*
- *Micromonas commoda*
- *Micromonas pusilla*
- *Ostreococcus sp. 'lucimarinus'*
- *Ostreococcus tauri*
- *Chlorella sp.*
- *Arabidopsis thaliana*
- *Nicotiana attenuata*
- *Oryza sativa*
- *Solanum lycopersicum*
- *Theobroma cacao*
- *Triticum sp.*
- *Zea mays*

RNA-Seq Hints

RNA-Seq alignments provide two types of features that are helpful for gene prediction:

- Spliced alignments of reads give information about introns.
- Coverage (e.e, how many reads are aligned to a particular position in the genome) gives information about exons.

The integration of coverage (exon part) information is not trivial. The problem is that coverage may not only be high in CDS regions, but also in UTRs and in partially retained introns. If the selected species do not have UTR parameters (see UTR Prediction parameter below), RNA-Seq hints are not recommended.

RNA-Seq data is required as sequencing reads in FASTA/FASTQ format. Reads are aligned to the genome using the STAR aligner software. If RNA-Seq hints are provided, please cite STAR as:

Dobin A, Davis CA, Schlesinger F, et al (2012). "STAR: ultrafast universal RNA-seq aligner." Bioinformatics, 29(1):15-21.

Protein Hints

Protein alignments can aid the prediction of CDSs (including the correct reading frame, start and stop codon positions) and the prediction of introns.

Protein data is required in FASTA format. Proteins are aligned to the genome using the GenomeThreader software. If protein hints are provided, please cite GenomeThreader as:

Gremme G, Brendel V, Sparks M E, and Kurtz S (2005). "Engineering a software tool for gene structure prediction in higher organisms". Information and Software Technology, 47(15):965-978.

EST & cDNA Hints

ESTs (Expressed Sequence Tags) and cDNAs are suitable for generating intron, exon part, and exon hints.

EST and cDNA sequences are required in FASTA format. ESTs and cDNAs are aligned to the genome using BLAT and pslCDnaFilter. If EST/cDNA hints are provided, please cite BLAT as:

Kent WJ (2002). "BLAT--the BLAST-like alignment tool". Genome Res., 656-64.

IsoSeq Hints

Single-molecule Pacific Bioscience (PacBio) RNA-seq reads can improve the identification of new isoforms. Circular Consensus Sequences (CCS) from IsoSeq often constitute near-full-length transcripts.

IsoSeq sequences are required in FASTA format. IsoSeq sequences are aligned to the genome using GMAP. If IsoSeq hints are provided, please cite GMAP as:

Wu TD, Watanabe CK (2005). "GMAP: a genomic mapping and alignment program for mRNA and EST sequences". Bioinformatics, 1;21(9):1859-75.

Please cite AUGUSTUS as:

Hoff KJ. and Stanke M. (2019). Predicting Genes in Single Genomes with AUGUSTUS. Current protocols in bioinformatics, 65(1), e57.

RUN EUKARYOTIC GENE FINDING

This functionality can be found under Genome Analysis → Gene Finding → Eukaryotic Gene Finding. The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, and Figure 3).

Input

- **Input Sequences:** Select the file containing the input DNA sequences. This application expects genomic sequences (e.g. genome, chromosomes, scaffolds…). Sequences must be in FASTA or multi-FASTA format. Everly letter other than A, C, G, and T is interpreted as an unknown base.

If repeat masked sequences are provided, masked regions must be indicated in lowercase (soft masking).

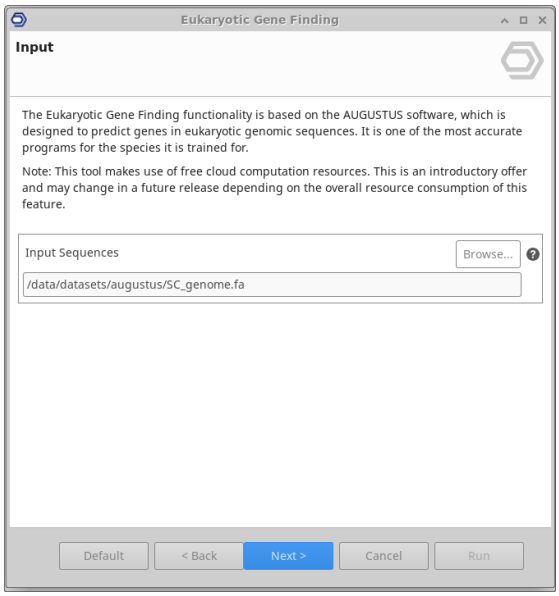


Figure 1: *Input Page*

Configuration: General

- **Closest Species:** AUGUSTUS has been trained for predicting genes in the following species. The closest species to the query should be selected. Each option shows the scientific names of the species, the kingdom, the phylum, and the class to which it belongs (if this information is available). Provide any of these taxonomies (e.g. class) to filter and find all the species related to the search term (e.g. if "Fungi" is provided, all species of the Fungi kingdom are displayed).
- **Strand:** Report predicted genes on both strands, just the forward or just the reverse strand.
- **Ignore Strand Conflicts:** Predict genes independently on each strand and allow overlapping genes on opposite strands.

This option is not available for prokaryotic species (archaea and bacteria).

- **Allowed Gene Structure:** Restrict the sear to one of these gene models:
- Partial: Allow prediction of incomplete genes at the sequence boundaries. This option is recommended.
- Intronless. Only predict single-exon genes like in prokaryotes and some eukaryotes.
- Complete: Only predict complete genes.
- At Least One: Predict at least one complete gene.
- Exactly One: Predict exactly one complete gene.
- **Output Genomic Features:** Specify which features should be reported: introns, start codons, and stop codons.
- **UTR Prediction:** Predict the untranslated regions in addition to the coding sequence. UTR prediction is only supported in combination with the Partial and Complete gene structures. UTR prediction is not possible in combination with the Ignore Strand Conflicts option. This option currently works only for a subset of species.

Species allowed for UTR prediction

- *Acyrthosiphon pisum*
- *Amphimedon queenslandica*
- *Apis melifera*
- *Bombus terrestris*
- *Caenorhabditis elegans*
- *Drosophila melanogaster*
- *Homo sapiens*
- *Trichinella spiralis*
- *Toxoplasma gondii*
- *Arabidopsis thaliana*
- *Chlamydomonas reinhartii*
- *Galdieria sulphuraria*
- *Solanum lycopersicum.*

If RNA-Seq hints are provided, this option is activated automatically (if possible), regardless of the user's choice.

- **No In-frame Stop Codons:** Do not report transcripts with in-frame stop codons. Otherwise, intron-spanning stop codons could occur.
- **Stop Codons Excluded From CDS:** By default, stop codons are included in CDSs, which is required by the GFF3 standard. Check this option to exclude stop codons from CDS.
- **Repeat Masked Sequences:** If repeat masked genome sequences are provided, mark this option. Note that AUGUSTUS expects the soft-masked version of the genome (repeat fragments are represented in lowercase characters).

Repeats can severely disturb gene prediction. It is strongly recommended to mask genome sequences for gene prediction. This task can be done within OmicsBox: Repeat Masking.

- **Sample:** AUGUSTUS reports the posterior probabilities of exons, introns, transcripts, and genes. The posterior probabilities are estimated using a sampling algorithm. This parameter adjusts the number of sampling iterations. The higher value is the more accurate is the estimation. The default is 100. If you do not need the posterior probabilities, set this parameter to 0.
- **Alternatives From Sampling:** Report alternative transcripts generated through probabilistic sampling. If this option is checked, the following parameters can be adjusted.

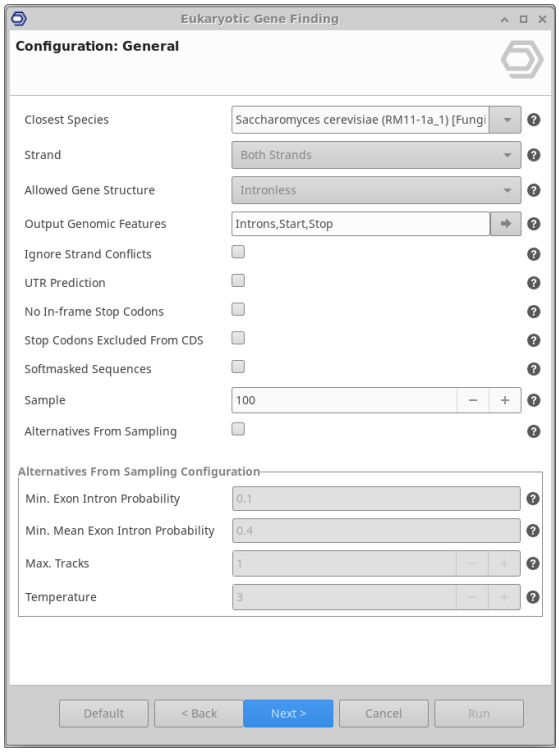


Figure 2: *General Configuration Page*

Alternatives From Sampling Configuration

- **Min. Exon Intron Probability:** Threshold between 0 and 1 to filter out transcripts with low exon and intron probabilities.
- **Min. Mean Exon Intron Probability:** Threshold between 0 and 1 to filter out transcripts with low mean exon and intron probabilities.
- **Max. Tracks:** Upper limit for the number of transcripts that span any given genome position.
- **Temperature:** If the aim is to produce a diverse, sensitive (including) set of gene structures, this parameter can be increased. The larger temperature the more alternatives are sampled. 3 is a good compromise between getting a high sensitivity but not getting too many exons sampled in total.

Configuration: Gene Finding Mode

- **Gene Finding Mode:** Choose the Gene Finding Mode.
- ***Ab initio* Prediction:** The *Ab initio* mode relies only on the pre-computed trained models. It predicts genes using probabilistic models based on Hidden Markov Models.
- **Prediction Using Extrinsic Evidence:** The Extrinsic Evidence mode uses experimental evidence to identify parts of gene structures, to uncover alternative splicing, o to overall improve annotation quality. If this option is selected, the Extrinsic Evidence Configuration section can be adjusted.
- **Extrinsic Evidence Data:** The Extrinsic Evidence Mode support extrinsic evidence hints from:
- RNA-Seq: Sequencing reads in FASTA or FASTQ format. If data is single-end, provide a single file as an RNA-Seq SE file. If data is paired-end, provide the upstream file as RNA-Seq SE/Upstream, and the downstream file as RNA-Seq Downstream.
- Protein: Protein sequences in FASTA format.
- EST/cDNA: EST or cDNA sequences in FASTA format.
- IsoSeq: Single-molecule Pacific Bioscience (PacBio) reads in FASTA or FASTQ format.

One file of each type is supported.

Extrinsic Evidence Configuration

- **Minimum Intron Length:** Define the minimum length of intron hints.
- **Maximum Intron Length:** Define the maximum length of intron hints.
- **Allow Hinted Splice Sites (AT/AC):** This option allows to predict the (rare) introns that start with AT and end with AC, in addition to the GT-AG and GC-AG introns that are allowed by default.
- **Alternatives From Evidence:** Report alternative transcripts when they are suggested by hints.

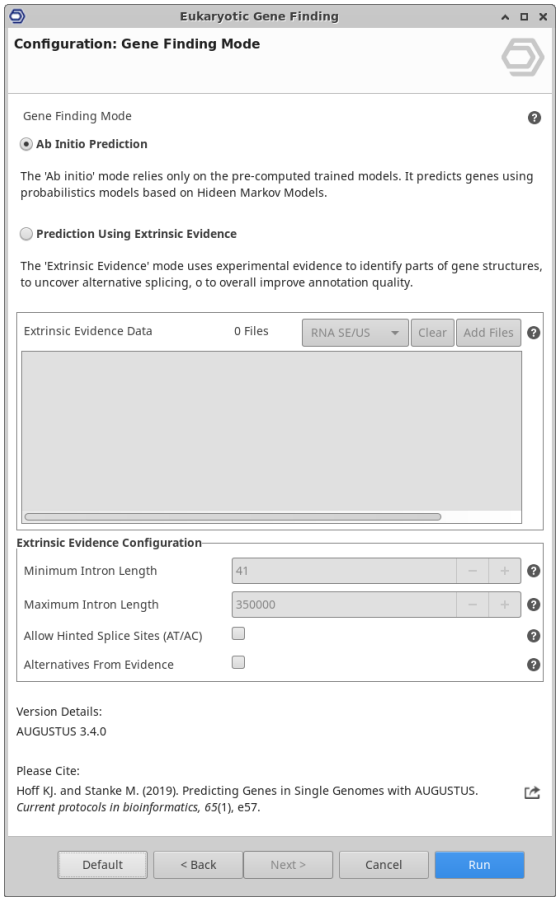


Figure 3: *Gene Finding Mode Page*

RESULTS

The Eukaryotic Gene Finding process returns the results in three projects (Figure 4):

- GFF Coordinates: This project contains the coordinates of the predicted genomic features in GFF format. It may contain genes, transcripts, introns, start codons, stop codons, and CDSs, depending on the "Output Genomic Features" selected when configuring the analysis (see the "Configuration: General" section).
- CDS Sequences: A sequence table that contains the nucleotide sequences for coding regions of the predicted genes.
- Protein Sequences: A sequence table that contains the protein sequences of the predicted genes.

In CDS and Protein projects, identifiers (SeqName) have the format "g1.t1". The "g1" indicates that the CDS / Protein comes from the "g1" gene. The "t1" indicates that the CDS / Protein comes from the "t1" transcript, which belongs to the "g1" gene. When the "Alternatives From Sampling" or "Alternatives From Evidence" options are provided, more than one transcript (isoform) per gene can be reported. Thus, the additional isoforms are called "g1.t2" and so on. The description column shows the genomic sequence to which each CDS or protein belongs.

The "coordinates" project follows the GFF format specification. It contains one line per predicted feature. The columns contain:

- SeqID: Name of the chromosome or scaffold.
- Source: Name of the program that generated this feature (AUGUSTUS).
- Type: Feature type name (e.g. gene, transcript, intron, CDS...).

Note that CDS entries in the GFF define exon regions. The sequences contained in the CDS project contain all CDS entries for the corresponding gene/transcript.

- Start: Start position of the feature.
- End: End position of the feature.
- Score: AUGUSTUS reports the posterior probabilities of exons, introns, transcripts, and genes. The reported probability of a gene is the probability that some coding sequence is in the reported range on the reported strand, regardless of the exact transcript. The posterior probabilities are estimated using a sampling algorithm.
- Strand: Defined as + (forward) or - (reverse).
- Phase: Indicates the base of the feature that is the first base of a codon (0, 1 or 2).
- Attributes: Provide additional information about the feature.
- Attr.ID: Feature identifier.
- Atrr.Parent: Identifier of the parent feature.
- Attr.HintSupport: Hint support percentage. It is the percentage of the feature that has been supported by the extrinsic evidence data provided.

The "Attr.HintSupport" column is only displayed when the Prediction Using Extrinsic Evidence mode is used.

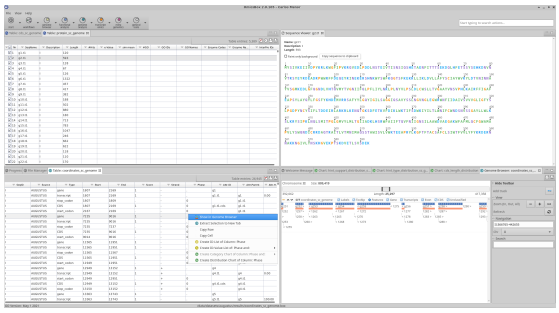
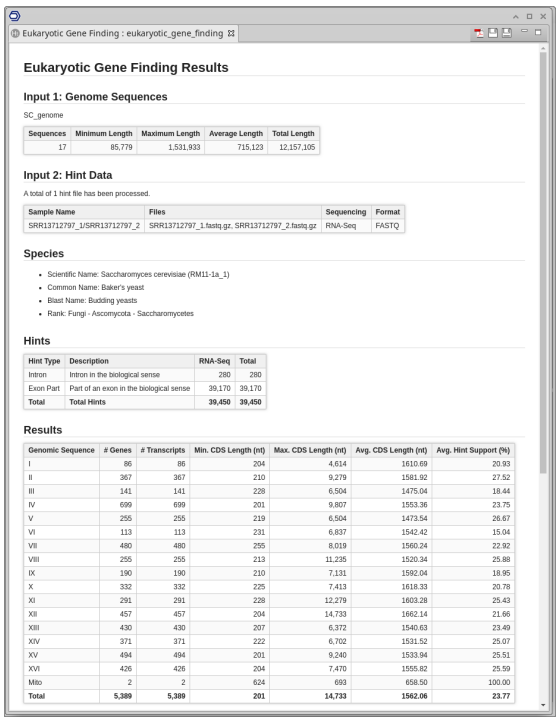


Figure 4: *Eukaryotic Gene Finding Results*

In addition to GFF and sequence projects, a result page will show a summary of the "Eukaryotic Gene Finding" results (Figure 5). This page provides information about the input data and the selected species, as well as a quick evaluation of the results obtained. If hint data was provided, an additional section is included, which summarizes the information obtained from the hint data.



Eukaryotic Gene Finding : eukaryotic_gene_finding

Eukaryotic Gene Finding Results

Input 1: Genome Sequences

SC_genome

Sequences	Minimum Length	Maximum Length	Average Length	Total Length
17	85,779	1,531,933	715,123	12,157,105

Input 2: Hint Data

A total of 1 hint file has been processed.

Sample Name	Files	Sequencing	Format
SRR13712797_1/SRR13712797_2	SRR13712797_1.fastq.gz, SRR13712797_2.fastq.gz	RNA-Seq	FASTQ

Species

- Scientific Name: Saccharomyces cerevisiae (RM11-1a_1)
- Common Name: Baker's yeast
- Blast Name: Budding yeasts
- Rank: Fungi - Ascomycota - Saccharomycetes

Hints

Hint Type	Description	RNA-Seq	Total
Intron	Intron in the biological sense	280	280
Exon Part	Part of an exon in the biological sense	39,170	39,170
Total	**Total Hints**	**39,450**	**39,450**

Results

Genomic Sequence	# Genes	# Transcripts	Min. CDS Length (nt)	Max. CDS Length (nt)	Avg. CDS Length (nt)	Avg. Hint Support (%)
I	86	86	204	4,614	1610.69	20.93
II	367	367	210	9,279	1581.92	27.52
III	141	141	228	6,504	1475.04	18.44
IV	699	699	201	9,807	1553.36	23.75
V	255	255	219	6,504	1473.54	26.67
VI	113	113	231	6,837	1542.42	15.04
VII	480	480	255	8,019	1560.24	22.92
VIII	255	255	213	11,235	1520.34	25.89
IX	190	190	210	7,131	1592.04	18.95
X	332	332	225	7,413	1618.33	20.78
XI	291	291	228	12,279	1603.28	25.43
XII	457	457	204	14,733	1662.14	21.66
XIII	430	430	207	6,372	1540.63	23.49
XIV	371	371	222	6,702	1531.52	25.07
XV	494	494	201	9,240	1533.94	25.51
XVI	426	426	204	7,470	1555.82	25.59
Mito	2	2	624	693	658.50	100.00
Total	**5,389**	**5,389**	**201**	**14,733**	**1562.06**	**23.77**

Figure 5: *Eukaryotic Gene Finding Report*

Furthermore, different charts are generated for a global visualization of the results.

Length Distribution Chart

This chart shows the distribution of lengths of the predicted CDS sequences (Figure 6). Note that this distribution is computed from the sequences contained in the CDS project.

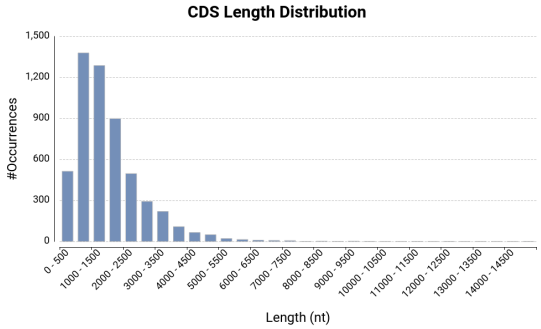


Figure 6: *Length Distribution Chart*

Hint Support Distribution Chart

This chart shows the distribution of hint support (%) of the predicted CDS sequences (Figure 7). It is only available for the Prediction Using Extrinsic Evidence mode.

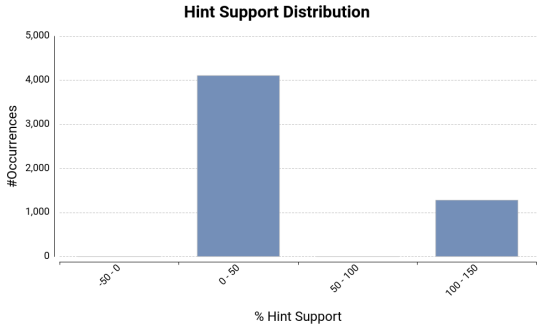


Figure 7: *Hint Support Distribution Chart*

Hint Type Distribution Chart

This chart shows the distribution of hint types that have been obtained from the extrinsic evidence data provided (Figure 8). A description of each hint type is included in the summary report. This chart is only available for the Prediction Using Extrinsic Evidence mode.

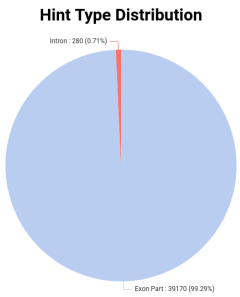


Figure 8: *Hint Type Distribution Chart*

Prokaryotic Gene Finding by Glimmer

INTRODUCTION

Glimmer (Gene Locator and Interpolated Markov ModelER) is a system for finding genes in microbial DNA, especially the genomes of bacteria, archaea, and viruses. Glimmer uses Interpolated Markov Models (IMMs) to identify the coding regions and to distinguish them from noncoding DNA. Glimmer was the primary microbial gene finder used at The Institute for Genomic Research (TIGR), where it was first developed, and since then has been used to annotate the genomes of hundreds of bacterial and archaeal species from TIGR and other labs.

The precision of Glimmer lies in its Interpolated Context Models (ICM), which are built for every query genome, by calculating and adapting the algorithm parameters to the GC content, the start and stop codons, etc.

First, this tool takes all provided FASTA files to build the most accurate model for the genome under study. Once the model is built, it performs the gene finding for each input sequence. In addition, the prokaryotic gene finding application allows saving the model created with all the sequences from the same organism, and use it to perform gene prediction on short sequences without loading the complete genome. This could be useful to run this procedure on small genomic fragments. Furthermore, if the complete genome of the target organism is not available, a model can be created from the genome of a close evolutionary species.

Please cite Glimmer as:

Delcher AL., Harmon D., Kasif S., White O. and Salzberg SL. (1999). Improved microbial gene identification with GLIMMER. Nucleic acids research, 27(23), 4636-41.

RUN PROKARYOTIC GENE FINDING

This functionality can be found under **Genome Analysis → Gene Finding → Prokaryotic Gene Finding.** The wizard allows to provide input files and adjust analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

Input

- **Input Sequences:** Provide the files containing the DNA input. It must be uncompressed and in FASTA or multi-FASTA format. In order to create a robust and accurate model, all the FASTA selected will be combined in one multi-FASTA, which will be used to create the Interpolated Context Model. Please, select the FASTA files or folder containing FASTA files for the query organism.

Note: Be sure to select only the FASTA files containing the sequences of the query organism.

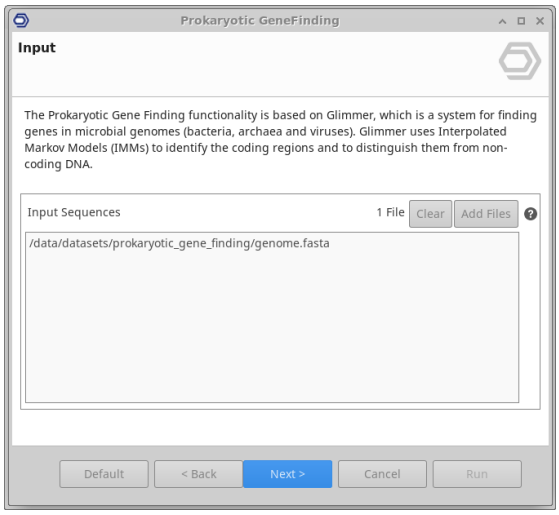


Figure 1:*Input Page*

Gene Settings

- **Genetic Code**: Choose the most appropriate genetic code for the query genome. Available genetic codes:
- The Standard Code (1st).
- The Mold, Protozoan, Coelenterate Mitochondrial, and the Mycoplasma/Spiroplasma Codes (2nd).
- The Bacterial, Archaeal, and Plant Plastid Codes (11th).
- **Minimum Gene Length**: ORFs shorter than this value (nucleotides) will not be considered as genes.
- **Maximum Gene Overlap**: Set the maximum overlap length (bp) for the predicted genes. Unlike eukaryotic genes, prokaryotic genes often have their genes overlapped.
- **Minimum Gene Score**: Each ORF found has an assigned score depending on its length, start, and stop codons. Here the limit of the score necessary to consider an ORF as a gene can be adjusted. Decreasing this parameter increases the number of genes found but also increases the errors in the prediction. Increasing this parameter decreases the number of genes found but also increases their reliability.

- **Genome Shape**: Select the shape of the genome under study. If "Linear" is selected, there will be no genes that "span" the junction between the start and the end of the sequence.

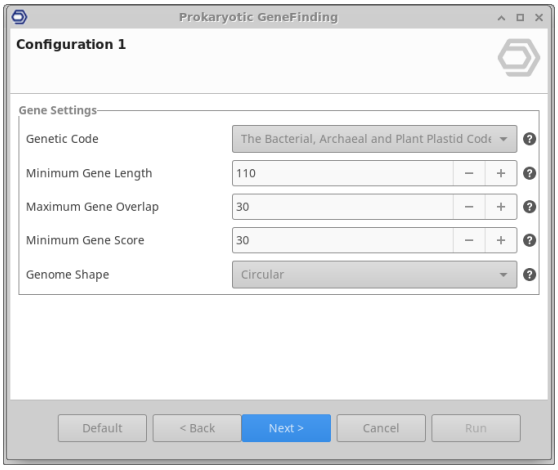


Figure 2: *Configuration Page 1*

ICM Settings

- **Choose ICM Option:** Choose between creating a new ICM model or using an existing one.

Note: The ICM model is species-specific: the more sequences used to build it, the more accurate the model will be.

- **Set Advanced ICM Parameters:** Allows modifying the Interpolated Context Model creation.
- **Allow in-frame Stops**: If checked, ORFs with in-frame stop codons are considered to build the ICM model. The stop codons are determined by the genetic code.
- **ICM Depth:** Set the maximum number of positions in the context window that will be used to determine the probability of the predicted position.
- **ICM Width**: Set the width of the ICM to the desired number including the predicted position. It refers to the width of the slicing window that builds the model.
- **ICM Period:** Set the number of different submodels for different positions in the text in a cyclic pattern.

For example, if the period is 3:

- The first submodel will determine positions 1, 4, 7, ...
- The second submodel will determine positions 2, 5, 8, ...
- The third submodel will determine positions 3, 6, 9, ...
- **Gene Entropy Cutoff**: The initial set of candidate ORFs can be filtered using entropy distance, which generally produces a more accurate training set, particularly for high-GC-content genomes.

Only genes with an entropy distance score smaller than the given value will be considered. This parameter is inspired by the fact that the coding sequences can be translated to an amino acid sequence (protein), whereas the non-coding sequences do not have this function. The class of amino acid sequences that are able to fold into a protein has a global organizational order in contrast to those pseudo-amino-acid sequences generated from non-coding (or completely random) DNA sequences.

Looking at the amino acid composition (or abundance) of a sequence, the entropy of the resulting protein can be determined, which allows to cluster two types of sequences (coding and non-coding).

- **Save ICM Model:** Allows to save the ICM resulting file to use in the next runs.
- **Precomputed ICM Model:** Select the file containing the Interpolated Context Model (ICM).

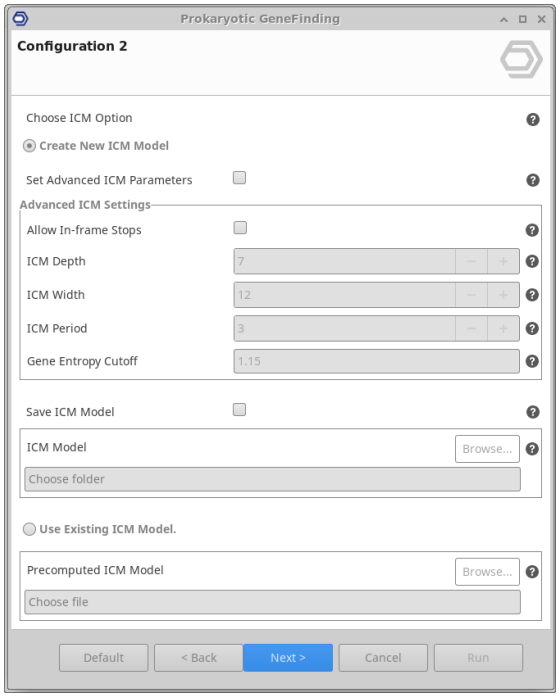


Figure 3: *Configuration Page 2*

Advanced Parameters

- **Run Model:** The single-mode executes Glimmer once. The iterated mode executes Glimmer twice, calculating automatically many parameters and using the results from the first run to generate a training set for the second one. This approach could increase the accuracy.
- **Define GC content**: Allows set the GC content (%). Otherwise, the GC% is calculated from the query genome.
- **GC Content:** Establish the GC content (%).
- **Set Start Codons**: This allows to set the start codons. Otherwise, the start codons are automatically set.
- **Start Codons:** Establish start codons (comma-separated list).
- **Start Codons Weight**: Specify the probability of the provided start codons (same number and order as in the `Start codons' parameter). If weights are not provided, the same weight will be used for all start codons.
- **Set Stop Codons:** This allows to set the stop codons. Otherwise, the stop codons are automatically set.

- **Stop Codons**: Establish stop codons (comma-separated list).

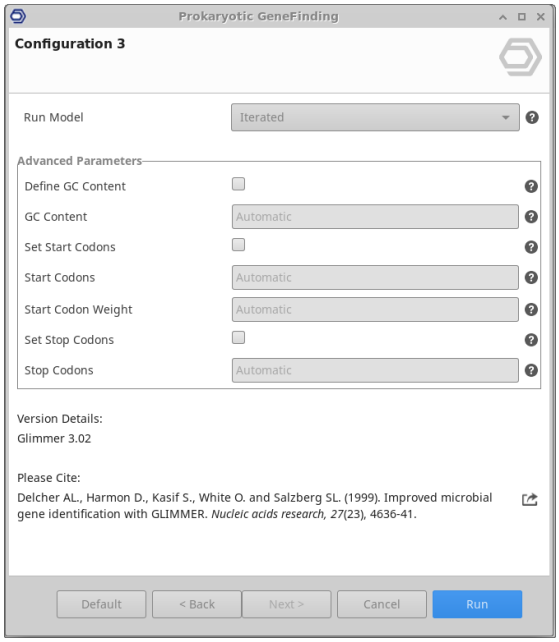


Figure 4: *Configuration Page 3*

RESULTS

Once the prokaryotic gene-finding tool has finished, two projects are automatically opened:

- **Sequence table:** OmicsBox sequence table containing the nucleotide sequence of the predicted genes. The sequence name corresponds to the FASTA ID line plus a gene identification.
- **GFF3 table:** Here you can see the results as a GFF file with:
- Sequence: The name of the source sequence that belongs to this feature.
- Source: The name of the program that has predicted this feature, in this case, 'Glimmer'.
- Type: The type of the feature (e.g. 'region', 'gene', and 'CDS').
- Start: The coordinate of the start codon.
- End: The coordinate of the stop codon.
- Score: The score assigned to the feature, except the exons.
- Strand: The strand of the feature, where a `+' means that the feature is forward-oriented and` -' backward.
- Phase: The correct frame to translate this feature, the values can be `0',` 1' or `2'. A gene` set' of features can have variant phase values, due to a frameshift in an intron.
- Attributes: Here we can see all the attributes assigned to each feature. The attributes are `ID' that assigns an id to each feature,` parent' present on the CDS and exon features, and provides information about the feature to which it belongs (refereeing to the sequence by its ID).

The resulting GFF3 can be inspected using the Genome Browser. To display a GFF entry right-click on it and select the **Show in the Genome Browser**option (Figure 5). For more information about this feature visit the Genome Browser documentation section.

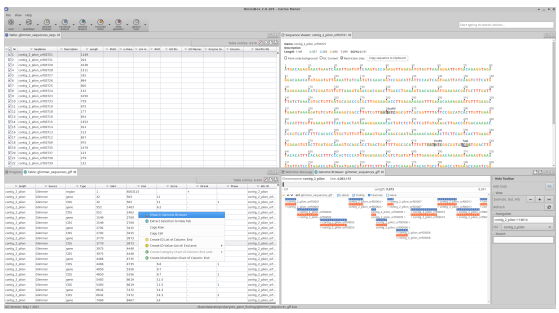


Figure 5: *How to open the Genome Browser*

A Result Viewer is also opened to display the name of each sequence present in the FASTA file, the number of genes per sequence, the minimum and maximum gene length, and the strand position of the genes found (Figure 6).

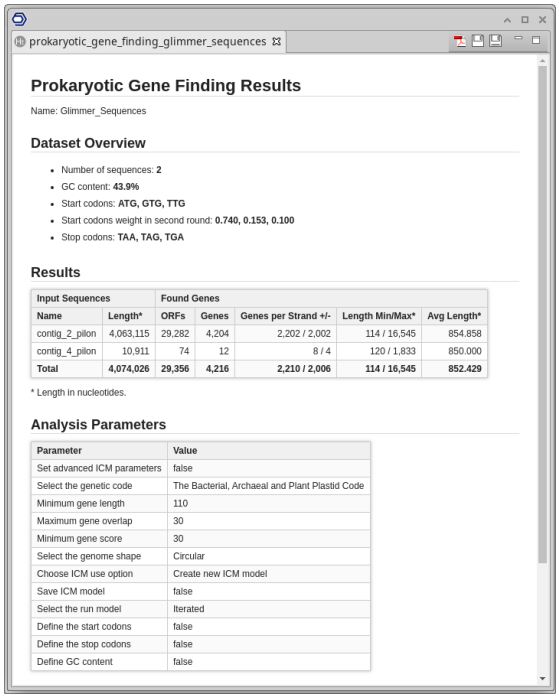


Figure 6:*Summary Report*

4.2.9 Multi-Locus Sequence Typing (MLST)

Introduction

Multi-locus sequence typing (MLST) is a useful tool for studying the genetic diversity of important public health pathogens that have provided a portable and reproducible typing system. It is a nucleotide sequence-based approach of an unambiguous procedure of characterizing isolates of bacterial species using the sequences of internal fragments of (usually) seven housekeeping genes. For this, approx. 4 450-500 bp internal fragments of each gene are used, as these can be accurately sequenced on both strands using an automated DNA sequencer. For each housekeeping gene, the different sequences present within a bacterial species are assigned as distinct alleles and, for each isolate, the alleles at each of the seven loci define the allelic profile or sequence type (ST).

For more info please click here.

Please cite MLST as:

Larsen MV et al. (2012). Multilocus sequence typing of total-genome-sequenced bacteria. Journal of clinical microbiology, 50(4), 1355-61.

Run MLST

This functionality can be found under **Genome Analysis → Multi-locus Sequence Typing (MLST)**. The wizard allows to select files and set the parameters (Figure 1 and Figure 2).

INPUT

- **Input Data Files:** Choose between Assembled or Draft Genome/Contigs (FASTA) or Raw Sequencing Reads (FASTQ). Single-end or paired-end reads can be used when selecting Raw Sequencing Reads (FASTQ). Note that if paired-end is selected, two files per sample are required.
- **Paired-end configuration:** In the case of paired-end reads, the pattern to distinguish upstream files from downstream files is required. The provided patterns are searched right before the extension, and the start of the name should be the same for both files of each sample. Files whose names match with upstream and downstream patterns will be treated as paired-end data. The remaining files and those for which no partner is detected will be treated as single-end data.
- **Upstream Files Pattern:** Establish the pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern:** Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is named SRR3666079_1.fastq and the downstream one SRR3666079_2.fastq, you should establish "_1" as the upstream pattern and "_2" as the downstream pattern.

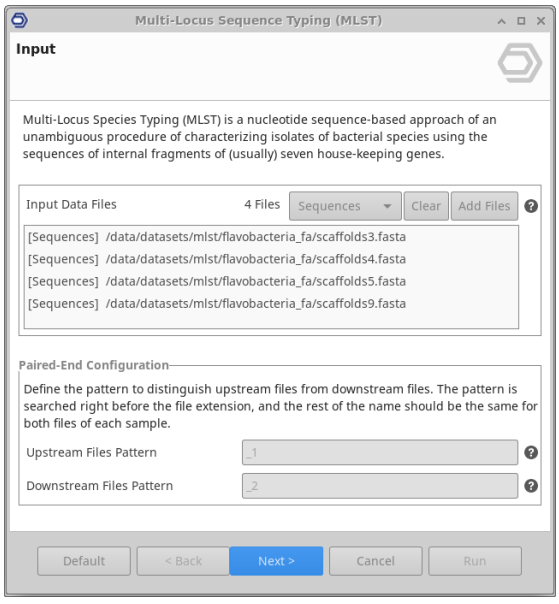


Figure 1: *Input Data Page*

CONFIGURATION

- **MLST Configuration:**Select the species database that will be used as a template for MLST prediction. If a wrong species is selected, the run may fail or the output will show no (zero) or minimal identity and coverage. MLST allele sequence and profile data are obtained from PubMLST.org.

For four organisms, two or three different MLST schemes are available. These are:

1. Acinetobacter baumannii: (Acinetobacter baumannii #1, Acinetobacter baumannii #2)
2. Escherichia coli: Escherichia coli #1, Escherichia coli #2)
3. Pasteurella multocida: (Pasteurella multocida #1 (RIRDC), Pasteurella multocida #2 (multihost))
4. Leptospira: (Leptospira #1, Leptospira #2, Leptospira #3)

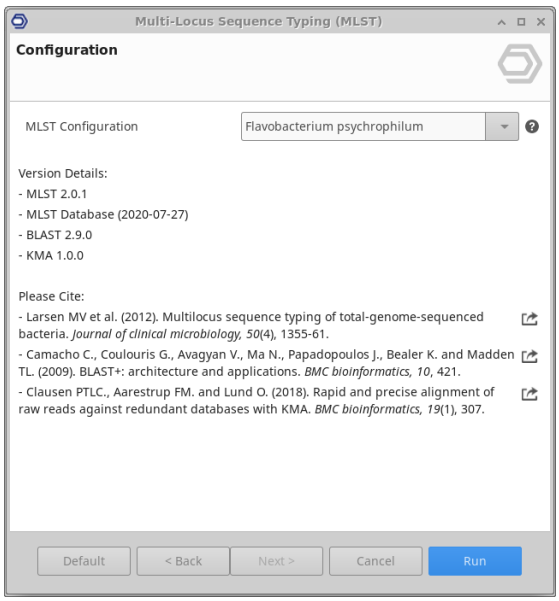


Figure 2: *MLST Configuration Page*

Results

When the MLST completes, it creates a sequence table containing the MLST results (Figure 3). This table will contain:

1. **Tags:** It contains a quick overview of the MLST result of your sample. It will generate three possible reports:
2. **Matched:** When a complete match was found with no errors or SNPs, therefore the average identity between all the housekeeping genes templates and the query reads/sequences, as well as the average coverage, is 100%. All samples with "Matched" results will be highlighted in green.
3. **Partial:** When partial matches were found the average identity between all the housekeeping genes templates and the query, as well as the average coverage, is less or equal to 99%. It happens when some potential errors or SNPs have been detected. All samples with "Partial" results will be highlighted in orange.
4. **No Matched:** The query sequence did NOT match any housekeeping gene template within the chosen MLST configuration. All samples with "Not Matched" results with be highlighted in red.
5. **Name:** It displays the input file name.
6. **Sequence Type:** It contains the corresponding MLST sequence type. Please note that for all "Partial" results, the sequence type will have a number and an asterisk. This asterisk is to indicate that the Sequence Type number shown here is not a 100% match and alleles with discrepancies will be indicated in the "Note" column.
7. **Average Identity:** All query reads/sequences that match a housekeeping gene template sequence in the database will return the percentage identity of the alignment. This percentage identity obtained for each housekeeping gene will be averaged and this average will be displayed in this column.
8. **Average Coverage:**All query reads/sequences that match a housekeeping gene template sequence in the database will return the percentage coverage of the alignment. This percentage coverage obtained for each housekeeping gene will be averaged and this average will be displayed in this column.
9. **Notes:** It contains important information relevant to the sequence type result generated. "Matched" results will NOT any information in this column, however, "Partial" results will display all the alleles with discrepancies. This discrepancy may indicate that a novel allele was found, errors or SNPs. A detailed report containing the nucleotide(s) differences and location within the alleles can be found in the "MLST Alignment Report". "No Matched" results will indicate that no MLST loci were found in the input data, make sure that the correct MLST scheme was chosen.

Sequence Type: Unknown

Please note that "Unknown" can be the Sequence Type for samples with "Matched" result and the reason for this is that even though all the alleles in the query are matching 100% of alleles in templates sequences in the database, the combination of the alleles does not have an MLST number assigned yet. In the case of samples with "Partial" results, the "Unknown" Sequence Type is not because of the discrepancies, but because the combination of the alleles does not have an MLST number assigned yet either. Lastly, for samples with "Not Matched" results, the "Unknown" Sequence Type is because there was not an MLST locus that matched with the input data. This is most common when the wrong MLST scheme was chosen in the MLST configuration wizard page.

Some "Unknown" results could also report a "Nearest ST..." if there is enough coverage and identity found between the query reads/sequences and any template sequence in the database. Figure 3 shows an example of this case in sample name "scaffolds4", in which the sequence type is reported as "Unknown, Nearest ST: 34, 196".

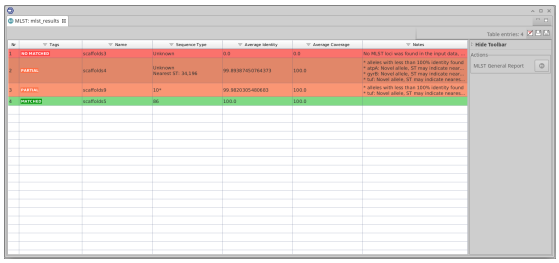


Figure 3: *Results Table Page*

When the MLST completes, it also creates an **MLST Report**. This contains the information relevant to the MLST run, including the input data, MLST configuration used, MLST results, and the parameters used for the analysis (Figure 4).

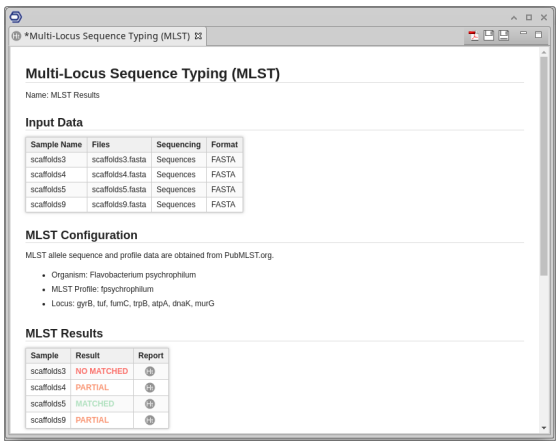


Figure 4: *MLST Report Page*

An **MLST Results** report will be generated with sample or file name and all the different housekeeping genes sequences found in the query reads/sequences (Figure 5). To access this report, right-click on the row of the sample, and select the "Show MLST Result" option.

The **MLST Result** report contains the information relevant to that specific sample or input file. This includes the input data, MLST configuration used, MLST results, and the parameters used for the analysis. The MLST results section contains a table with the **locus**, which is the name of the housekeeping gene that the query reads/sequences have been aligned to; the **identity**, which refers to the percentage of the query reads/sequences that matched a template sequence in the database; the **coverage**, which refers to how much of the template sequence in the database has been covered by the query reads/sequences; the **alignment length**, which refers to the total number of nucleotides between the query reads/sequences that have aligned against a template sequence in the database; the **allele length**, which refers to

the total number of nucleotides in the allele or template sequence in the database; **gaps**, this will indicate if any gaps or deletions have been detected; and lastly, the **allele**, which refers to the name of the housekeeping gene or sequence in the database.

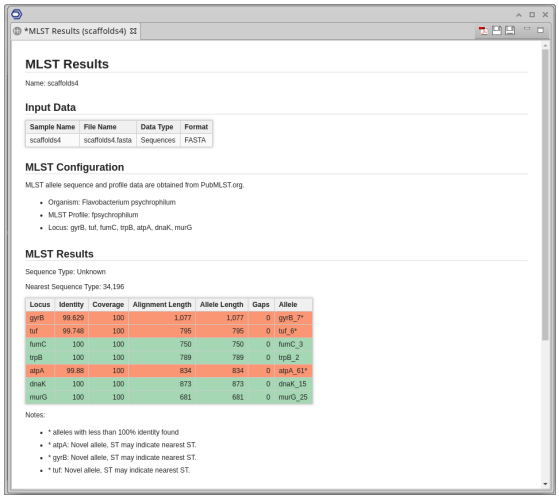


Figure 5: *MLST Results Page for a Specific Sample or Input File*

An **MLST Alignments** report will be generated with sample or file name and all the different housekeeping genes sequences found from the query reads aligned against the housekeeping genes template sequences (Figure 6). To access this report, right-click on the row of the sample, and select the "Show MLST Alignment Report".

The **MLST Alignment**report contains a detailed and colored report of the alignments. In this report, the alignment between the query reads/sequences and the template sequence in the database is divided by each allele detected. Alleles can be identified by the 'pound sign' (#) in front of the allele name. Discrepancies are highlighted.

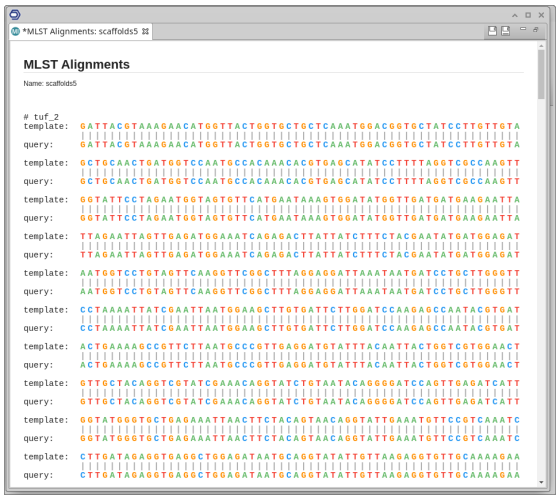
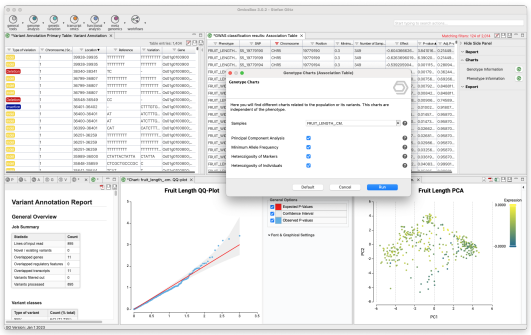


Figure 6: *MLST Alignment Report Page*

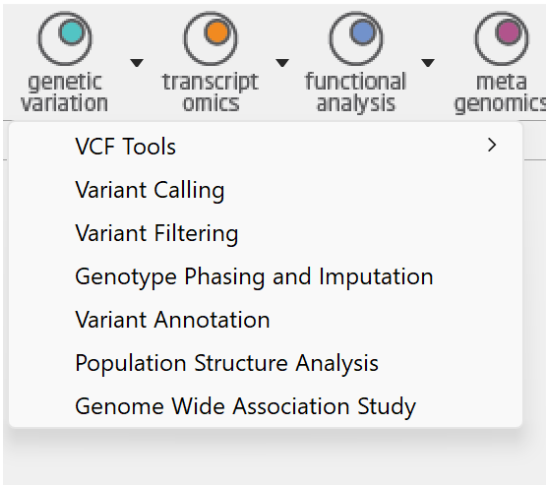
4.3 Module Genetic Variation

4.3.1 Module Genetic Variation



The Genetic Variation Module allows to Identify and analyze genetic variations within a population or a species. Cutting-Edge Genetic Variation Analysis.

- **Fast Variant Calling:** Choose between two of the best algorithms to detect variants in your alignment files: BCFtools mpileup and Freebayes. With OmicsBox, you not only receive a VCF file but also gain the ability to visualize various charts for quality control of your results.
- **Model & Non-Model Variant Annotation:** Gather information about the coding and genetic consequences of all the variants included in a VCF file with Ensembl's Variant Effect Predictor. You only require the reference genome used in the Variant Calling step and the annotation file.o
- **End-to-end Analysis:** Create your VCF files with a Variant Calling algorithm, merge or extract information, and filter your VCF files. Then, you will be able to phase and impute your variants and perform a downstream analysis such as GWAS or Population Structure Analysis.
- **Variant Filtering:** Remove any variants from the VCF file that do not meet our quality standards across different fields.
- **Guided Genome-Wide Association Studies:** Associate variants to different quantitative traits using GAPIT3 R package in a guided and easy way.
- **Supports GBS and WGS data:** All these methods can be used with either Genotype By Sequencing (GBS) data to simplify the genome being sequenced, or with Whole Genome Sequencing (WGS) to comprehensively identify variants across the entire genome.



4.3.2 VCF Tools

VCF Tools

Genetic Variation experiments generate a high volume of data, making it necessary to merge VCF files. This tool allows for the combination of diverse VCF files, facilitating the consolidation of genetic variant information from multiple sources. Additionally, OmicsBox offers a tool to extract features from a VCF file for more targeted analysis in order to focus on specific samples or chromosomes within the genome.

Merge VCFs

INTRODUCTION

Due to the high volume of data that Genetic Variation experiments handle nowadays, a tool to merge different VCF files has become necessary in OmicsBox. With this utility you will be able to seamlessly combine diverse VCF (Variant Call Format) files. This tool might be useful when you need to consolidate genetic variant information from multiple sources, such as various experiments or datasets. By merging different VCF files you will find easier to analyze and interpret genetic variation data.

MERGE VCFS

The tool to merge VCFs can be found in the Genetic Variation Module of OmicsBox under **VCF Tools → Merge VCFs.**The wizard consists of two pages and allows you to define the input and output options as well as different options to merge VCF files (Figure 1, Figure 2).

Input

In the first page you will be able to select the input files and how to merge files.

- **VCF Files:** select VCF files to merge in one single VCF file.
- How to merge files:
- **By sample:** choose this option if you intend to generate a unified VCF file by combining samples from various VCF files, each containing different subsets of samples.
- **By chromosome:** select this option if you want to create a single VCF file from multiple VCF files with different chromosomes.

The first option could be appealing for generating a single VCF file in scenarios where, for instance, you have executed a Variant Calling job with certain samples and subsequently you obtained another VCF file containing a distinct set of samples from the identical dataset.

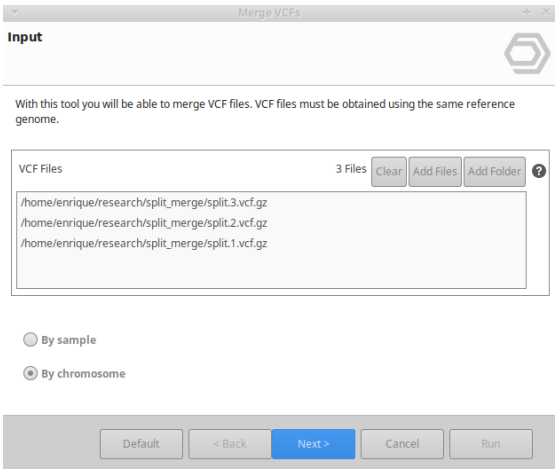


Figure 1. *Input Page*

Output

- **VCF File:** specify the folder where you want to save the final VCF File.

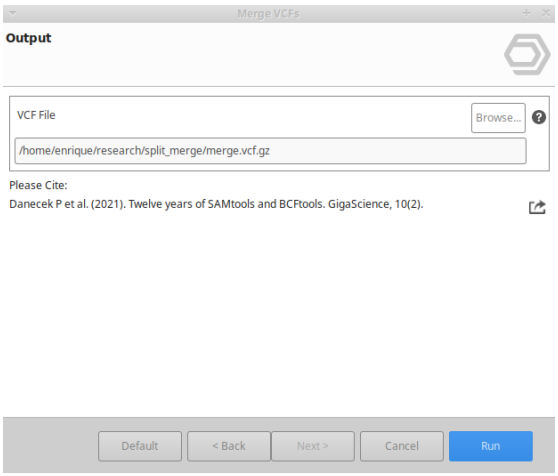


Figure 2. *Output Page*

Summary Report

Appart from a VCF file that is the result of the merge of all the input VCF files, a Summary Report will appear. This report will have the following information:

- **Input Data:** file names of all the VCF used as input.
- **Summary Information:**
- Types of Variants: frequency of the different types of variants.
- Number of alleles in a variant: abundance of the alleles per variant.
- Statistics: number of total variants, number of total genotypes, number of heterozygotes and missing data.

Genetic Variation Tools Report

Input Data

VCF Files:

- not_selected.vcf.gz
- selected.vcf.gz

Summary Information

Type of variant	Frequency
SNP	6970
MIXED	70
MNP	1482
INDEL	731

Number of alleles in a variant	Frequency
2	8730
3	392
4	60
5	18
6	17
7	7
8	12
9	4

Extract VCFs

INTRODUCTION

Sometimes, although you have a huge whole-genome VCF file with a lot of samples, you just want to analyze only some samples or you want to focus on some chromosomes of the genome. Because of that, we have added a tool to extract features from a VCF to OmicsBox.

EXTRACT VCF

The tool to extract features from a VCF can be found in the Genetic Variation Module of OmicsBox under **VCF Tools → Extract from VCF.**The wizard consists of two pages and allows you to define the input and output options as well as different options to extract information from VCF files (Figure 1, Figure 2).

Input

In the first page you will be able to select the input file and what information you want to extract.

- **VCF File:** select the VCF file from which to extract information.
- Features to extract by:
- **Samples:** choose this option if you intend to generate a VCF file with the same number of variants as the original one but with only certain samples.
- **Chromosomes:** select this option if you want to create a VCF file with all the samples as the original file but with variants from selected chromosomes.
- **Features to extract:** select the samples/chromosomes that you want to extract.
- **Get Also Unselected Features:** select this option if you want to obtain not only a VCF file with the selected features but also another one with the opposite subset of features.

The last option might be interesting in the case that you have a lot of chromosomes/samples and you want to extract the majority of them. As selecting all of them might be a bit arduous, you can select the features you are not interested in and then check this option.

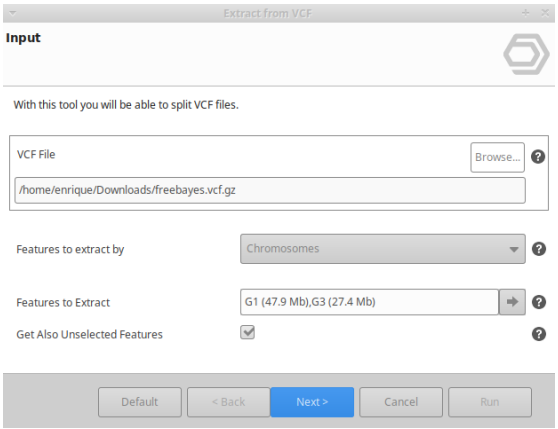


Figure 1. *Input Page*

Output

- **VCF File:** specify where to save the VCF file with the extracted features.

- **Complementary VCF File:** specify where to save the VCF file with the complementary set of features.

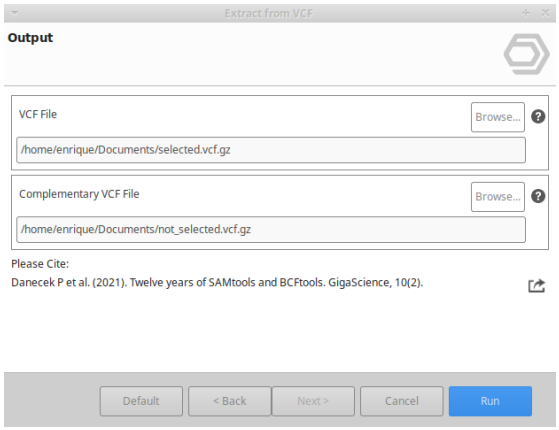


Figure 2. *Output Page*

Summary Report

Appart from the VCF file(s), a Summary Report will appear. This report will have the following information:

- **Input Data:** file names of all the VCF used as input.
- **Summary Information:**
- Types of Variants: frequency of the different types of variants.
- Number of alleles in a variant: abundance of the alleles per variant.
- Statistics: number of total variants, number of total genotypes, number of heterozygotes and missing data.

Genetic Variation Tools Report

Input Data

VCF File: varitome_all.vcf.gz

Summary Information

Type of variant	Frequency
SNP	5414599
INDEL	680874

Number of alleles in a variant	Frequency
2	6095473

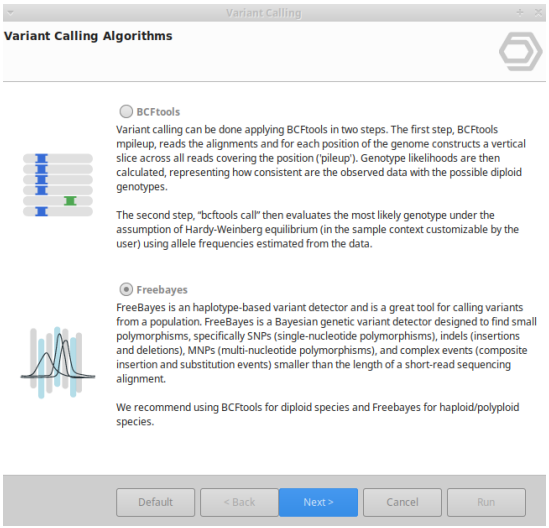
Statistics	Count (% total)
Number of Variants	6095473
Number of Genotypes	926511896
Number of Heterozygotes	9454565 (1.02%)
Missing Data	17367184 (1.87%)

4.3.3 Variant Calling

Variant Calling

Variant calling is the process by which we identify variants (such as SNPs, insertions or deletions) from sequence data. This data, which is stored in a VCF file, can be used to annotate variants, if you are interested in the genetic and coding consequences of each mutation, and to associate variants to phenotypic traits, in order to know if a variant is significantly related to a characteristic. There are different algorithms specifically designed to achieve this goal:

- **BCFtools**: this is a widely-used variant calling tool, especially among non-human species, which is characterized by its small time of execution and its precision.
- **Freebayes**: this tool is characterized by its capability to use it with polyploid genomes.



Variant Calling using BCFtools

INTRODUCTION

BCFtools is a widely-used variant calling tool, especially among non-human species, which is characterized by its small time of execution and its precision.

BCFtools uses two algorithms. The first one is called *mpileup*. This algorithm reads the alignments and, for each position of the genome, constructs a vertical slice across all reads covering the position ("pileup"). Genotype likelihoods are then calculated, representing how consistent are the observed data with the possible diploid genotypes. The calculation takes into account mapping qualities of the reads, base qualities, and the probability of local misalignment, per-base alignment quality (BAQ), and the user can set thresholds for each of these parameters. The second step, '*bcftools call*', then evaluates the most likely genotype under the assumption of Hardy-Weinberg equilibrium.

Please cite BCFtools as:

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., ... & Li, H. (2021). Twelve years of SAMtools and BCFtools. *Gigascience*, *10*(2), giab008.

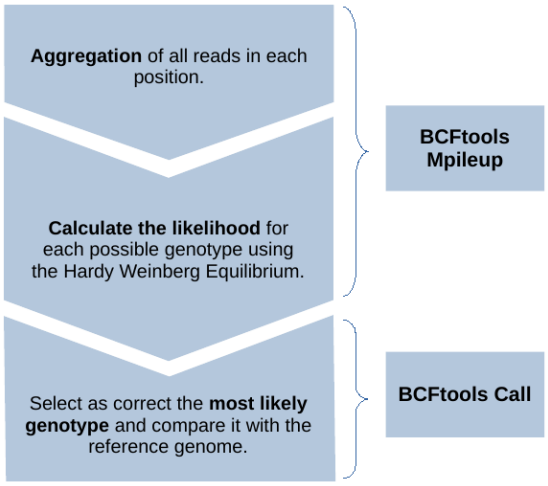


Figure 1. *Workflow of the BCFtools Mpileup-Call Pipeline*

RUN BCFTOOLS FOR VARIANT CALLING

BCFtools can be found under **Genetic Variation → Variant Calling → BCFtools.**The wizard consists of 4 pages and allows to define the input and output options as well as the analysis parameters (Figure 2, Figure 3, Figure 4 and Figure 5).

Input

- First of all, this tool requires two types of necessary files:
- **BAM files:** alignment files in BAM format. To obtain them, you must align FASTQ files using a DNA-Seq Alignment Strategy, like BWA or Bowtie 2.
- **Reference Genome:** FASTA file with the reference genome.

• Make sure that read alignment was executed using the same reference genome as the one that is used here as input.

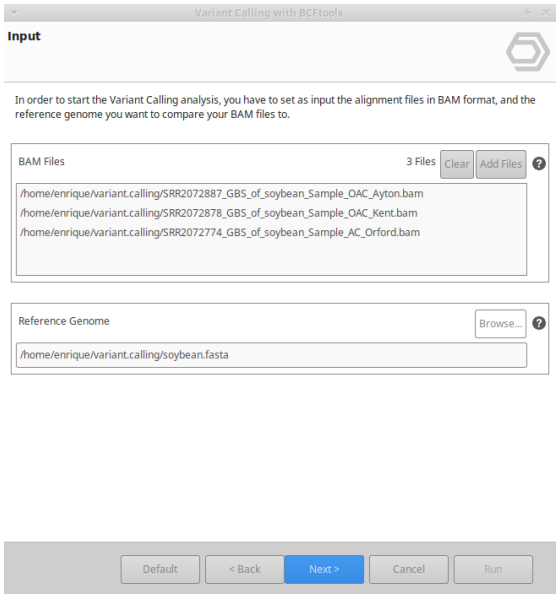


Figure 2. *Input Page*

Configuration 1

In this page, you can set the option of BAM Preprocessing using Picard and **basic parameters** for the mpileup step.

- **Remove Duplicates:** Mark this option if you have Whole Genome Sequencing or Whole Exome Sequencing in order to remove PCR duplicates. For GBS or RADSeq dataset, this option is not recommended.
- **Adjust Mapping Quality:** coefficient for downgrading mapping quality for reads containing excessive mismatches. This parameter is disable by default (0 value).
- **Maximum Depth:** maximum raw per-file depth. By default, this value is set to 250, as this depth is enough to achieve good results. In this case, if the coverage depth is higher than this threshold, the algorithm will pick up 250 alignments at random. Nevertheless, if you want to tune this value higher, keep in mind that the time of execution will also be higher.
- **Minimum Mapping Quality:** minimum mapping quality for an alignment to be used. The Mapping Quality quantify the probability that a read is misplaced.
- **Minimum Base Quality:** minimum base quality for a nucleotide to be used. This base quality is the Phred score calculated at each position by the sequencing machine.
- **Ignore @RG Tags:** in a BAM file, the RG tag means Read Group. A Read Group is a set of alignments that come from the same sample. If this option is checked and BAM files have different RG values, all alignments in a BAM files are treated as from the same sample.
- **BAQ options.** These options are related to the realignment of some sequences in order to check the per-Base Alignment Quality (BAQ). BAQ evaluates the probability that there is a misalignment in each base:
- **No BAQ:** disable realignment. This option is recommended as it helps to reduce false positives.
- **Redo BAQ:** recalculate BAQ values in problematic regions.
- **Full BAQ:** redo BAQ in all positions.

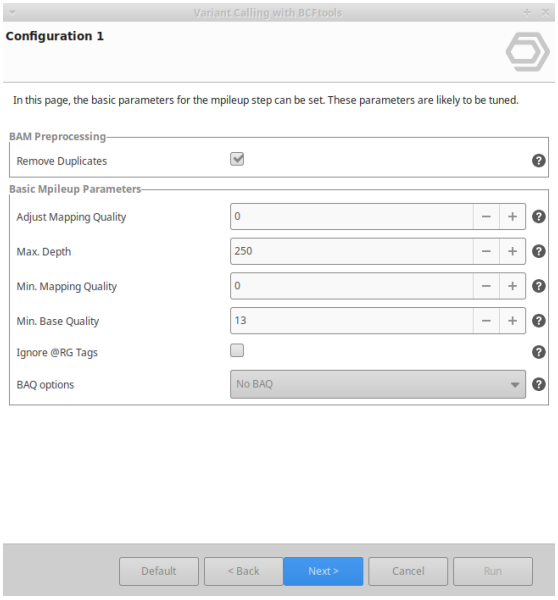


Figure 3. *Configuration 1 Page*

Configuration 2

In this page, you can adjust the **advanced parameters** for the mpileup step. These parameters do not have great consequences in the output.

- **Extension Error Probability:** phred-scaled gap extension sequencing error probability. Reducing this value leads to longer indels.
- **Minimum Fraction of Gapped Reads:** threshold of the proportion of reads that have a gap in order to call an indel in a position.
- **Tandem Quality:** coefficient for modeling homopolymer errors. A higher value of this parameter means a higher reliance on indels in homopolymers. Nevertheless, this parameter will not affect in a high degree to your results, although if you want a higher reliance on SNPs rather than in indels, set a smaller value.
- **Skip Indel Calling:** if this parameter is check, the VCF will only contain SNPs. The time of execution will also be shorter.
- **Gapped Reads for Indel:** number of reads necessary to call an indel.
- **Phred Open Sequencing Error:** phred-score gap opening error. If you set a smaller value, more indels will be called.

In addition, some **parameters for the call step** can also be set:

- **Keep Alternate Alleles:** keep all alternate alleles, even those that appear in the mpileup step but are not called in any of the genotypes in the second step.
- **Use Groups:** this option allows to group samples into populations and apply the Hardy Weinberg Equilibrium within population. If this option is disabled, the Hardy Weinberg Equlibrium is applied within all samples as one big population. By using this option, you gain power on SNPs shared between samples but lose power on singleton SNPs. We recommend to use this option only if you have a low-coverage dataset and you are going to do GWAS.
- **Group Experiment File:** tab-delimited file with no header and two columns. The first column has sample names and the second one has population names.

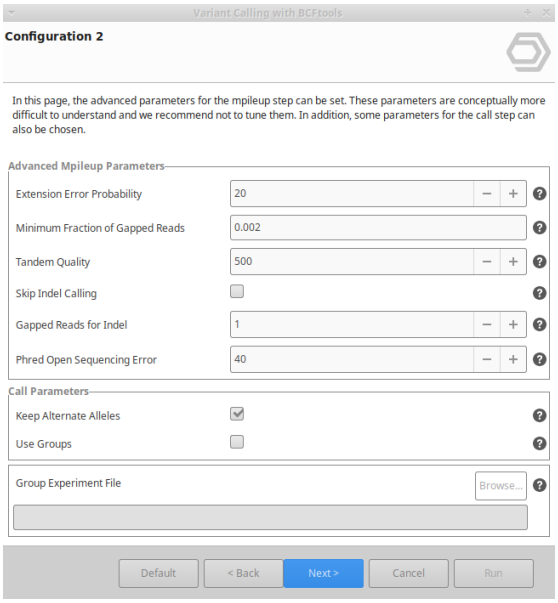


Figure 4. *Configuration 2 Page*

Output

- **VCF File**: filename of the resulting VCF file.

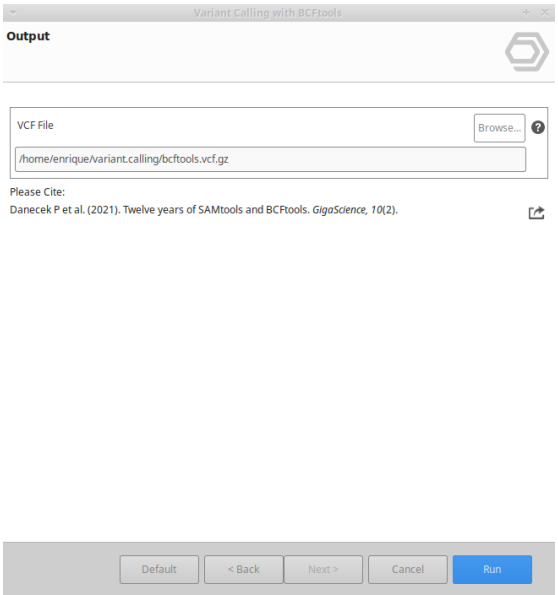


Figure 5. *Output Page*

RESULTS

Variant Calling has the following outputs:

- **VCF file** with all the variants that were found.
- **Report** with summary details:
- Input data: name of the input reference file and number of BAM files used.
- Information about the resulting VCF: information about the types of variant found and the number of alleles per variant.
- Adjusted parameters: as you might want to repeat the variant calling with other parameters, it is important to keep this table for reproducibility.

Variant Calling Report (BCFtools)

Input Data

Reference: GCF_000004515.6_Glycine_max_v4.0_genomic.fna.gz

Number of BAM Files Used: 3

Results

VCF File Saved as: /home/enrique/OmicsBoxWorkspace/soybean_remove.vcf.gz

Type of variant	Frequency
SNP	21784
INDEL	1833

Number of alleles in a variant	Frequency
2	23368
3	212
4	37

Statistics	Count (% total)
Number of Variants	23617
Number of Genotypes	70851
Number of Heterozygotes	4845 (6.84%)
Missing Data	31424 (44.35%)

Parameters

Parameter	Value
Remove Duplicates	false
Adjust Mapping Quality	0
Max. Depth	250
BAQ options	No BAQ

Figure 6. *Summary Report from BCFtools.*

- **Distribution charts** of different quality variables found in the VCF. This charts might be important to know how to filter the VCF subsequently:
- Raw Read Depth Histogram: in BCFtools, Raw Read Depth (the DP field) means the number of times that site was read no matter the nucleotide it is in that position (the sum of the reference nucleotide and other variants).

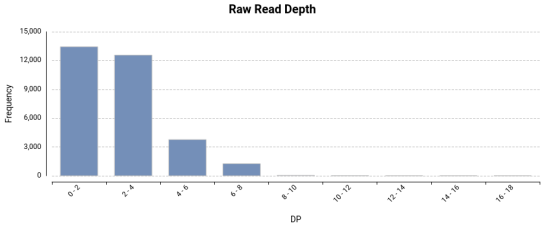


Figure 7. *Raw Read Depth Histogram.*

1. Proportion 'Quality/Depth' Histogram: the quality column in VCF files generated using BCFtools is the Phred-scaled probability that the site has no variant. Nevertheless, it is better to rely on this quality normalized by depth.

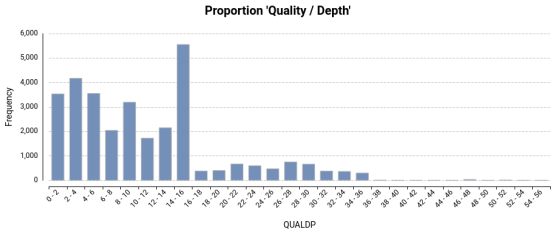


Figure 8. *Proportion 'Quality/Depth' Histogram.*

1. Average Mapping Quality Histogram: the MQ value in the info field of a BCFtools VCF file relates to the average of all mapping qualities of the reads supporting the variant.

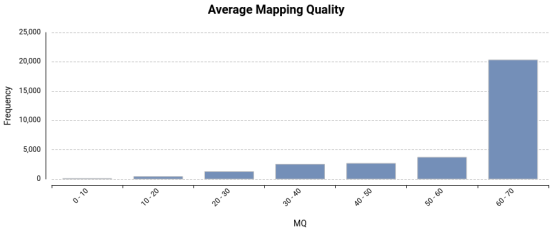


Figure 9. *Average Mapping Quality Histogram.*

Variant Calling using Freebayes

INTRODUCTION

Freebayes is a variant calling tool characterized by its capability to use it with polyploid genomes.

This algorithm is a Bayesian genetic variant detector designed to find small polymorphisms, specifically SNPs (single-nucleotide polymorphisms), indels (insertions and deletions), MNPs (multi-nucleotide polymorphisms), and complex events (composite insertion and substitution events) smaller than the length of a short-read sequencing alignment.

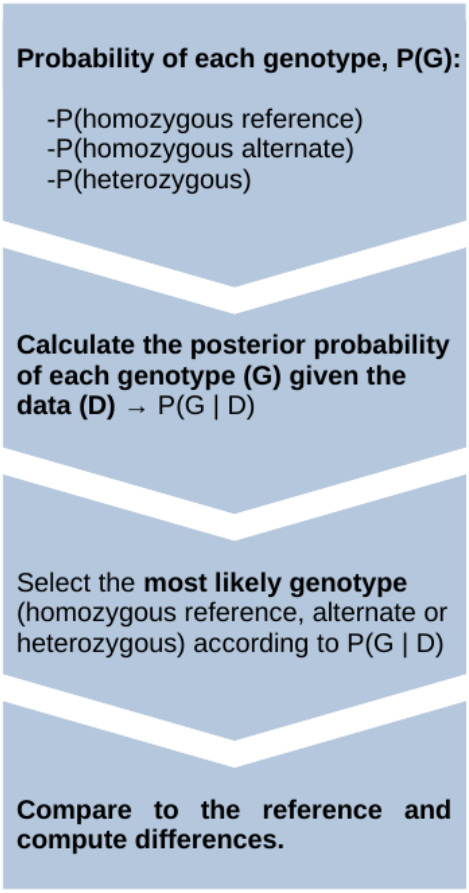


Figure 1. *Freebayes Workflow*

RUN FREEBAYES FOR VARIANT CALLING

Freebayes can be found under **Genetic Variation → Variant Calling → Freebayes**.The wizard consists of 3 pages and allows to define the input and output options as well as the analysis parameters (Figure 2, Figure 3 and Figure 4).

Input

- **BAM files:** alignment files in BAM format. To obtain them, you must align FASTQ files using a DNA-Seq Alignment Strategy, like BWA (**highly recommended**) or Bowtie 2.
- **Reference Genome:** FASTA file with the reference genome.
- **Group Experiment File (optionally):** tab-delimited file with sample names in one column and population names in another. If this file is added, the population-based bayesian inference model will then be partitioned on the basis of the populations.

Make sure that read alignment was executed using the same reference genome as the one that is used here as input.

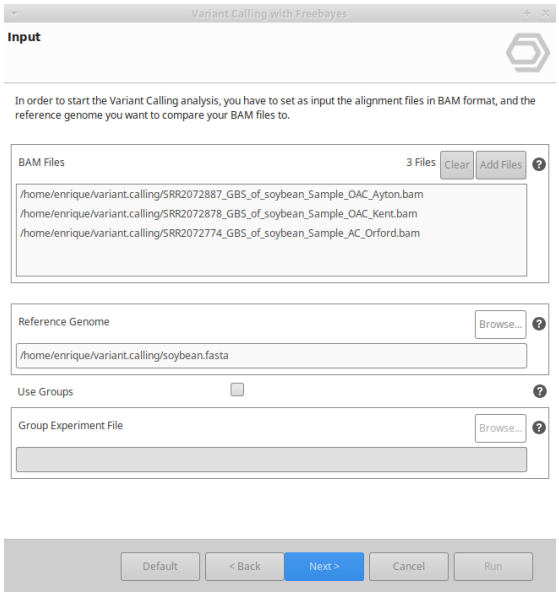


Figure 2. *Input Page*

Configuration 1

In this page, the preprocessing step using Picard and ploidy parameters for Freebayes can be set.

- **Remove Duplicates:** mark this option if you have Whole Genome Sequencing or Whole Exome Sequencing in order to remove PCR duplicates. For GBS or RADSeq dataset, this option is not recommended.
- **Samples with Mixed Ploidy:** check this option if you want to perform variant calling in samples with different ploidy (e.g., a diploid and an hexaploid sample).
- **Ploidy:** sets the species ploidy for the analysis. This option will be only enabled when you are not going to perform mixed-ploidy variant calling.
- **Copy Number Variation File:** this text file consists of two columns with no header. In the first column, the sample name of each individual (i.e., the BAM file name without the ".bam" extension) must appear, and in the second one, the copy number must be shown (just a number, e.g., 1 for haploids, 2 for diploids, 3 for triploids, etc.)
- **Calculate Genotype Quality:** genotype Quality in Freebayes is the likelihood that a genotype is correct. Although it is recommended to let this parameter in true, please consider to switch it to false when you are doing polyploid variant calling (specially with hexaploids) and with several samples (more than 10).
- **Minimum Alternate Fraction:** require at least this fraction of observations supporting an alternate allele within a single individual in order to evaluate the position.
- **Minimum Alternate Count:** the same as before but in absolute numbers.
- **Minimum Alternate Quality Sum:** require at least this count of observations supporting an alternate allele within the total population in order to use the allele in analysis.

In polyploid variant calling, it is recommended to set higher thresholds for Min. Alternate Fraction and Count. Alternatively, Min. Alternate Quality Sum can be raised too, which may be more flexible.

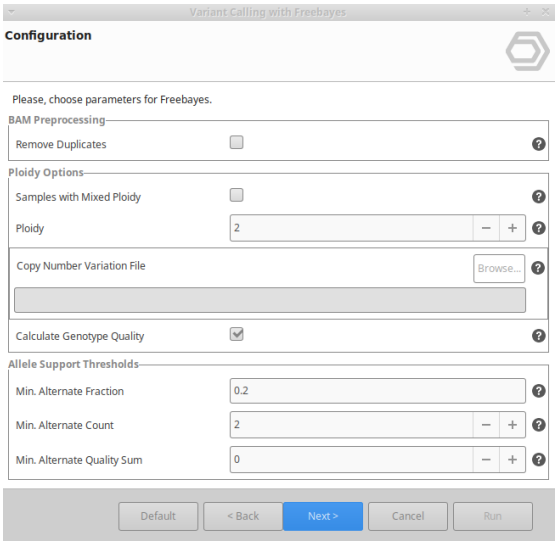


Figure 3. *Configuration 1 Page*

Configuration 2

In this page, other parameters for Freebayes can be set.

- **Minimum Mapping Quality:** exclude alignments for the analysis if they have less than this value of mapping quality.
- **Minimum Base Quality:** exclude alleles for the analysis if they have less than this value of base quality.
- **Minimum Allele Quality Sum:** exclude alleles for the analysis if the sum of the base quality of the supporting observations is lower than this value.
- **Minimum Allele Mapping Quality Sum:** exclude alleles for the analysis if the sum of the mapping quality of the corresponding alignments of the supporting observations is lower than this value.
- **Mismatch Base Quality:** base quality to call a mismatch.
- **Minimum Coverage:** coverage needed to process a site.
- **Maximum Coverage:** downsample per-sample coverage to this level if it is greater than this coverage.
- **Use Mapping Quality:** use mapping quality of alleles when calculating data likelihoods.
- **P-value:** report sites if the probability that there is a polymorphism at the site is greater than N. Note that post-filtering is generally recommended over the use of this parameter.

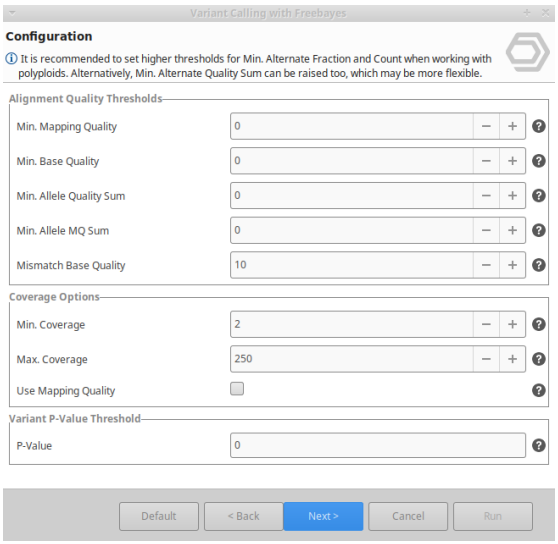


Figure 4. *Configuration 2 Page*

Output

- **Set Name for VCF:** VCF filename.
- **Directory to Save the VCF:** directory to save the VCF file.

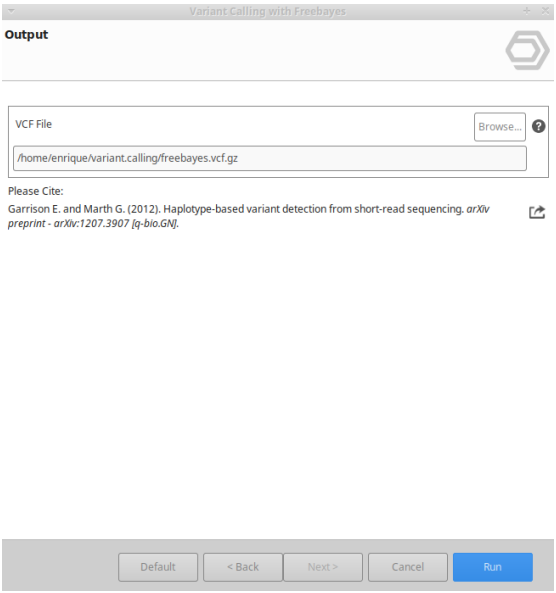


Figure 5. *Output Page*

RESULTS

Variant Calling has the following outputs:

- **VCF file** with all the found variants.
- **Report** with summary details:
- Information about the resulting VCF: information about the types of variant found and the number of alleles per variant.
- Adjusted parameters: as you might want to repeat the variant calling with other parameters, it is important to keep this table for reproducibility.

Just in case you repeat the Variant Calling Analysis with BCFtools, please keep in mind that Freebayes is able to separate MNPs from SNPs, although BCFtools is not able to do it, and MNPs are registered as different SNPs. Nevertheless, it is no of great importance.

Variant Calling Report (Freebayes)

Input Data

Reference: GCF_000004515.6_Glycine_max_v4.0_genomic.fna.gz

Number of BAM Files Used: 3

Results

VCF File Saved as: /home/enrique/Downloads/freebayes_duplicates.vcf.gz

Type of variant	Frequency
SNP	2400
MIXED	1
MNP	129
INDEL	186

Number of alleles in a variant	Frequency
2	2711
3	5

Statistics	Count (% total)
Number of Variants	2716
Number of Genotypes	8148
Number of Heterozygotes	127 (1.56%)
Missing Data	0 (0.00%)

Parameters

Parameter	Value
Use Groups	false
Remove Duplicates	true
Ploidy	2

Figure 6. *Freebayes Summary Report*

- **Distribution charts** of different quality variables found in the VCF. This charts might be important to know how to filter the VCF subsequently:

Depth Histogram: In Freebayes, Depth (the DP field) means the total read depth at the locus, that is to say, the number of times that site was read, but not necessarily that variant (also the reference nucleotide, or other variants).

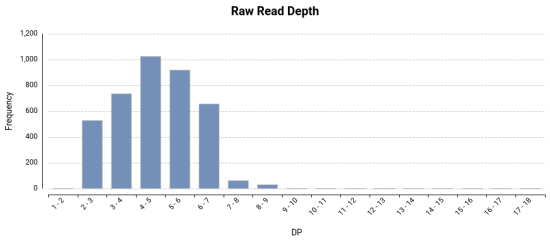


Figure 7. *Depth Histogram*

Proportion 'Quality/Depth' Histogram: the quality column of VCF files is the Phred-scaled probability that the site has no variant. Nevertheless, it is better to rely on this quality normalized by depth.

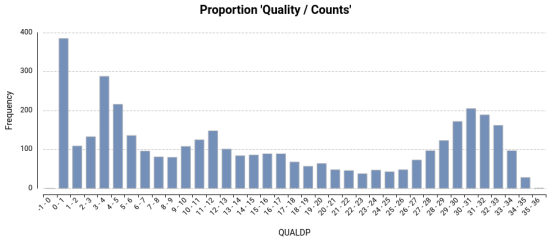


Figure 8. *Proportion 'Quality/Dept' Histogram*

Mapping Quality in Alternate Alleles Histogram: the MQ value in the info field of a BCFtools VCF file relates to the average of all mapping qualities of the reads supporting the variant.

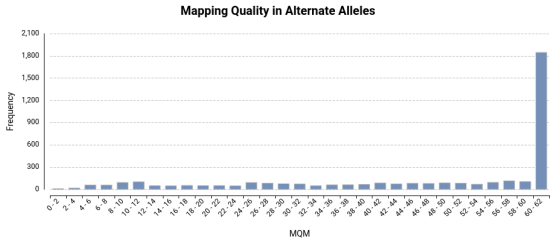


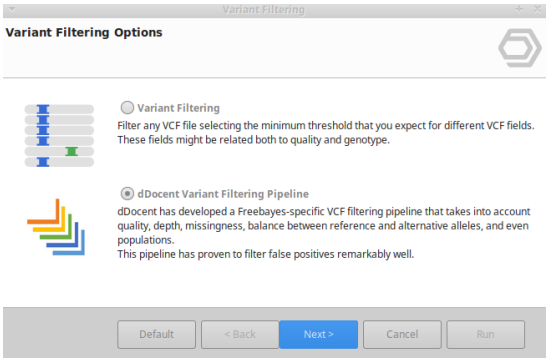
Figure 9. *Mapping Quality in Alternate Alleles Histogram*

4.3.4 Variant Filtering

Variant Filtering

Variant filtering can be used to remove variants that are not reliable. This filtered dataset of variants will lead to more reliable and robust results in downstream analysis. Moreover, as this dataset will be smaller, all analysis will go faster. There are two ways to filter variants in OmicsBox:

- **General Variant Filtering**: with this simple filter you will select different threshold for the main
- **dDocent Variant Filtering Pipeline**: dDocent has a pipeline to filter VCF files whose variants were sequenced using a RAD-Seq or GBS-like protocol.



General Variant Filtering

INTRODUCTION

Variant filtering is a secondary analysis operation that follows the Variant Calling step consists of identifying highly confident variants and removing the ones that are more likely to be falsely called. To filter out those false variants, the user must select the threshold considered to be adequate for different information fields of the VCF. Keep in mind that this step is crucial to avoid the analysis of false positives, what leads to the speed up of every subsequent step because less variants are being studied.

For VCF files generated either with BCFtools or Freebayes, the user is able to put different thresholds in parameters like quality, depth, average mapping quality and quality normalized by depth. In addition, the user can remove those SNPs with more than one variant (multiallelic variants), which might be interesting to perform an association analysis. Moreover, we have incorporated two additional Freebayes-specific parameters: the user can select variants with at least one read in each strand and/or variants that are supported by reads at both sides of the strand.

RUN VARIANT FILTERING

This tool can be found under **Genetic Variation → Variant Filtering**.The wizard consists of 2 pages and allows to define the input and the filtering parameters, and output options (Figure 1 and Figure 2).

Input and parameters

- **VCF file:** this file must come from a Variant Calling analysis.
- **Proportion 'Quality / Counts':** proportion between the Phred quality of the SNP and the count of full observations of alternate haplotypes. If there is more than 1 alternative haplotype, the mean is taken. This filter is more powerfull than using only the Phred quality or the counts of observations by their own.
- **Raw Read Depth:** total number of reads overlapping that position.
- **Phred Quality:** phred-scaled quality score for the assertion that alternative allele exists and it is not a sequencing error.
- **Average Mapping Quality (MQ):** mapping quality measures how unique that read is. That is to say, the probability that the read is misplaced.
- **Genotypic Options:**
- **Remove Variants with Multiple Alleles**: this is recommended if you are going to run a Genome-Wide Association Study.
- **Missing Genotypes per Variant:** maximum fraction of genotypes that can be missed out in a variant. For example, if you set this value to 0.1, only variants with at least 10 out of 100 genotypes will be available.
- **Genotype Depth (GD) Threshold:** minimum number of reads that might support that genotype.
- **Genotype Quality (GQ) Threshold:** it represents the Phred-scaled confidence that the genotype assignment is correct.
- **Minimum Allele Frequency Threshold:** Minimum Allele Frequency (MAF) represents the fraction of the least frequent allele in a population for a variant. Variants with a MAF smaller than this threshold will be filtered out.

If the population is just one individual (i.e., you only introduced one BAM file with aligned reads from one sample in the Variant Calling Step), **Genotype Quality** will be equal to **Phred Quality**, and **Genotype Depth** will be equal to Variant **Raw Read Depth**. You can just set "0" in that parameters in order to disable them.

- **Freebayes-specific Parameters:**
- **Check Reads in Both Strands**: check to verify if there is at least one read in each strand. It is recommended to check this parameter, as if a variant is real, it should have been discovered in both DNA strands.
- **Check if Reads are Balanced**: check if there are at least two reads 'balanced' to each side of the site (e.g. there is at least one read place right and another one place left of the variant).

Variant Filtering strongly depends on the genotyping protocol used to obtain the dataset. The main two experiments in this field are WGS for GWAS analysis and reduced-representation techniques such as GBS or RADseq.

Table 1. Recommended Parameters

Parameters	WGS	GBS or RADSeq
Quality / Counts	2	2
Raw Read Depth	10	2
Phred Quality	20	20
Average MQ	55	55
Remove Variants with Multiple Alleles	True	True
Missing Genotypes	0.6	0.1
GD Threshold	8	1
GQ Threshold	20	1
MAF Threshold	0.05	0.05

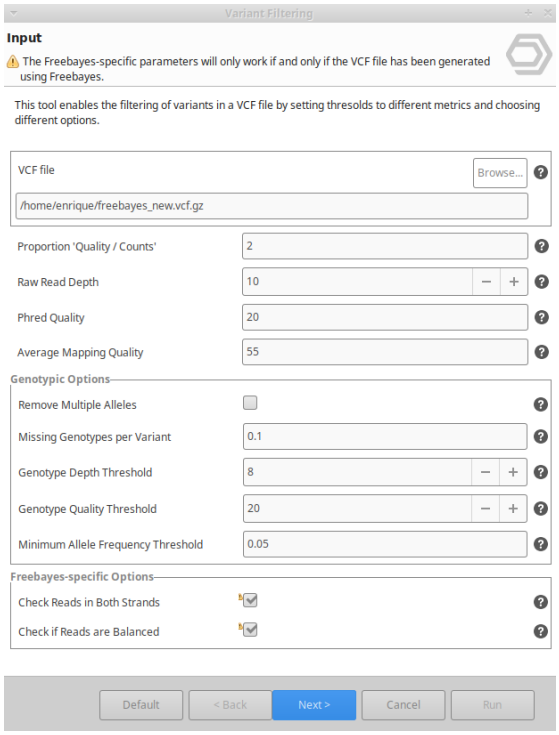


Figure 1. *Input and Parameters Page*

Output

- **Filtered VCF:** filename for the filtered VCF file.

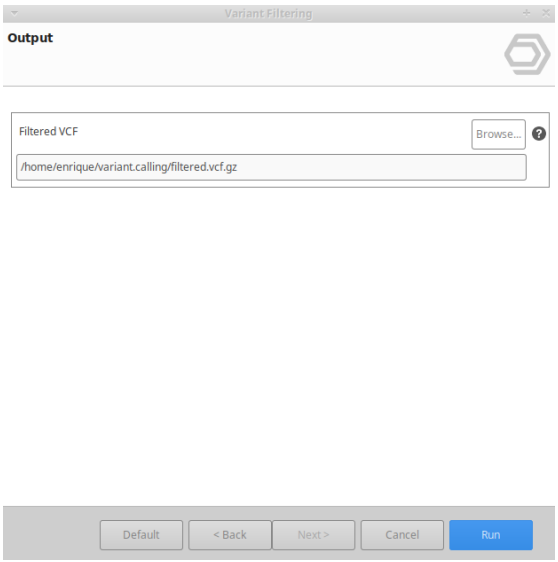
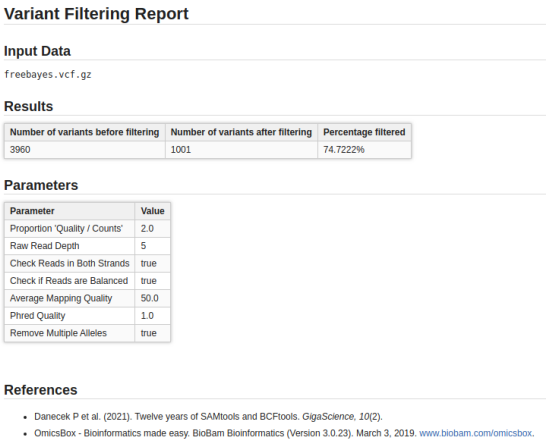


Figure 2. *Output Page*

RESULTS

Variant Filtering of BCFtools VCF files has the following outputs:

- **Filtered VCF file.**
- **Report** with information about the number of variants before and after filtering, and the used parameters.
- **Quality Control Charts:** these charts are the same as the ones that appear in the Variant Calling step. In addition, the Phred Quality distribution and the MAF distributuion are also added, as they are also set as filtering parameter.



Variant Filtering Report

Input Data

freebayes.vcf.gz

Results

Number of variants before filtering	Number of variants after filtering	Percentage filtered
3960	1001	74.7222%

Parameters

Parameter	Value
Proportion 'Quality / Counts'	2.0
Raw Read Depth	5
Check Reads in Both Strands	true
Check if Reads are Balanced	true
Average Mapping Quality	50.0
Phred Quality	1.0
Remove Multiple Alleles	true

References

- Danecek P et al. (2021). Twelve years of SAMtools and BCFtools. *GigaScience, 10*(2).
- OmicsBox - Bioinformatics made easy. BioBam Bioinformatics (Version 3.0.23). March 3, 2019. www.biobam.com/omicsbox.

Figure 3. *Summary Report of Variant Filtering*

Figure 4. *Phred Quality Distribution*

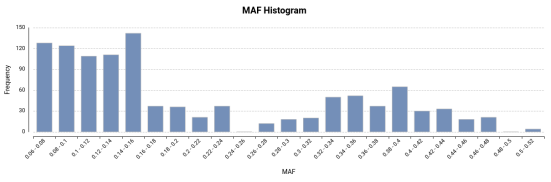


Figure 5. *Minimum Allele Frequency Distribution*

dDocent Variant Filtering Pipeline

INTRODUCTION

dDocent has developed a Freebayes-specific VCF filtering pipeline that takes into account quality, depth, missingness, balance between reference and alternatives alleles, and even populations.

This pipeline has proven to filter false positives remarkably well.

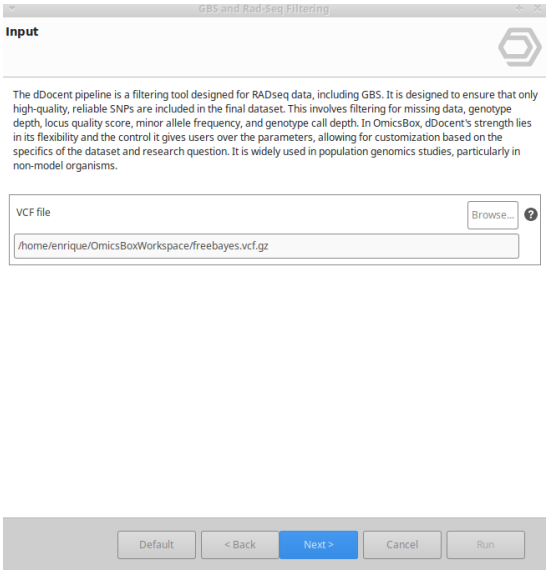
RUN DDOCENT VARIANT FILTERING PIPELINE ON OMICSBOX

The dDocent pipeline can be found under **Genetic Variation → Variant Filtering → dDocent Variant Filtering Pipeline.**The wizard consists of 4 pages and allows to define the input and output options as well as the analysis parameters (Figure 1, Figure 2, Figure 3 and Figure 4).

Input

The only necessary file that this pipeline needs is the VCF file.

This pipeline can only work for VCF files that were created with Freebayes using RAD-seq or GBS protocols.



CONFIGURATION 1

In this page you will set the parameters for the first four filters:

- **First Filter:** in this first step, common filters such as Allele Count thresholds are applied:
- Max. Missingness:threshold to filter out variants that have less than this fraction of genotypes called across all individuals.
- Minor Allele Count:filter out variants whose alternative genotype is found in fewer samples than this threshold.
- Minimum Quality:filter out variants with a quality score lower than this value.
- **Genotypic Depth Filter:** in this step you can filter out variants according to the number of supporting reads.
- Minimum Depth:filter out variants with a raw read depth lower than this value.
- **Sample Filter:** with this filter you can filter out samples that have a lot of missing information.
- Fraction of Individual Missingness:filter out individuals with less than this percentage of variants sampled.
- **Variant Missingness Filter:** this filter will remove variants with little information.
- Max. Missingness 2nd:second round of missingness threshold in the dDocent pipeline.
- Minimum Allele Frequency Threshold:the Minimum Allele Frequency (MAF) is the fraction of the least frequent allele in a population for a variant. Variants with a MAF smaller than this threshold will be filtered out.
- Min. Mean Depth:threshold for the average depth for a variant in all samples.

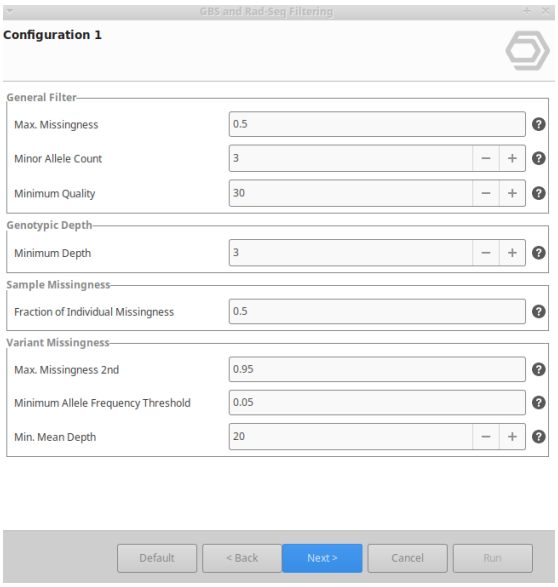
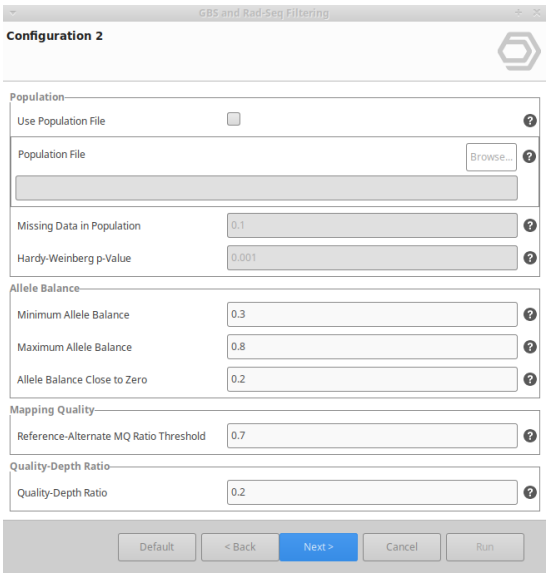


Figure 2. *Configuration 1 Page*

CONFIGURATION 2

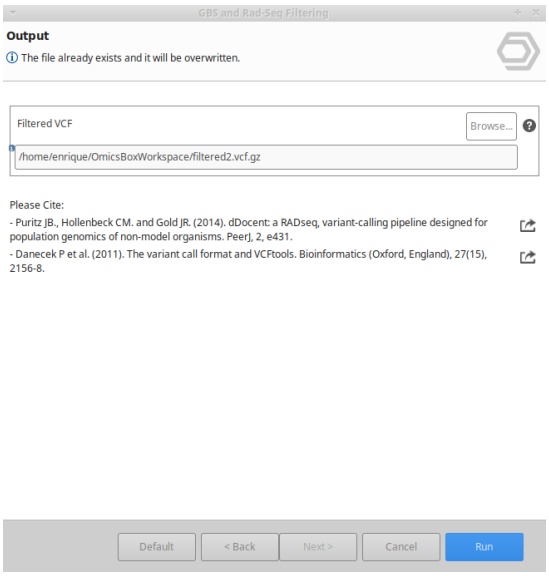
In this page you will be able to configure

- **Population Filter:** this filter consists of the removal of variants regarding different population metrics:
- **Use Population File:** check this parameter if you want to add a Population File to filter your variants using the Hardy-Weinberg Equilibrium and missingness inside population.
- **Population File:** tab-delimited text file with sample names in the first column and group names in the second column.
- **Missing Data in Population:** maximum fraction in the population that can have missing data in a variant before it is filtered out.
- **Hardy-Weinberg p-Value:** minimum cutoff for Hardy Weinberg p-value. Errors tend to have a very low p-value.
- **Allele Balance Filter:** this filter is used to remove variants that are biased towards some allele in case they are heterozygous.
- **Minimum Allele Balance:** minimum allele balance acceptable before filtering a site. Allele balance is calculated for heterozygotes as the number of bases supporting the least-represented allele over the total number of base observations.
- **Maximum Allele Balance:** maximum allele balance acceptable before filtering a site.
- **Allele Balance Close to Zero:** Allele Balance considered to be close to zero. This is necessary to catch loci that are fixed variants (all individuals are homozygous for one of the two variants).
- **Mapping Quality Filter:** this filtering step will remove all the variants whose reads have qualities biased towards the reference or the alternate allele.
- **Reference-Alternate MQ Ratio Threshold:** set the threshold ratio between the reference mapping quality and the alternate mapping quality. Variants with an absolute ratio lower than this value will be filtered out, as the mapping quality should be the same for both reference and alternative nucleotides.
- **Quality - Depth Filtering:** final filtering step to remove variants whose quality is not high enough for the depth they have.
- Quality-Depth Ratio: threshold for the quality-depth threshold for a variant.



OUTPUT

In this page you will only have to add where you want to save the filtered VCF file.



RESULTS

The main result is the filtered VCF file. Nevertheless, other outputs will be displayed in order to help you to interpret the results:

- **Report:** this report will summarize the main features of your VCF file and the filtering step (percentage of filtered variants, number of homozygous and heterozygous sites and proportion of missing data), just as the summary report from the general variant filtering. Nevertheless, there is an estimation of the number of erroneous genotypes that might be still in your dataset based on probabilities according to the genotype depth (see figure 5).

Variant Filtering Report

Input Data

`freebayes.vcf.gz`

Results

Number of variants before filtering	Number of variants after filtering	Percentage filtered
8095590	31918	99.6057%

Type of variant	Frequency
SNP	28139
MIXED	9
MNP	3034
INDEL	736

Number of alleles in a variant	Frequency
2	31779
3	138
4	1

Statistics After Filtering	Count (% total)
Number of Variants	31918
Number of Genotypes	255344
Number of Heterozygotes	155479 (60.89%)
Missing Data	0 (0.00%)

Figure 5. *Summary Report*

- **Charts:** different charts will show the distribution of values in different quality fields (see figure 6):
- MAF Histogram: fraction of the least frequent allele in a population for a variant.
- Proportion Quality / Counts: Phred-scaled probability that the site has no variant divided by the number of reads that support that site.
- Mapping Quality in Alternate Alleles: average of all mapping qualities of the reads supporting the variant.
- Raw Read Depth: total read depth at the locus, that is to say, the number of times that site was read, but not necessarily that variant (also the reference nucleotide, or other variants).

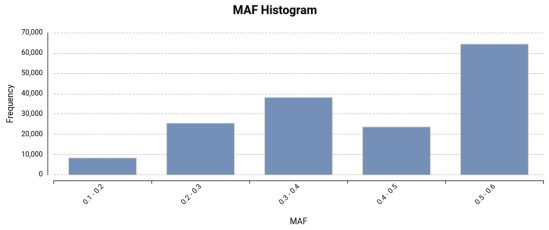


Figure 6. *Distribution of Minimum Allele Frequency*

4.3.5 Phasing and Imputation

Introduction

Phasing and imputation are two critical processes in the field of genetic variation. **Phasing** refers to the process of separating maternally and paternally inherited copies of each chromosome into haplotypes. The goal of phasing is to get a complete and accurate representation of each copy of the genome or region of interest. **Imputation**, on the other hand, is the statistical inference of unobserved genotypes. It is achieved by using known haplotypes in a population.

The importance of phasing and imputation in genetic variation field is significant, especially in **genome-wide association analysis pipelines**. Phasing provides a complete picture of genetic variation, which is crucial for understanding the breadth of biological variation within a species. Imputation, meanwhile, allows researchers to infer the identity of a missing marker based on the surrounding variants.

Run BEAGLE to phase and impute your VCF file.

BEAGLE can be found in the Genetic Variation Module of OmicsBox**.**The wizard consists of 3 pages and allows to define the input and output options as well as the analysis parameters (Figure 1, Figure 2).

INPUT

In the first page you will be able to select the VCF input file. The VCF file must contain variants from multiple samples and all samples must have the same ploidy (i.e., it is not possible to have a VCF file with mixed ploidy).

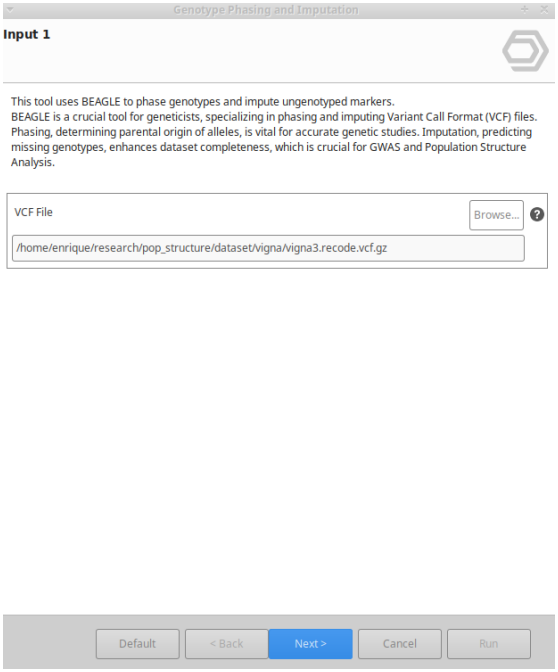


Figure 1. *Input Page*

CONFIGURATION

In this page you will be able to select parameters related to phasing and imputation, and other general parameters.

- **Phasing Parameters:**
- Max. Burn-in Iterations: the 'Burn-in' term describes the practice of throwing away some iterations at the beginning of a Markov chain Monte Carlo (MCMC) run. this parameter set the maximum number of burn-in iterations used to estimate an initial haplotype frequency model for inferring genotype phase.
- Phasing Iterations: number of iterations to estimate a genotype phase. The greater the value, the longer the computation time and the higher the accuracy.
- Model States for Phasing: number of models used to estimate the phasing of a genotype.
- **Imputation Parameters:**
- Model States for Imputation: number of models used to estimate a genotype.
- Imputation Segment: minimum length of haplotypes in centiMorgan (cM) that will be incorporated in the Hidden-Markov Model (HMM) for a target haplotype.
- Imputation Step: length in cM of the step used for detecting short Identity-By-State (IBS) segments.
- Number of Imputation Steps: number of steps to find IBS segments.
- Cluster Size: specifies the maximum cM distance between individual markers that are combined into an aggregate marker when imputing ungenotyped markers.

Identity by state is a method to measure similarity between unrelated individuals. It just considers the similarity between genotypes at each locus and averages over all the loci of interest. Two haplotype segments are identical by state if they are the same but they do not come from a common ancestor.

- **General Parameters:**
- Estimate Effective Population Size: if this parameter is checked, BEAGLE wil calculate the effective population size using an expectation maximization (EM) algorithm.
- Effective Population Size: specifies the effective population size. Beagle will automatically estimate the effective population size prior to phasing unless the previous parameter is unchecked.
- Sliding Window: specifies the cM length of each sliding window.
- Overlap Between Adjacent Windows: specifies the cM length of overlap between adjacent sliding windows.

Reducing the value of the Sliding Window parameter will reduce the amount of memory required for the analysis. Nevertheless, a very small window might mean that some slinding windows do not have a lot of variants to estimate genotypes, which will lead in an error.

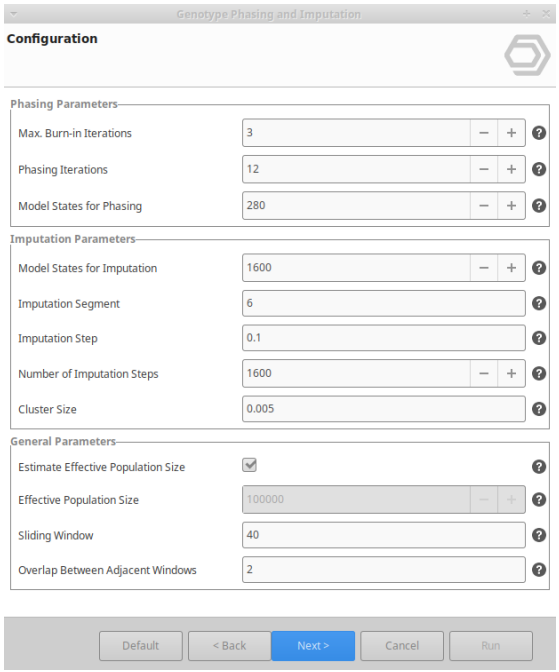


Figure 2. *Configuration Page*

Output

The main output of BEAGLE will be your phased and imputed VCF file.

After running BEAGLE on your VCF file, the new VCF file will lose the INFO and GT fields; the new VCF file will only have information about the genotypes. Because of that, if you want to filter variants do it before the use of BEAGLE.

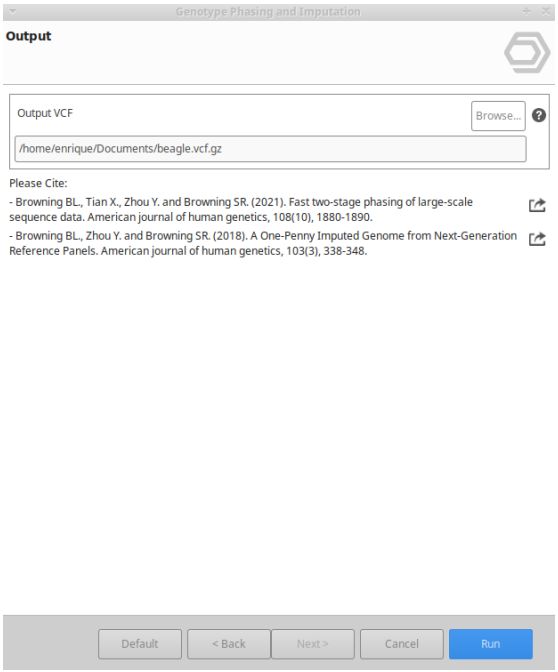


Figure 3. *Output Page*

Report

In addition to the phased and imputed VCF file, a summary report will be done with input information, parameters that were used in case you want to repeat the analysis with other parameters, and references (see Figure 4).

If there are too few variants in a scaffold or chromosomes, Beagle will throw and error. In OmicsBox, instead of an error, a warning will be thrown to let the user know that the scaffold or chromosome has been removed, and the report will show the name of the scaffolds or chromosomes removed.

Figure 4. *Beagle Summary Report*

4.3.6 Variant Annotation

Introduction

Variant annotation is the process by which variants are assigned functional information (for example, coding and genetic consequences of a variant) and it is a crucial process in genomic sequence analysis. The outcomes of such annotation are beneficial because they can directly influence the conclusions arrived at in disease studies.

The Variant Annotation tool in OmicsBox uses Ensembl Variant Effect Predictor (VEP) to get information of the variants present in the VCF introduced. This tool in OmicsBox does not only determine the effect on genes, transcripts and/or protein sequences using VEP, but also the transition/transversion ratio and other population genetics variables. To use the implementation, you just need to introduce your VCF file and a genome and annotation file in GTF format.

Please cite VEP using:
McLaren, W., Gil, L., Hunt, S. E., Riat, H. S., Ritchie, G. R., Thormann, A., ... & Cunningham, F. (2016). The ensembl variant effect predictor. Genome biology, 17(1), 1-14.

Run VEP for Variant Annotation

This tool can be found under **Genetic Variation → Variant Annotation**.The wizard consists of just one page to introduce some the inputs.

INPUT

- **VCF File:** VCF file created in the Variant Calling step. This VCF might be filtered in the Variant Filtering step.
- **Genome File:** FASTA file with the reference genome.
- **Annotation File:** both GTF and GFF formats are accepted.

Make sure that the reference genome and the reference annotation have the same version. In addition, take into account that the genome file introduced here as input must be the same one used in the Variant Calling step.

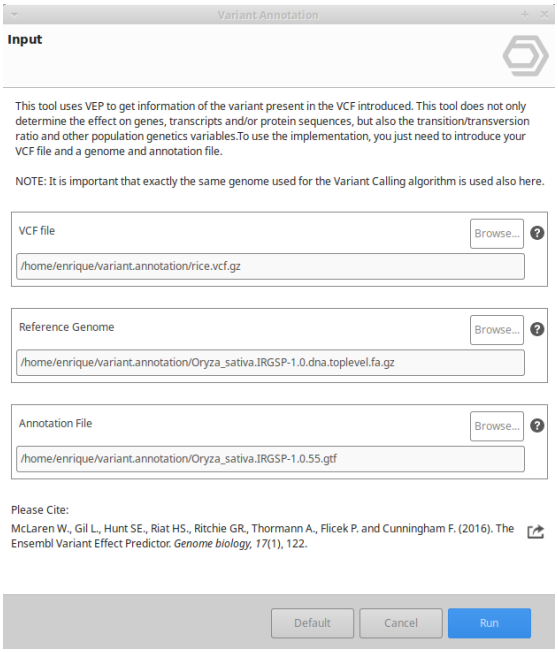


Figure 1. *Input Page*

Results

The Variant Annotation tool has the following outputs:

- **Table** listing the information of all variants that have been annotated.
- **Summary Report** with different type of information: general information, coding and genic consequences of the variants.
- **Type of Variation Distribution** in a pie chart in order to have a quick outline of the annotated variants.
- **Quality-control Charts**. This charts can be displayed using the sidebar buttons and they help ensure that the variant dataset does not have any kind of anomaly (a chromosome with a significantly higher number of variants, very long indels, etcetera).

TABLE

- **Table** with information of all variants. This table has the next columns:
- **Type of Variation:** according to ENSEMBL, this column can be:
  - SNP: Single Nucleotide Polymorphism. A change of one nucleotide.
  - Substitution: a sequence alteration where the length of the change in the variant is the same as that of the reference.
  - Insertion: addition of one or several nucleotides.
  - Deletion: removal of one or several nucleotides.
  - Indel: an insertion and a deletion, affecting 2 or more nucleotides.
  - Other: structural variation, etcetera.
  - **Chromosome/Scaffold:** chromosome where the variant is located according to the VCF file.
  - **Location:** 1-based position inside the chromosome.
  - **Reference:** nucleotide or sequence that appear in the reference genome.
  - **Variation:** nucleotide or sequence appearing in the VCF as variant.
  - **Gene(s):** affected genes by that variant If more than one gene is affected, several genes will appear separated by semicolon.
  - **Pi:** measures nucleotide divergence among all samples in that position. It is calculated as the average proportion of nucleotide differences between all pairs of sequences within a population. A higher value of Pi indicates a higher level of genetic diversity within a population.
  - **HWE p-value:** reports a p-value for each site from a Hardy-Weinberg Equilibrium test. The Hardy-Weinberg equilibrium is a theoretical state in which the frequency of alleles and genotypes in a population remains constant from generation to generation in the absence of any genetic or environmental influences that can affect the distribution of alleles.

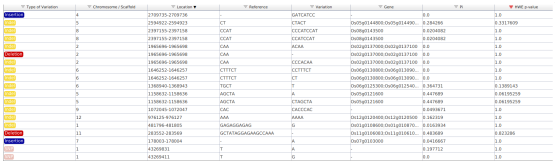


Figure 2. *Variant Annotation Main Table*

SUMMARY REPORT

The summary report has three main parts:

- **A general outline** with a summary of the variant annotation job and the variant classes that were found in the VCF file. The job summary shows information about the process of annotation itself:
- Number of different variants that the VCF file contains.
- Fraction of novel variants against the ones registered under some ID.
- Number of overlapping genetic attributes (genes, regulatory features and transcripts).
- Variants that were not annotated (filtered out).
- Number of variant sites that were actually processed.

Although the first and the last entry of this table seem to be analogues, they do not. The first entry takes into account the number of lines in the VCF file without the header, whereas the last entry focuses on variants that could be mapped inside the annotation file.

- **Variant classes.** Refers to the type of variants used in the variant annotation process. This numbers must be equal to the ones that appear in the distribution pie chart.

The total number in this table might be higher than the number of 'Variants processed' in the Job Summary table as some sites might have multiallelic variants, that is to say, sites with more than one variation.

Results

General Information

Statistic	Count
Lines of input read	200000
Novel / existing variants	0
Overlapped genes	37352
Overlapped regulatory features	0
Overlapped transcripts	43870
Variants filtered out	0
Variants processed	199331

Variant Classes

Type of variant	Count (% total)
SNP	264985 (85.46%)
Substitution	0 (0.00%)
Insertion	5566 (1.80%)
Deletion	6401 (2.06%)
Indel	33111 (10.68%)
Other	0 (0.00%)
Total	310063 (100.00%)

Figure 3. *General Overview of the Variant Annotation Job*

- **An overview of the effects** of the variants both at the genetic and the coding level. In the case of the genetic consequences, the percentage of severe consequences is also disclosed. The most common severe consequences are:
- Stop Gained: a sequence variant whereby at least one base of a codon is changed, resulting in a premature stop codon, leading to a shortened transcript.
- Stop Lost: a sequence variant where at least one base of the terminator codon (stop) is changed, resulting in an elongated transcript.
- Frameshift: variant that causes a disruption of the translational reading frame, because the number of nucleotides inserted or deleted is not a multiple of three.
- Splice Donor or Splice Acceptor: it changes the outcome of the splicing event.
- Initiator Codon: variant that misplaces an initiator codon.
- Stop Retained Variant: a sequence variant where at least one base in the terminator codon is changed, but the terminator remains.
- Missense Variant: variant that changes one amino acid in the protein but the length is preserved.
- Inframe Insertion: an inframe non-synonymous variant that inserts bases into in the coding sequence.
- Inframe Deletion: An inframe non-synonymous variant that deletes bases from the coding sequence.

Effect of the Variants

Genetic Consequences

Name	Count (% severe)
3 prime UTR variant	8438 (51.40%)
5 prime UTR variant	5814 (53.20%)
Coding sequence variant	26 (46.15%)
Downstream gene variant	156978 (15.79%)
Frameshift variant	927 (74.97%)
Inframe deletion	569 (60.28%)
Inframe insertion	588 (64.12%)
Intergenic variant	126292 (61.51%)
Intron variant	33761 (47.49%)
Missense variant	7645 (55.04%)
Non coding transcript exon variant	1011 (45.00%)
Non coding transcript variant	875 (0.00%)
Protein altering variant	271 (56.46%)
Splice acceptor variant	64 (62.50%)
Splice donor 5th base variant	90 (42.22%)
Splice donor region variant	153 (45.75%)
Splice donor variant	78 (51.28%)
Splice polypyrimidine tract variant	1032 (52.03%)
Splice region variant	669 (15.70%)
Start lost	54 (62.96%)
Start retained variant	6 (16.67%)
Stop gained	327 (88.07%)
Stop lost	26 (61.54%)
Stop retained variant	8 (12.50%)
Synonymous variant	4829 (45.27%)
Upstream gene variant	164629 (38.76%)
Total	515160 (38.00%)

Genetic Consequences refer to the outcome of the variant at DNA and RNA level.

Because of that, all variants have consequences at that level and they can have more than one consequence.

Figure 4. *Genetic Consequences of the Variants*

Coding Consequences

Name	Count (% total)
Coding sequence variant	26 (0.17%)
Frameshift variant	927 (6.07%)
Inframe deletion	569 (3.72%)
Inframe insertion	588 (3.85%)
Missense variant	7645 (50.05%)
Protein altering variant	271 (1.77%)
Start lost	54 (0.35%)
Start retained variant	6 (0.04%)
Stop gained	327 (2.14%)
Stop lost	26 (0.17%)
Stop retained variant	8 (0.05%)
Synonymous variant	4829 (31.61%)
Total	15276 (100.00%)

Coding Consequences refer to the effect of the variant at protein level.

Thus, only variants that fall inside conding regions will have a consequence.

Figure 5. *Coding Consequences of the Variants*

- **Population genetics statistics** such as the transition/transversion ratio (Ts/Tv ratio) and a table with the inbreeding coefficient.
- The Ts/Tv ratio can be used to know if the analysed samples form a normal population if you know the Ts/Tv ratio for a normal population of the species that is being studied.
- The inbreeding coefficient (F) of a sample is the probability that two alleles at any locus in that individual are identical by descent from the common ancestor(s) of the two parents. F stands for fixation index, because of the increase in homozygosity, or fixation, that results from inbreeding. If this coefficient is negative, the number of actual homozygotic sites is smaller than the number of expected homozygotic sites. If it is positive, there are more homozygotic sites than expected.

Population genetics

Ts/Tv ratio

Change	Fraction
AC	0.081
AG	0.344
AT	0.104
CG	0.048
CT	0.343
GT	0.080
Overall Ts/Tv ratio	2.196

Heterozygosis in samples

Sample	Observed Homozygotes	Expected Homozygotes	Number of Sites	Inbreeding Coefficient (F)
Z274	32047	32080.8	38061	-0.00564
Z30	44048	44999.4	53459	-0.11246
Z407	23469	23315.9	27697	0.03495
Z1002	26943	26810.1	31823	0.02651
Z44	45806	47983.7	57140	-0.23784
Z1145	22856	22448.2	26672	0.09655
Z206	34283	34330.1	40811	-0.00727
Z857	34410	34535.9	40971	-0.01956
Z366	36005	36134.1	42892	-0.01911
Z440	22650	22273.1	26435	0.09056
Z637	26726	26556.9	31501	0.0342
Z1488	22599	22157.3	26259	0.10768
Z282	28995	28880.6	34293	0.02113
Z770	35865	36464.4	43350	-0.08706
Z591	29533	29758.0	35360	-0.04017
Z321	26362	26348.8	31278	0.00268
Z1402	45499	47116.6	56106	-0.17995
Z1020	44006	44770.3	53156	-0.09115
Z1183	37259	37151.1	44146	0.01542
Z478	20087	20092.6	23926	-0.00146

***Figure 6**. Population Genetic Results*

TYPE OF VARIATION DISTRIBUTION

In this pie chart you can see all the variants that have been annotated. Take into account that in the same position, for example, a SNP and an indel can be found, so both variants will be taken into account for the pie chart.

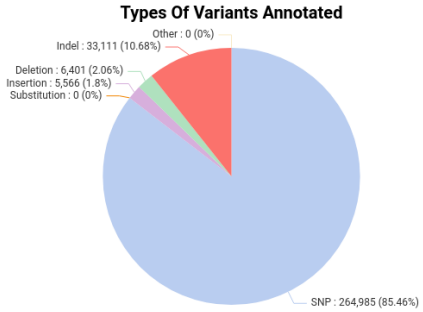


***Figure 7.** Pie Chart with the Types of Variants*

QUALITY-CONTROL CHARTS

There are three differents quality-control charts:

- **Distribution of Indel Lengths:** it might follow a normal distribution.

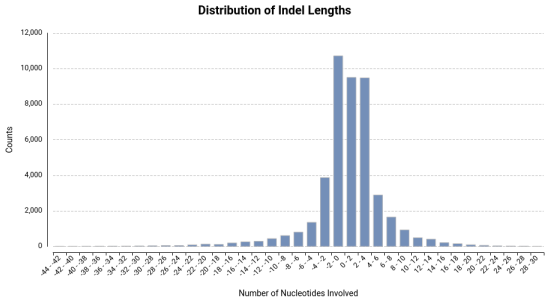


***Figure 8**. Distribution of Indel Lengths*

- **Variants per Chromosome:** this histogram should be even for all chromosomes. That means that all chromosomes have approximately the same number of variants.

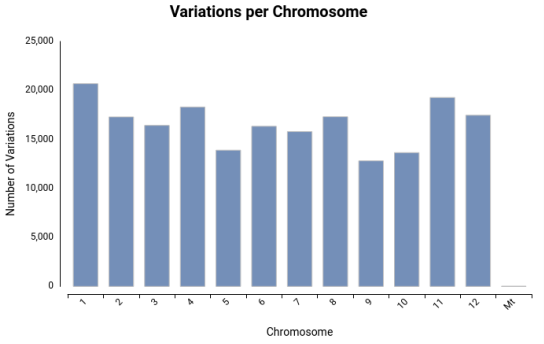


***Figure 9**. Distribution of Variants per Chromosome*

- **Position in Protein:** this histogram should also be even, as that will mean that variants that are in coding regions distribute equally.

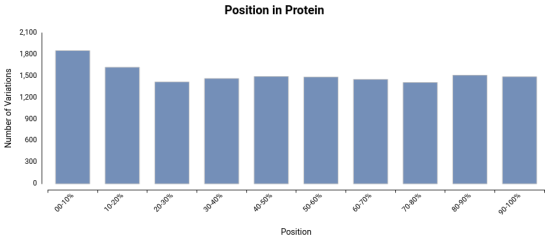


***Figure 10.** Distribution of Coding Variants in Proteins*

DETAILED INFORMATION

If you want more information about a variant, you can click with the right button the line of the variant and click in "Show Annotation Details". Then, an report will open with one table per gene affected by that variant. Each table will have one row per gene feature affected by that variant. That table will have the following columns:

- **Feature ID:** name (or ID) of the gene feature affected by that variant.
- **Feature Type:** class of gene feature. It can be: transcript, regulatory feature or motif feature.
- **Consequence:** effect of that variant. This consequence will be one among the ones that appeared in the report.
- **cDNA Position:** relative position of base pair in cDNA sequence. It will only appear if that variant falls inside a region that can be transcribed.
- **CDS Position:** relative position in coding sequence. It will only appear if that variant is inside a region that is translated.
- **Protein Position:** the same as before but in the protein.
- **Amino Acids:** amino acid change. Only given if the variant affects the protein-coding sequence.
- **Codons:** the alternative codons with the variant base in upper case.
- **Impact:** the impact modifier for the consequence type.
- **Distance:** shortest distance to transcript.
- **Strand:** 1 for positive strand and -1 for negative strand.

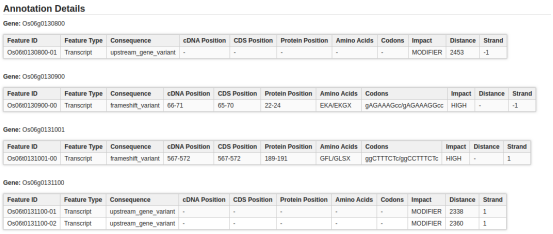


***Figure 11**. Variant Consequences Information*

4.3.7 Population Structure Analysis

Introduction

Population Structure can be defined as the presence of a systematic difference in allele frequencies between different groups of individuals of the same species. Population Structure is important in numerous application areas, including evolution, sample selection in agriculture or conservation. Population Structure may arise for a variety of reasons, but a common cause is that individuals have been drawn from geographically isolated groups or different locales across a geographic continuum.

Understanding the structure in a group of samples is necessary before more sophisticated analyses are undertaken, such as Genome-Wide Association Studies (GWAS) in order to know if population structure might be a confounding factor in association analysis. Moreover, Population Structure Analysis can be important to infer divergence times between two populations.

Please cite ADMIXTURE as:

D.H. Alexander, J. Novembre, and K. Lange. Fast model-based estimation of ancestry in unrelated individuals. Genome Research, 19:1655–1664, 2009.

Run ADMIXTURE to analyze population structure.

Population Structure can be found in the Genetic Variation Module of OmicsBox.The wizard consists of 3 pages and allows to define the input and output options as well as the analysis parameters (Figure 1, Figure 2).

INPUT

In the first page you will be able to select the input files.

- VCF File: select a VCF File with all the samples whose population structure needs to be analyzed.
- Use Supervised: select this option if you know the ancestry or demographic group to which certain samples belong.
- Population File: this file must be a tab-file with two columns: the first one must be sample name (identical to the sample name in the VCF file. The second column must be the ancestry they belong to. This input file can only be uploaded if the "Use Supervised" option is enabled.

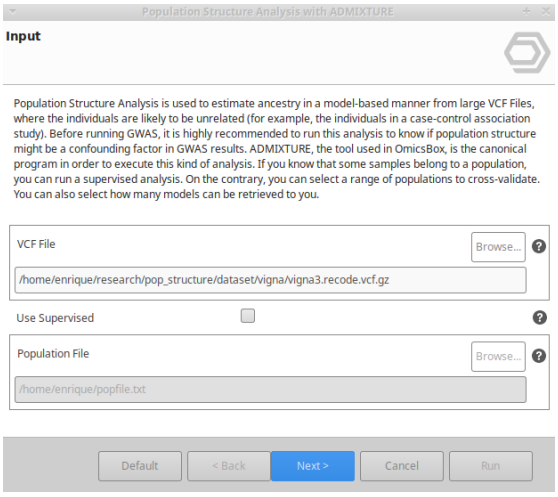


Figure 1. *Input Page*

CONFIGURATION

In this page the parameters for Linkage Disequilibrium Pruning (LD Pruning) and ADMIXTURE. LD Pruning is a common step before running Population Structure Analysis, as ADMIXTURE does not take LD into consideration, and LD pruning might lead into a smaller dataset but just as informative with redundant variants being removed.

- **Linkage Disequilibrium Pruning:**
- Maximum Linkage Disequilibrium: threshold of the degree of correlation (R2) between two loci.
- Window Size: number of nucleotides to look for Linkage Disequilibrium.
- **Admixture Parameters:**
- Use Haploid Mode: check this box if your data is haploid (i.e. the species you are working with only has one copy of the genome, for example bacteria).
- Minimum number of populations to fit: ADMIXTURE will find the optimum number of subpopulations present in your dataset. The program will look in a range from the number set in this parameter to the number set in the following parameter.
- Maximum number of populations to fit: maximum number of populations to analyze.
- Cross Validation: cross validation folds.

In the supervised mode, if you have a population file with samples that belong to *n* different populations. ADMIXTURE will try to classify the other samples inside those populations or in an extra one, being the total number of populations *n+1*.

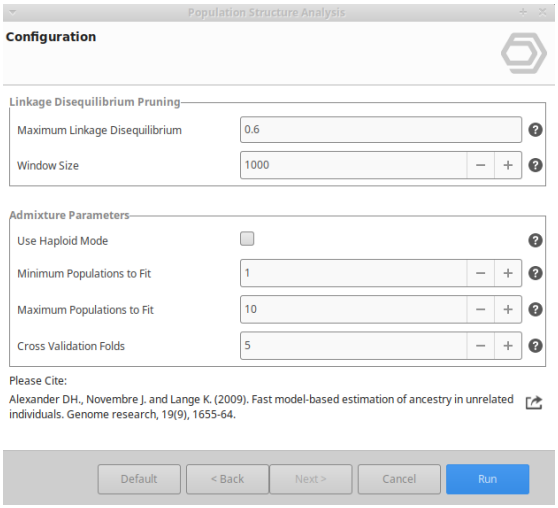


Figure 2. *Configuration Page*

Results

Population Structure on OmicsBox will have two main outputs:

- **Main table:** it will have as many rows as the models that were tested using ADMIXTURE (Figure 3). It will have two columns:
- Number of populations tested.
- Cross Validation error of the model.
- **Summary Report:** in this report you will see the name of the VCF file used and the parameters set to obtain those results.
- **Summary Chart:** this chart will display the same information as the table: the cross validation error for each of the models tested (Figure 4).

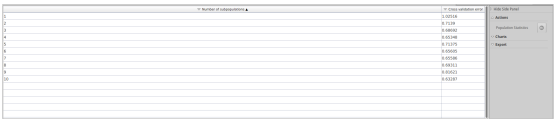


Figure 3. *Summary Table*

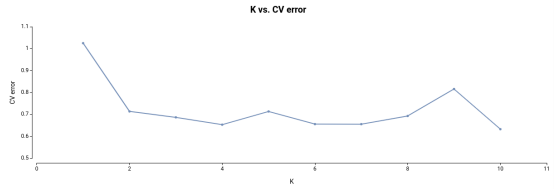


Figure 4. *CV Error vs K Chart*

Extract information from models

In order to extract information of a model, you can use the sidepanel of the table (see Figure 3). In this sidebar you will be able to select different types of information: a summary report with values of a variety of population statistics, a set of charts, and you can even export a pair of files that might be useful a information in case you are interested in some (or even all) models, you will be able to extract more information from each one.

POPULATION STATISTICS

In order to obtain a report with different population statistics per subpopulation, click on **Actions → Population Statistics** in the Sidebar. In this new wizard, please select the model that you want to study. A new report will appear (see Figure 5).

Information about the following Population Statistics will appear:

- **Tajima's D:** this test help distinguish a population following the Hardy-Weinberg equilibium (value close to 0) from a changing population (a positive value mean that there is a balancing selection and a negative value that the population is expanding).
- **Pi (π):** this variable is also known as nucleotide diversity and it is a measure of genetic variation. This statistic may be used to monitor diversity within or between populations, to examine the genetic variation in crops and related species, or to determine evolutionary relationships. A higher value of Pi indicates a higher level of genetic diversity within a population.
- **Heterozygosity:** fraction of heterozygous variants in the total of variants.

Population Statistics Report

Population	Tajima D	Pi	Heterozygosity
Pop1	0.99	0.19	0.91
Pop2	0.40	0.08	0.17
Pop3	0.94	0.19	0.78
Pop4	0.77	0.10	0.52
Pop5	0.69	0.12	0.46
Pop6	0.57	0.07	0.03

References

- Alexander DH., Novembre J. and Lange K. (2009). Fast model-based estimation of ancestry in unrelated individuals. *Genome research, 19*(9), 1655-64.
- Danecek P et al. (2011). The variant call format and VCFtools. *Bioinformatics (Oxford, England), 27*(15), 2156-8.
- OmicsBox - Bioinformatics made easy. BioBam Bioinformatics (Version 3.1.100). March 3, 2019. www.biobam.com/omicsbox.

Figure 5. *Population Statistics Report for K = 6*

POPULATION CHARTS

To obtain population charts click on **Charts → Population Charts**. A new wizard like the one in Figure 6 will be displayed.

- **Populations:** choose the model with the desired number of populations.
- **Stacked Barchart.** Check this option if you want to obtain a stacked barchart with the proportion of genetic ancestry components for each individual or population. Samples are colored according to the population or ancestry they belong to (Figure7).
- **Principal Component Analysis.** Check this option if you want to obtain a PCA of the genetic composition of all samples. Each sample is colored according the population it belongs to (Figure 8).
- **Heatmap.** Check this option if you want a heatmap with the genetic distance (e.g. resemblance) between different populations (Figure 9).

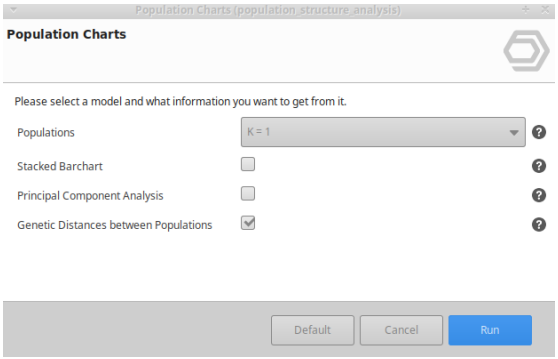


Figure 6. *Population Charts Wizard*

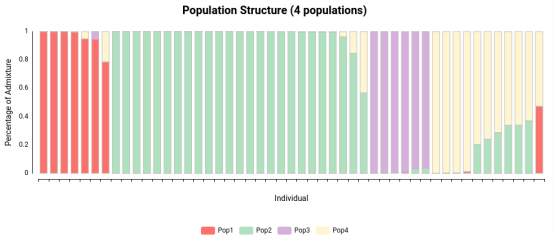


Figure 7. *Stacked Barchart*

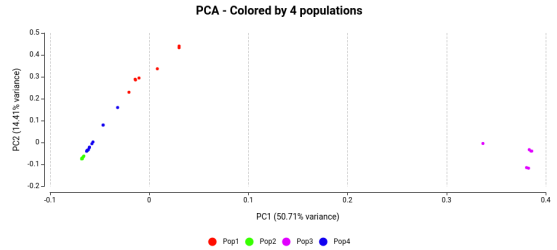


Figure 8. *Colored PCA*

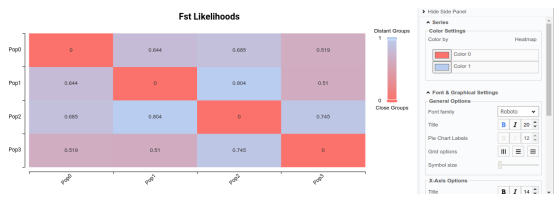


Figure 9. *Fst Likelihoods Heatmap*

EXPORT POPULATION INFORMATION

To obtain files with further information about the models, click on **Export → Population Information.** A new wizard will appear (see Figure 10) You will be able to obtain the following information:

- **Allele Frequency File.** File with the population allele frequencies for each SNP. The first two columns are the chromosome and the position where SNPs are found. The rest of the columns are the allele frequencies in different populations.
- **Population File.** File with the sample names and the population (or ancestry) they belong to.

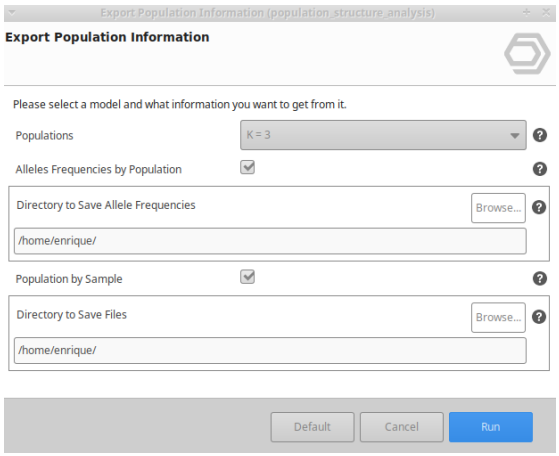


Figure 10. *Extract Population Information Wizard*

4.3.8 Genome Wide Association Study

Genome Wide Association Study

INTRODUCTION

Genome Wide Association Studies (GWAS) test from thousands to millions of genetic variants across many genomes to find those statistically associated with a specific trait or disease. GWAS results have a range of applications, such as gaining insight into a phenotype's underlying biology, estimating its heritability, and calculating genetic correlations.

Please cite GAPIT3 as:

Wang, J., & Zhang, Z. (2021). GAPIT Version 3: boosting power and accuracy for genomic association and prediction. *Genomics, proteomics & bioinformatics*, *19*(4), 629-640.

RUN GAPIT3 FOR GWAS

The Genome-Wide Association Studies Tool can be found under **Genetic Variation → Genome Wide Association Study.**The wizard consists of 4 pages and allows to define the input and output options as well as the analysis parameters (Figure 1, Figure 2, and Figure 3).

Input

- First of all, FLAIR requires two types of necessary files:
- **VCF File:** this file must contain the SNPs that are going to be studied. It is originated from the Variant Calling step and might be filtered, although it is not necessary.
- **Phenotype Data:** tab-delimited file with the same sample names that in the VCF file in the first column and traits in several columns. Header is necessary.

Although you can use a Phenotype File with all the traits you want, the more traits you introduce in a single file, the more time it will take. In order to be more efficient, please separate your traits in different files and run several GWAS.

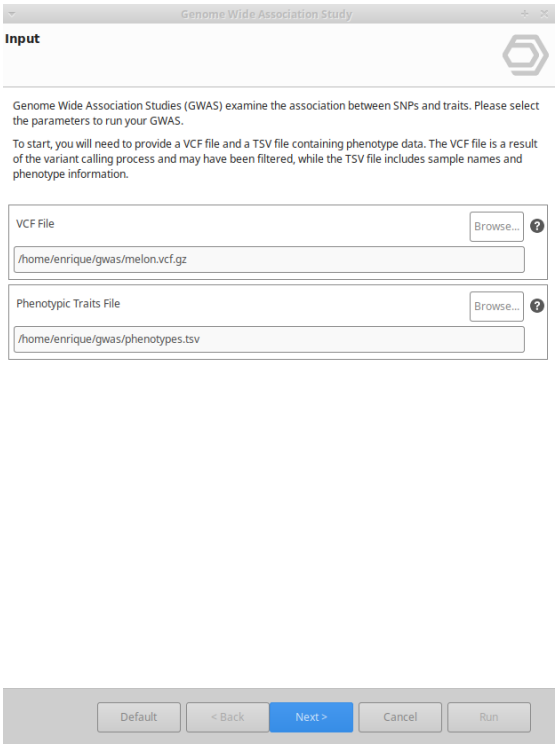


Figure 1. *Input Page*

Configuration 1

In this page, you can set parameters to filter your VCF file in terms of population genetics parameters (for instance, Hardy-Weinberg Equlibrium p-value or Minor Allele Frequency). In addition, you can set whether you want to normalize your phenotype data, as it is recommended that measures follow a normal distribution.P

- **Population Genetics Filter Parameters:**
- **Hardy-Weinberg Equilibrium P-value**: assesses sites for Hardy-Weinberg Equilibrium using an exact test, as defined by Wigginton, Cutler and Abecasis (2005). Sites with a p-value below the threshold defined by this option are considered to be out of HWE, and therefore excluded.
- **Minor Allele Frequency (MAF) Threshold**: minor allele frequency (MAF) is the frequency at which the second most common allele occurs in a given population. Include only sites with a Minor Allele Frequency greater than or equal to this value.
- **Missingness Threshold**: exclude **sites** on the basis of the proportion of missing data. If a variant is missing in a higher percentage of samples than the threshold set here, it is excluded.
- **Sample filtering:**
- **Sample Missingness Threshold**: exclude **samples** on the basis of the proportion of missing data. If a sample do not have the minimum percentage of variants set here, this sample is excluded.
- **Linkage Disequilibrium Pruning:**
- **Perform LD Pruning:** check this box in order to perform LD pruning. Linkage disequilibrium (LD) is a measure of correlation of genotypes between a pair of variants. LD-pruning is the process of filtering variants so that those that remain have LD measures below a given threshold. This procedure is typically used to identify independent subsets of variants. This is often the first step in evaluating relatedness and population structure to avoid having results driven by clusters of variants in high LD regions of the genome.
- **Maximum Linkage Disequilibrium:** variants with a correlation greater than this value will be removed.
- **Window Size:** window of variants to look for linked variants.
- **Phenotype Data Preprocessing:**
- **Remove Phenotype Outliers**: remove outliers in the phenotype data (e.g. outside 1.5 times the interquartile range above the upper quartile and below the lower quartile.
- **Normalize Phenotype Data**:check this option if you think your data does not follow a normal distribution. The normalization method is rank-based inverse normal transformation.

It is not recommended to remove outliers and normalize phenotype data at the same time.

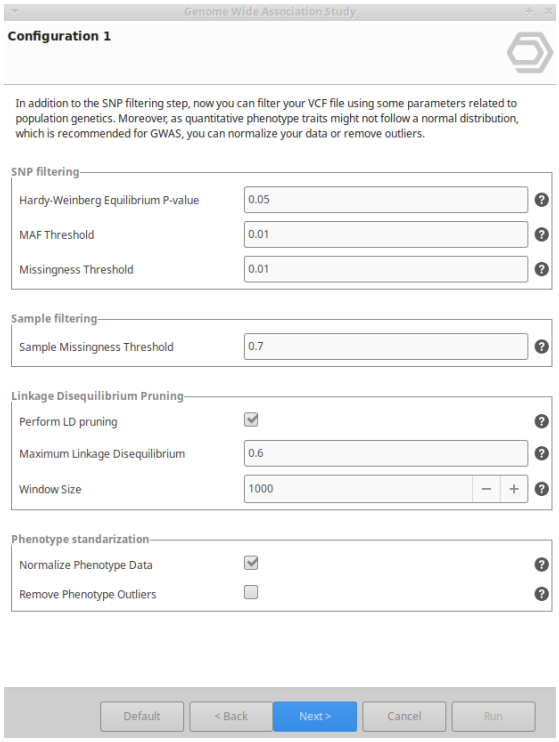


Figure 2. *Configuration 1 Page*

Configuration 2

In this page, different parameters in order to conduct the GWAS can be set:

- **Use Kinship Matrix:** check this parameter if you want to use your own Kinship Matrix. Otherwise, it will be calculated before running the GWAS inside OmicsBox. A kinship matrix is an all-vs-all comparison among samples used to measure the degree of relatedness between individuals. It is used in GWAS to control for the effects of ancestry or family relatedness on the trait studied.
- **Kinship Matrix:** file with the Kinship Matrix. The kinship matrix file must be formatted as an n by n+1 matrix where the first column contains sample names, and the rest is a square symmetric matrix. The first row of the kinship matrix file does not consist of headers.
- **Kinship Group:** set which measure you want to use to group your samples in order to make the kinship matrix.
- **Kinship Algorithm:** establish the algorithm to make the kinship matrix.
- **Kinship Cluster**: establish how to cluster your samples to perform the kinship analysis.
- **Use Covariate Matrix:** check this parameter if you want to use your own Covariate Matrix. Otherwise, it will be calculated before running the GWAS inside OmicsBox. Covariates are variables that are thought to be related to the data being analyzed, but are not of primary interest. Regarding to GWAS, a covariate matrix is a table of variables that are used to adjust for the effects of other variables on the data being analyzed. It can be obtained using Principal Component Analysis (PCA).
- **Covariate Matrix:** file with metadata information to generate covariate matrix. This file must be formatted similarly to the phenotypic files (header and sample names in the first column). If all metadata is quantitative, covariate matrix will be generated using PCA. If data is qualitative as a whole, MCA will be used, and if metadata has mixed types FAMD will be performed.
- **Number of Dimensions:** number of dimensions to make a PCA and get the covariate matrix.
- **Model:** establish the model to use in the GWAS analysis. This model represents how to analyze the relationship between a trait and genetic variation.

Take into account that all models present in this tool are linear models, so it is recommended to only associate variants to quantitative traits that follow a normal distribution, but not to qualitative ones, as this kind of data should need logistic regression models.

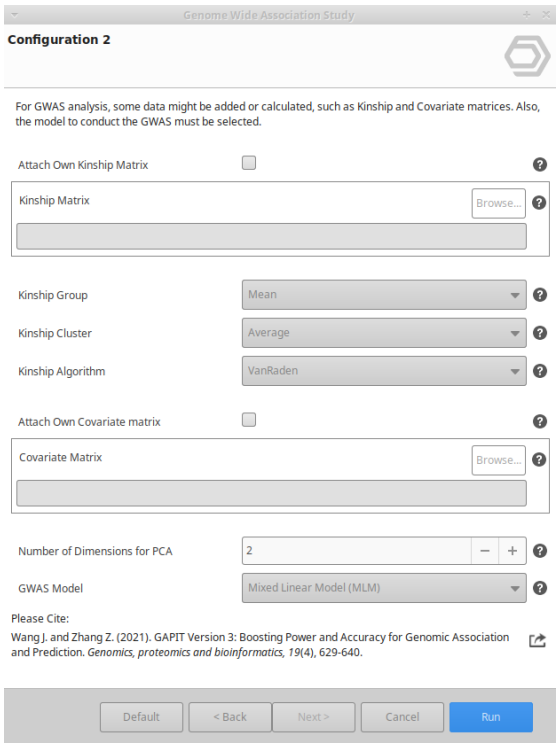


Figure 3. *Configuration 2 Page*

Linear Models available on GAPIT3

It is important to explain the different linear models that can be used in GAPIT3:

- **General Linear Model (GLM):**
- GLM is a basic linear regression model that tests the association between genetic variants and a trait. It assumes a **linear relationship** between genetic markers and the trait of interest.
- This model is a good starting point for GWAS when there is **little concern about population** structure or relatedness among individuals. It's less suitable when dealing with structured populations, as it may lead to false positives.
- **Mixed Linear Model (MLM):**
- MLM extends the GLM by incorporating a random effect term to account for population structure and relatedness among individuals. This **reduces the risk of false positives** by controlling for cryptic relatedness and population stratification.
- This is suitable for involving populations with complex structures, such as human populations with diverse ethnic backgrounds or **plant breeding** populations.
- **Compression MLM (CMLM):**
- CMLM is an enhancement of the MLM that uses a **compression step** to reduce computational complexity. It can be used when performing **large-scale GWAS** where computational efficiency is essential, but you still need to account for population structure and relatedness.
- **MLMM (Multi-Locus Mixed Model):**
- MLMM is an extension of MLM that explicitly models multiple associated loci simultaneously, allowing for **more accurate detection of complex genetic** architectures.
- MLMM is suitable for traits influenced by **multiple loci with small effects.**
- **FarmCPU:**
- FarmCPU stands for Fixed and random model Circulating Probability Unification and it aims to **control false positives** so it can be useful when dealing with data where there is a high risk of spurious associations due to confounding factors.
- **gBLUP (Genomic Best Linear Unbiased Prediction):**
- gBLUP is primarily used for predicting breeding values in animal or plant breeding. It estimates the genetic variance and covariance between individuals based on genomic information and pedigree data.
- It is valuable in breeding programs to estimate genetic merit and **make selections for breeding** based on genomic information.
- **BLINK (Bayesian Information and Genomic Selection):**
- BLINK stands for Bayesian-information and Linkage-disequilibrium Iteratively Nested Keyway and it is an enhanced version of FarmCPU.
- BLINK is useful when you want to perform both association testing and genomic prediction simultaneously, as in genomic selection in plant and animal breeding.
- **SUPER (Set-based Unified P-value Combination):**
- SUPER sums up p-values from multiple SNPs within predefined sets (e.g., gene-based sets) to increase statistical power to detect associations.
- It is useful when you have prior knowledge **suggesting that variants** within specific gene sets or pathways are collectively associated with the trait.

The choice of GWAS model depends on the specific research question, the characteristics of the study population, and the available computational resources. Researchers often perform multiple analyses using different models to robustly identify genetic associations.

Output

- **Destination Folder:** should you treat your phenotypic data with OmicsBox (see fig. 2) or provide metadata information, in this folder you will received your new phenotypic data or/and the covariate matrix.

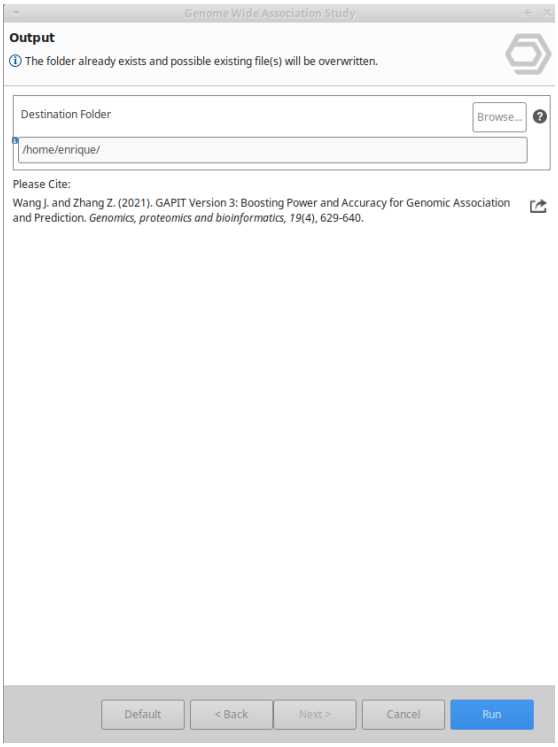


Figure 4. *Output Page*

RESULTS

Variant Calling has the following outputs:

- **Association Table** with information for each SNP and Phenotype.
- **Genotype Charts:** information related only to the population structure and not to any phenotype introduced.
- **Phenotype Charts:** data associated to phenotypes, so there will be one chart per phenotype that was used.
- **Summary Report**.

Association Table

This table has the following information for each SNP.

- Phenotype: phenotype associated to that SNP.
- SNP: ID of the SNP. If the VCF did not have any ID, this field will have a combination of the chromosome name and the position.
- Chromosome: chromosome where the variation is found.
- Position: 1-based position in the chromosome where the variation was found.
- Minimum Allele Frequency: frequency at which the second most common allele occurs in a given population.
- Number of Samples Used: this number can vary among the different phenotypes depending on the phenotypic information for each sample. In addition, some samples could have been filtered out during the sample filtering step.
- Effect: phenotypic variance attributable to that variant. If it is positive, the presence of the variant increment the power of the characteristic, whereas a negative value means that the presence of the variant diminish the quantitative value of that characteristic. A greater absolute value means a higher importance of the variant regarding to the phenotype.
- P-value: significance of the association.
- Adjusted P-Value: the Benjamini-Hochberg method is used.

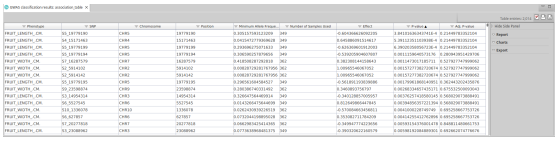


Figure 5. *Association Table*

Genotype Charts

These charts are related to the set of a variants that is possessed by a population and they are not related to any phenotype.

- **PCA:** this PCA is done with distances among samples taking into account only their genotypes, which appear in the VCF file. This PCA can be coloured by phenotypic values.
- **MAF Histogram:** distribution of the frequency at which the second most common allele in the whole population.
- **Marker Heterozigosity:** heterozigosity is the condition of having two different alleles at a locus. This histogram shows the proportion of sites that are heterozygotic. High level of heterozygosis indicated low quality.
- **Sample Heterozigosity:** this histogram shows the percentage of heterozygotic sites per sample. Again, high level of heterozygosis indicated low quality.

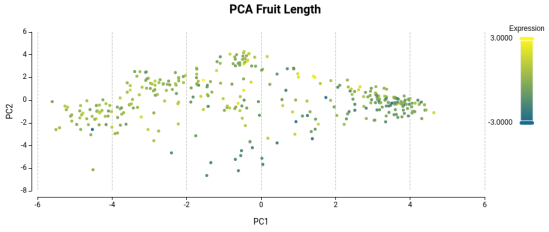


Figure 6. *PCA with Genotypic Information (color represents a normalized intensity)*

Phenotype Charts

These charts depend on the values of the phenotypic traits in the population, so there are as many of each type as the number of phenotypes included in the analysis. There are two types:

- **QQ-plot:** a qq-plot (quantile-quantile plot) shows the deviation of the observed P-values from the null hypothesis. That is to say, the X-axis represents the negative logarithm of expected p-values, and the Y-axis, the negative logarithm of the observed p-values. In a theoretical GWAS case where there are not causal polymorphisms, this plot will represent a diagonal line. Those variants that are significantly associated to the phenotype, will be represented plotted above the diagonal.

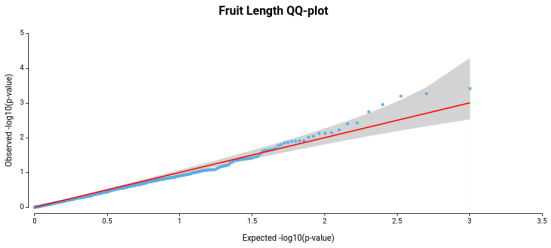


Figure 7. *QQ-plot of the Fruit Length Phenotype*

- **Manhattan Plot:** summary chart that represents every position with a variant in the genome in the X-axis, and its negative logarithm p-value in the Y-axis. If a SNP is related to a specific phenotype according to the chosen model, that variant will be above the red horizontal line, which represented the threshold to accept a significant adjusted p-value.

The threshold value used in the horizontal line is the critical p-value. That is to say, the smaller p-value whose BH-adjusted p-value is bigger than 0.05.

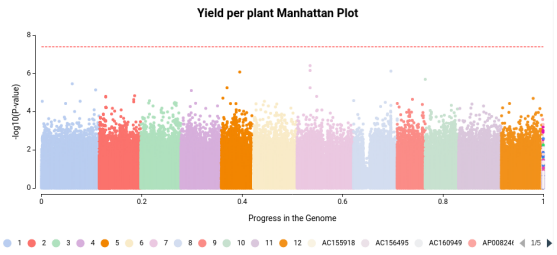
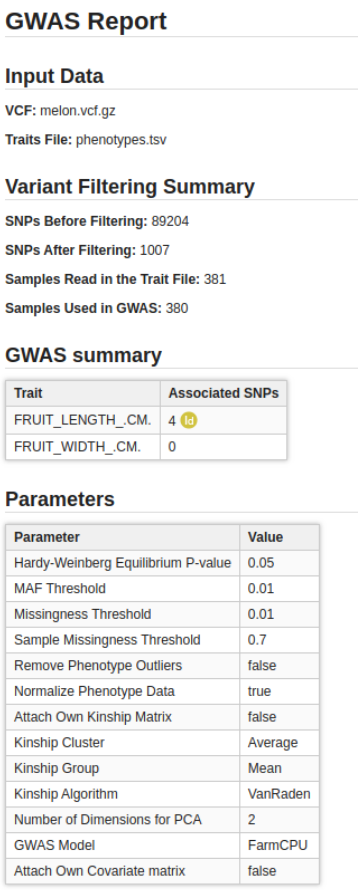


Figure 8. *Manhattan Plot of the Yield Per Plant Phenotype*

Summary Report

Report with information of the filtering step and the GWAS itself:

- **Filtering Summary:** information about the number of SNPs and samples before and after the filtering (see filtering parameters).
- **GWAS Summary:** number of significant SNPs associated to different phenotypes.



GWAS Report

Input Data

VCF: melon.vcf.gz

Traits File: phenotypes.tsv

Variant Filtering Summary

SNPs Before Filtering: 89204

SNPs After Filtering: 1007

Samples Read in the Trait File: 381

Samples Used in GWAS: 380

GWAS summary

Trait	Associated SNPs
FRUIT_LENGTH_.CM.	4
FRUIT_WIDTH_.CM.	0

Parameters

Parameter	Value
Hardy-Weinberg Equilibrium P-value	0.05
MAF Threshold	0.01
Missingness Threshold	0.01
Sample Missingness Threshold	0.7
Remove Phenotype Outliers	false
Normalize Phenotype Data	true
Attach Own Kinship Matrix	false
Kinship Cluster	Average
Kinship Group	Mean
Kinship Algorithm	VanRaden
Number of Dimensions for PCA	2
GWAS Model	FarmCPU
Attach Own Covariate matrix	false

Figure 9. *GWAS Summary Report*

How to Get Variant Information from Manhattan Plot

1- First of all, you must save a Variant Annotation Object of the same VCF file as the one used for GWAS. Then, once you have a GWAS table, you have to click in the **Charts** sidebar, and then in **Phenotype Information** in order to get the same wizard as in figure 10.

2- Subsequently, you will have to select the Manhattan Plot and the Variant Annotation Object mentioned before.

3- Finally, click on ‘Run’.

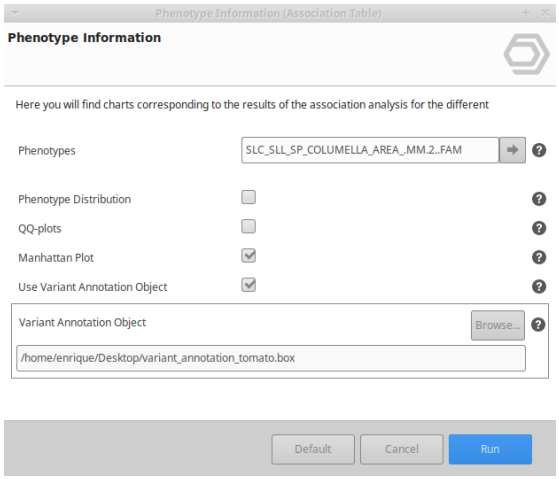


Figure 10. *Phenotype Information Wizard*

After some time (depending on the number of SNPs used in the GWAS), a Manhattan Plot will appear with a Selection Interface (figure 11). Select all the interesting variants (for example, all the variants significantly associated with a phenotype) and a subset of the Variant Annotation Object with information of those variants will appear (figure 12).

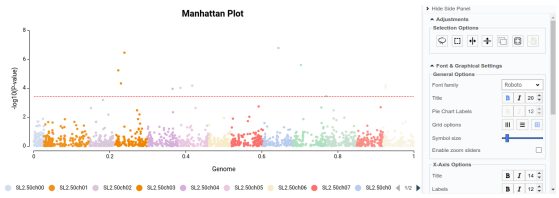


Figure 11. *Manhattan Plot with the Selection Interface*

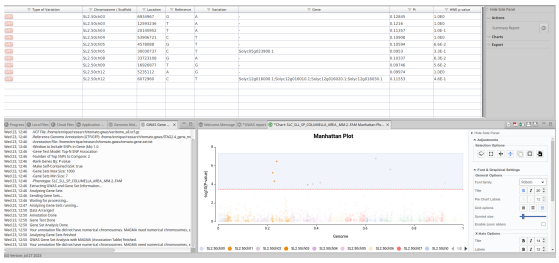


Figure 12. *Subset of Variant Annotation Object with Significantly-Associated Variants*

Gene Set Analysis with MAGMA

INTRODUCTION

MAGMA (Multivariate Analysis of Genomic Annotation) is a tool designed for the analysis of Genome-Wide Association Study (GWAS) results, facilitating researchers the integration of genetic information with functional annotations, allowing for gene tests, which assess if a gene is related to a change in a phenotypic trait. Moreover, with the gene-level p-values calculated by MAGMA, this tool is able to identify genes sets (GO, pathways, enzymatic complexes, etc) that may be involved in the phenotype under study. This gene-centric approach is crucial for understanding the biological mechanisms underlying complex traits and diseases, aiding in the prioritization of target genes and/or gene sets for further investigation. With its ability to interpret GWAS results at gene level, MAGMA proves indispensable in unraveling the genetic basis of various traits and diseases.

MAGMA consists of three main steps:

1. Annotation: MAGMA starts by mapping variants to their corresponding genes.
2. Gene Test: MAGMA performs a gene-based test by aggregating the variant-level association statistics within each gene. This consolidation provides a gene-level statistic, such as a p-value, that quantifies the evidence for association between the gene and the phenotype of interest.
3. Gene Set Analysis: MAGMA also enables gene set analysis, where it aggregates gene-level statistics into functional gene sets (GOs, pathways, etc.). This analysis allows for the identification of groups of genes that collectively contribute to the phenotype, providing a broader perspective on the biological mechanisms involved.

RUN MAGMA IN OMICSBOX

MAGMA can be found in the **Sidebar of the GWAS object**.The wizard consists of 2 pages and allows to define the input and the analysis parameters (Figure 1 and Figure 2).

Input

In this page you will be able to select the phenotype you want to do GSA with and the necessary files.

- **Phenotype:** select one of all the phenotypes analysed using GWAS in order to do a GSA analysis with the corresponding p-values that each SNP has regarding that trait.
- **VCF File: the same VCF** used to perform the GWAS analysis. This VCF file will be used to get the position and chromosome of each SNP in order to map it to the corresponding gene.
- **Reference Genome Annotation (GTF/GFF):** file to get the gene coordinates in order to map SNPs inside.
- **Annotation file:** file with data of each gene set. Gene sets can be GOs, KEGG IDs, enzymes, etc. Nevertheless, regardless of the type of gene set, this file must have one of the following two formats:
- Option 1:
  GeneSet1{TAB}Gene1
  GeneSet1{TAB}Gene2
  GeneSet2{TAB}Gene3
- Option 2:
  GeneSet1{TAB}Gene1, Gene2
  GeneSet2{TAB}Gene3, Gene4

MAGMA in OmicsBox can accept .box files with annotations, .annot files, or .txt files with the previous formats.

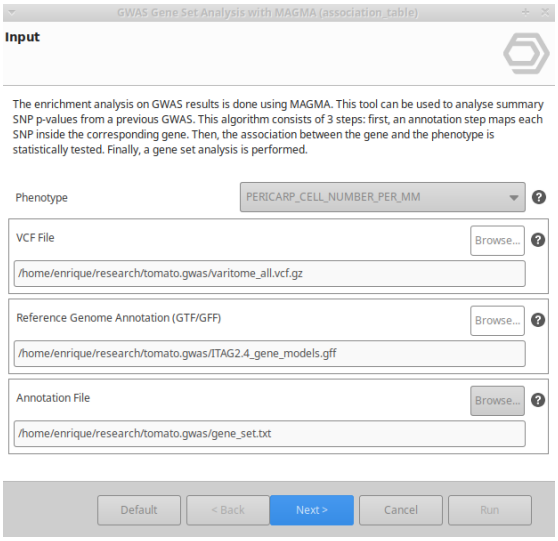


Figure 1. *Input Page*

Configuration

In this page, you will be able to select different parameters for each of the steps run by MAGMA.

- Window to Include SNPs in Gene (kb): Select the window (in kb) to look for SNPs around genes. By default, no window is added.
- Gene Test Model: Select the model to make the Gene Test. The Multiple Linear Principal Regression model is recommended when a Covariate Matrix is available. Nevertheless, each of the three models uses different concepts to test if a gene is associated with a genotype:
- **Multiple Linear Principal Components Regression**: A PCA is done. A regression with the principal components is performed in order to get which SNPs are more correlated with the phenotypes.
- **Mean SNP-wise association:** A distribution of SNPs p-values is done. Then sampling distribution to obtain gene p-value.
- **Top SNP-wise association:** Use Top-N SNPs and an empirical gene p-value is obtained using an adaptive permutation procedure.

The Multiple Linear Principal Components Regression does an internal QC (SNPs must have both a MAF <= 0.01 and a MAC <= 100). That is why some SNPs will disappear, hence some genes will not appear and results might not be consistent with the other two methods.

- Number of Top SNPs to Compare: Number of Top SNPs to compare when the Top-N SNP Assocation model is chosen.
- Rank Genes By: Select the column to rank SNPs for the Gene Analysis. We recommend the p-value column, as the adjusted p-value column might have more rank ties. **This is only necessary when the Multiple Linear Principal Components Regression Model is NOT used.**
- Make Self-Contained GSA: By default, the GSA that is performed is competitive. That is to say: GSA tests that genes in a gene set are more associated to a phenotype than other genes.
  Self-contained GSA tests that genes in a gene set are jointly associated with a phenotype. When the real causal SNPs are fully contained in one particular gene set, both test are approximately equally significant.
  However, when SNPs in multiple gene sets are associated with the disease or when causal genes are shared by multiple gene sets, using competitive tests may result in loss of power.
  Nevertheless, in a GWAS analysis, it is not likely that SNPs are equally distributed in all gene sets.
- Gene Sets Min Size: minimum number of genes in a set to take it into consideration.
- Gene Sets Max Size: maximum number of genes in a set to take it into consideration.

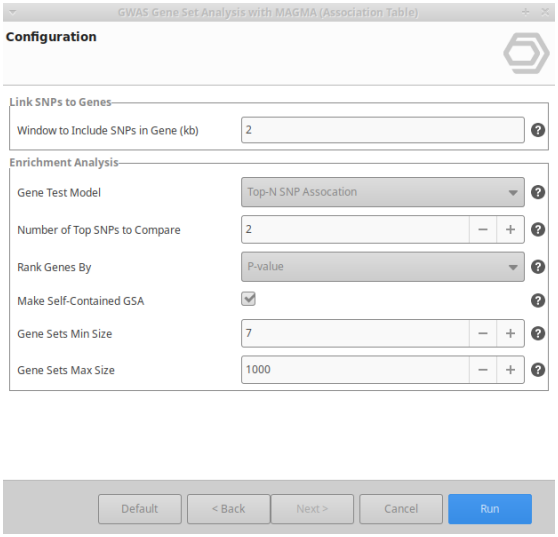


Figure 2. *Configuration Page*

RESULTS

- **Main table:** the main table will contain the gene sets tested as associated to the phenotype and different information about them:
- Significance: a red tag will appear when the test significantly associate a gene set with a phenotype.
- ID: identification of the gene set in the Gene Set File.
- GO (Optional): in case that GO IDs are used, the GO name will appear in this field. If other type of gene sets are used (KEGGs, enzyme families, etc.), this column will not appear.
- Number of genes: number of genes with SNPs that are present in the gene set.
- P-Value: estimates the statistical significance of the enrichment score for a single gene set.
- FDR: corrected p-value using the Benjamini-Hochberg method. Estimated probability that a gene set represents a false positive finding.

If you have chosen the self-contained gene set analysis, two tabs will appear on Omicsbox: one with the competitive gene set analysis, and another one with the self-contained gene set analysis.

- **Gene Set Details:** if you right-click in a gene set and then you click on "Show Gene Set Details" you will be able to see more information about the Gene Set (in case it is a Gene Ontology). In addition, information about the genes that have SNPs in the VCF file and about the SNPs themselves is displayed.

The gene p-value determines the significance of the association between the gene and the phenotype, and the SNP p-value is the same as the GWAS p-value (i.e., significance of the association between the SNP and the phenotype).

- **SNPs per gene:** a high proportion of SNPs analysed in a GWAS might belong to intergenic regions. With this chart you will be able to check whether the MAGMA analysis possess a significant number of SNPs inside genes.
- **Sidebar options:**
- Actions:
  - Word Cloud: Representation to summarise relevant gene sets in a fashionable way.
  - MAGMA Bar Chart: barchart with the main gene sets and their percentage of genes with SNPs from the total of genes that were grouped in that gene set.
  - Make Enriched Graph: use this option if you have GOs as Gene Sets to generate a representation on the GO DAG (see image below). Nodes are color-highlighted proportionally to their significance value. The user can choose which type of calculated p-value to use for highlighting and the threshold for filtering out nodes.
  - Reload Tags: choose between p-value or FDR column to update the red tag.

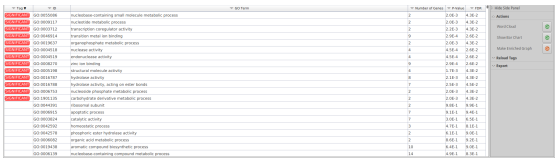
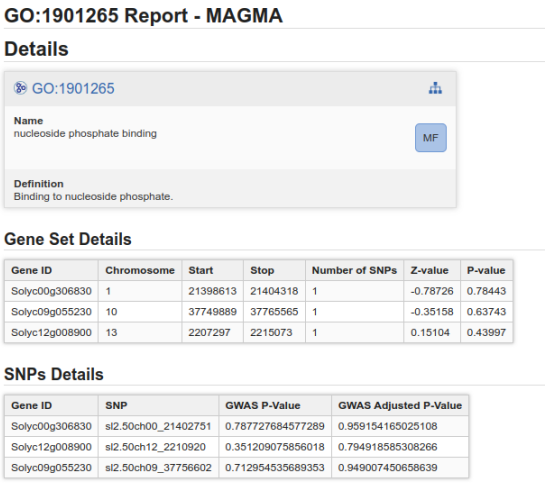


Figure 3. *MAGMA Table with Gene Set Information*



GO:1901265 Report - MAGMA

Details

GO:1901265

Name
nucleoside phosphate binding

MF

Definition
Binding to nucleoside phosphate.

Gene Set Details

Gene ID	Chromosome	Start	Stop	Number of SNPs	Z-value	P-value
Solyc00g306830	1	21398613	21404318	1	-0.78726	0.78443
Solyc09g055230	10	37749889	37765565	1	-0.35158	0.63743
Solyc12g008900	13	2207297	2215073	1	0.15104	0.43997

SNPs Details

Gene ID	SNP	GWAS P-Value	GWAS Adjusted P-Value
Solyc00g306830	sl2.50ch00_21402751	0.787727684577289	0.959154165025108
Solyc12g008900	sl2.50ch12_2210920	0.351209075856018	0.794918585308266
Solyc09g055230	sl2.50ch09_37756602	0.712954535689353	0.949007450658639

Figure 4. *Gene Set Details Report*

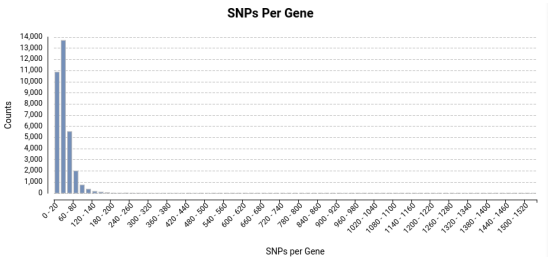


Figure 5. *Chart of SNPs Per Gene*

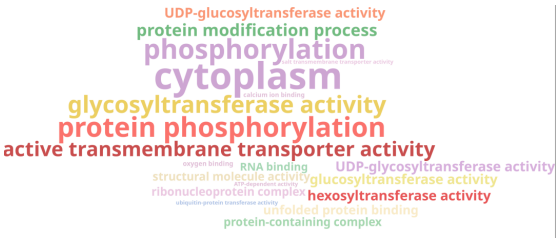


Figure 6. *Word Cloud*

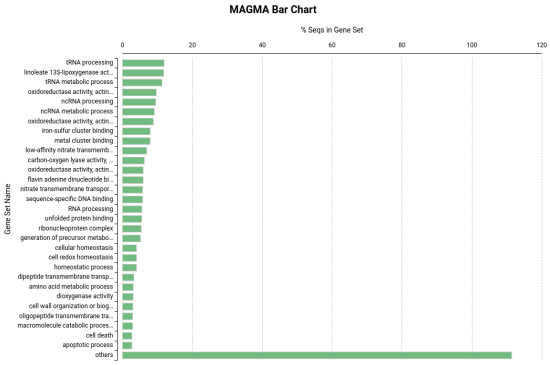


Figure 7. *MAGMA Bar Chart*

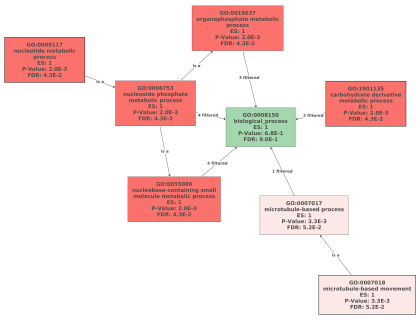
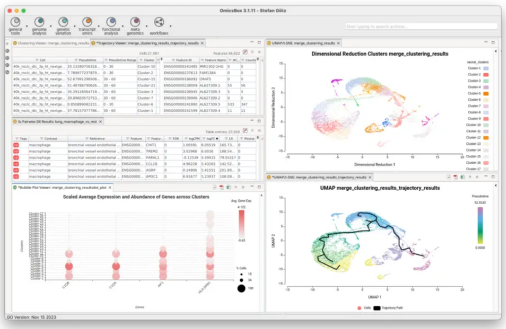


Figure 8. *Enriched GO Graph*

4.4 Module Transcriptomics

4.4.1 Module Transcriptomics



The OmicsBox Transcriptomics module allows you to process RNA-seq data from raw reads down to their functional analysis in a flexible and intuitive way.

Quality Control: Use FastQC and Trimmomatic to conduct quality control on your samples, filter reads, and eliminate low-quality bases.

De-Novo Assembly: Assemble brief reads with Trinity to obtain a de novo transcriptome without a reference genome. Evaluate the completeness of the transcriptome using BUSCO and cluster similar sequences with CD-HIT. Additionally, you can predict coding regions with TransDecoder or assess the coding potential of each sequence using CPAT.

RNA-Seq Alignment: Align your RNA-seq data to the reference genome using STAR (Spliced Transcripts Alignment to a Reference) or BWA (Burrows-Wheeler Aligner), irrespective of your hardware. Furthermore, BAM-QC offers several useful modules for assessing RNA-seq alignment files.

Quantify Expression: Quantify expression at gene or transcript level through HTSeq or RSEM and with or without a reference genome.

Differential Expression Analysis: Detect differentially expressed genes between experimental conditions or over time with well-known and versatile statistical packages like NOISeq, edgeR, or maSigPro. Rich visualizations help to interpret results.

Long-Read Analysis: Use LongQC to evaluate the quality of long-read datasets in the absence of a reference genome. First, identify transcripts sequenced with long reads using IsoSeq3, FLAIR, or IsoQuant. Subsequently, conduct an in-depth analysis and characterization of the long-read transcriptome using SQANTI3. This process will result in a refined transcriptome along with a comprehensive analysis report.

Single-Cell RNA-Seq: Obtain scRNA-Seq counts seamlessly with STARsolo for different library-prep technologies. Perform Single-Cell RNA-Seq clustering with Seurat to identify groups of cells and examine marker genes' expression. Gain insight into cell transitions with Monocle3 and visualize cell lineage trajectories in pseudo-time.

Enrichment Analysis: By integrating the findings of differential expression with functional annotations, enrichment analysis enables the identification of both overrepresented and underrepresented biological functions.

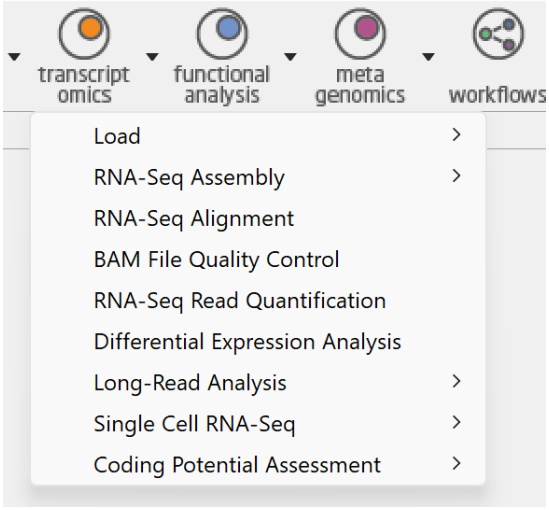


image-20240430-143721.png Additional Resources

- Transcriptomic Analysis use case: https://www.biobam.com/drug-response-transcriptomics/ .
- Transcriptomic Example Dataset: Download.

4.4.2 Load

Load external files into OmicsBox objects (Figure 1).

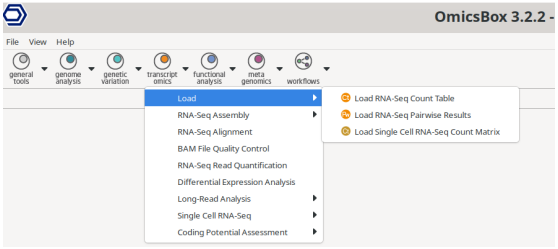


Figure 1. *Transcriptomics Load functions.*

LOAD RNA-SEQ COUNT TABLE

Load bulk RNA-Seq Count Tables from text files (Figure 2). The first line must contain the column (sample) names and the first column must contain gene/feature names (Figure 3).

- **Count Table File**. Specify the count table file in .txt, .tsv, or .csv formats.
- **Column Separator**. The character that separates the columns: tab (" "), space (" "), comma (","), or semicolon (";")
- **NA Values**. How to handle missing values:
- Skip Line: do not load the counts for the entire row (gene).
- Assume Zero Values: replace the NA value with a 0.

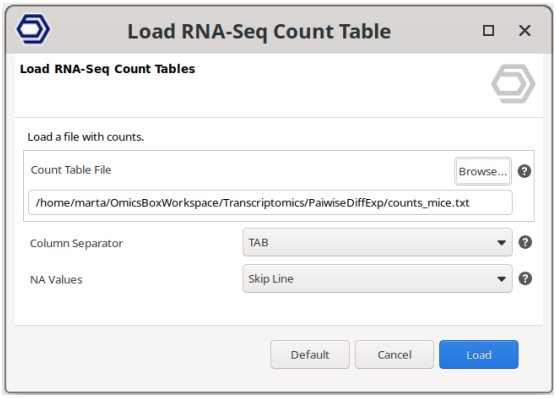


Figure 2. *Load RNA-Seq Count Table wizard.*

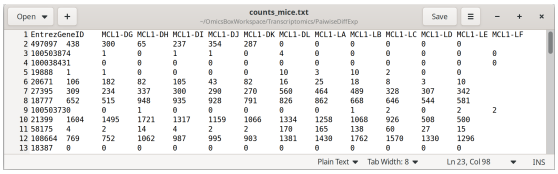
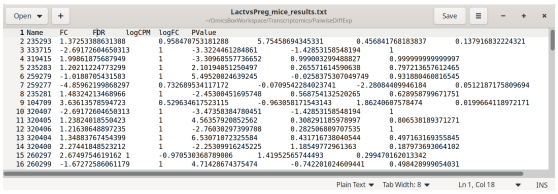


Figure 3*. Example RNA-Seq Count Table txt file, separated by tabs.*

LOAD RNA-SEQ PAIRWISE RESULTS

Load a tab-delimited file containing the results from a pairwise differential expression analysis (Figure 4). The table must meet the following conditions:

- It must contain the following mandatory fields: **logFC**, **logCPM**, **PValue,** and **FDR**.
- It must be **tab**-delimited.
- The type of information contained in each column must be indicated in its header (first line).
- The first column must contain the name of each sequence (gene or transcript).
- It may contain other column fields that are named according to their header.



	Name	FC	FDR	logCPM	logFC	PValue
2	235293	1.372533086631388		0.9584707531381288	5.754586943453331	0.45684176818383837	0.1379168322243421
3	333715	-2.691726046503113		1	-3.3224461284861	-1.4285315854819 4	1
4	319415	1.9986187568794 9		1	-3.3096855736652	0.9990032994888827	0.9999999999999997
5	235283	1.2921124773299		1	2.1019485125049 7	0.2655716145906 38	0.7972136576124 65
6	259279	-1.0180705431583		1	5.4952082463924 5	-0.0258375367049749	0.9318804680816545
7	259277	-4.859621598868297		0.7326895341171 72	-8.0709542284023741	-2.2008448994618 4	0.0512187175809694
8	235281	1.4832421346896 6		1	-2.4530845169574 8	0.5687541325202 65	0.6289587996717 51
9	104709	3.6361357859472 3		0.5296346175231 15	-0.9630581715431 43	1.8624068757847 4	0.0199664118972171
10	320407	-2.691726046503113		1	-3.4735838478045 1	-1.4285315854819 4	1
11	320405	1.2382401855042 3		1	4.5635792085256 2	0.3082911059789 97	0.8065381893712 71
12	320406	1.2163864889723 5		1	-2.7603029739970 8	0.2825068097075 35	1
13	320404	1.3488376745439 9		1	6.5307187232558 4	0.4317167386045 44	0.4971631693558 45
14	320408	2.2744184852321 2		1	-2.2538991624522 5	1.1854977296136 3	0.1879736930641 02
15	260297	2.6749754619162 1		-0.9705303607890 86	1.4195256574449 3	0.2994701620133 42	
16	260299	-1.6727258606117 9		1	4.7142867437547 4	-0.7422010246094 41	0.4984289990544 031

Figure 4. *Example Pairwise Results file.*

LOAD SINGLE-CELL RNA-SEQ COUNT MATRIX

Load Single-cell RNA-Seq (scRNA-Seq) count matrices in different formats into OmicsBox (Figure 5). An scRNA-Seq Count Matrix object will be opened in a new tab.

- **Input Type**. Select the format of the count table:
- Matrix Market File. The output of bioinformatics tools like CellRanger or STARsolo. It consists of three parts: an **MTX file** (.mtx) with the locations of non-zero counts, a **barcodes file** with cell IDs (column names), and a **features file** with gene names (row names). The MTX file must meet the specifications explained here. Some important points to note:
  - The barcodes and features files must not have a header.
  - The number of lines in the barcodes and features files must match the size specified in the MTX file (Figure 6).
  - The features file can have extra columns for feature metadata, separated by tabs. The first column should have Gene IDs, the second should have Gene Names, and any additional columns will be ignored.
  - Text File. Output of tools like RSEM, Drop-seq tools, etc. It is a text file similar to Figure 3, containing cells in columns and genes in rows.
  - Column Separator: Only enabled if Text File is selected. Specify the character that separates the columns: tab (" "), space (" "), comma (","), or semicolon (";")
  - CellRanger H5. The output of CellRanger with .h5 extension. Other count matrices in H5 are not supported. The CellRanger's format is explained in detail here.
  - H5 Annotated Data. Another type of h5 files (with the extension .h5ad) that follow the AnnData format. This format stores the count matrix in a group named "/X", the cell metadata in a group named "/obs", and gene metadata in a group named "/var". Further specifications can be found here. It is the most commonly used format in databases containing annotated scRNA-Seq references.
  - Loom Matrix File. It is the output of tools like Kallisto+BUStools, zUMIs, etc. It is another type of h5 file with a specific format, detailed here. The matrix has to be stored in a group named "/matrix", the cell names in a group named "/col_attrs/CellID", and the gene names in a group named "/row_attrs/Gene".

Additional options will be incorporated as new formats are developed to address user requirements.

When loading counts in MTX, Text, and H5 Annotated Data, **fractional counts** will be turned into integers. These fractional counts are generated by algorithms like Salmon, which distribute the counts of multi-mapping reads across the mapped genes, leading to decimal counts.

However, many scRNA-Seq analysis tools only accept integer counts. So, these fractional counts are converted to integers by removing the decimal part, not by rounding. This is done to avoid inflating the data size with many 0.5 counts being rounded up to 1, which wouldn't be a meaningful count.

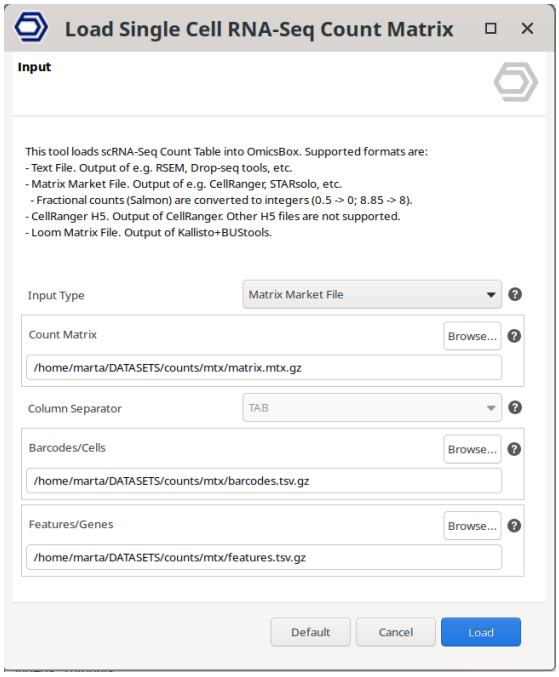


***Figure 5**. Load Single-cell RNA-Seq count matrix wizard.*

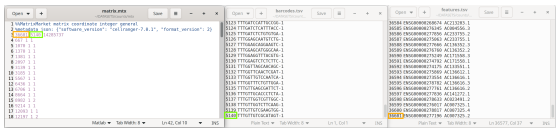


***Figure 6**. Example of a scRNA-Seq count matrix in MTX format.*

4.4.3 RNA-Seq de novo Assembly

Introduction

De novo transcriptome assembly is one of the most frequent analyses performed in bioinformatics and it consists of reconstructing the transcriptome from RNA sequencing data, assembling short nucleotide sequences into longer ones without the use of a reference genome. This functionality is based on **Trinity**, a well-known *de novo*sequence assembler software developed at the Broad Institute and the Hebrew University of Jerusalem.

Trinity combines three independent software modules applied sequentially to process large volumes of RNA-seq reads. Trinity partitions the sequence data into many individual de Bruijn graphs, each representing the transcriptional complexity at a given gene or locus, and then processes each graph independently to extract full-length splicing isoforms and to tease apart transcripts derived from paralogous genes.

Please, cite Trinity as:

Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, Adiconis X, Fan L, Raychowdhury R, Zeng Q, Chen Z, Mauceli E, Hacohen N, Gnirke A, Rhind N, di Palma F, Birren BW, Nusbaum C, Lindblad-Toh K, Friedman N, Regev A (2011). "Full-length transcriptome assembly from RNA-Seq data without a reference genome." Nature Biotechnology, 29(7):644-52.

Run RNA-Seq de novo Assembly

This functionality can be found under **Transcriptomics → RNA-Seq Assembly → RNA-Seq De novo Assembly**. The wizard allows to select files and set the parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

INPUT

- **Sequencing Data:** Choose the type of data to be preprocessed: single-end or paired-end reads. Note that if paired-end is selected, two files per sample are required.
- **Sequencing Format:** Select the format in which the sequencing reads are provided. All files should contain reads in the same format, FASTA or FASTQ.
- **Input Reads:**Provide the files containing sequencing reads. These files are assumed to be in FASTQ format.

If your data comes from SRA, be sure to dump the FASTQ file like so:

- Windows:
- SRA_TOOLKIT/fastq-dump --defline-seq @$sn[_$rn]/$ri --split-files SRR3233859
- Linux/MAC:
- SRA_TOOLKIT/fastq-dump --defline-seq '@$sn[_$rn]/$ri' --split-files SRR3233859
- **Paired-end configuration:**In the case of paired-end reads, the pattern to distinguish upstream files from downstream files is required. The provided patterns are searched right before the extension, and the start of the name should be the same for both files of each sample.
- Upstream Files Pattern: Establish the pattern to recognize upstream FASTQ files.
- Downstream Files Pattern: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is named SRR037717_1.fastq and the downstream one SRR037717_2.fastq, you should establish "_1" as the upstream pattern and "_2" as the downstream pattern.

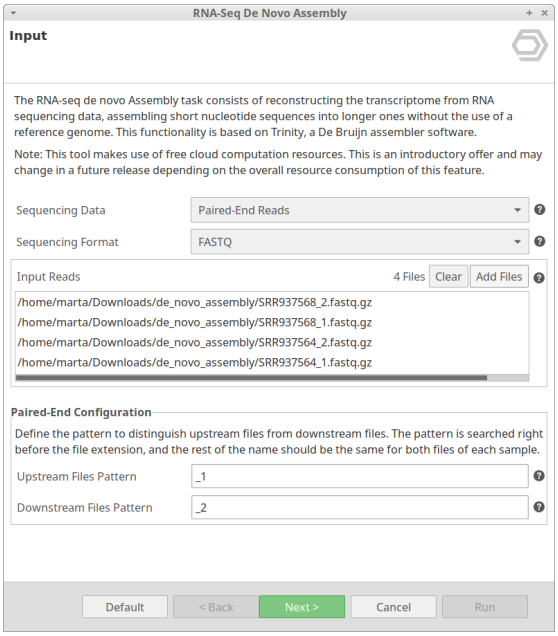


Figure 1:*Input Data Page*

GENERAL

- **Strand Specificity:**This option defines the strandedness of the RNA-seq reads:
- Non-Strand Specific: This refers to non-strand-specific protocols.
- Strand Specific Forward: For single-end data, the single read is in the sense (forward) orientation. In the case of paired-end data, the first read of fragment pair is sequenced as sense (forward), and the second is in the antisense strand (reverse).
- Strand Specific Reverse: For single-end data, the single read is in the antisense (reverse) orientation. In the case of paired-end data, the first read of fragment pair is sequenced as anti-sense (reverse), and the second read is in the sense strand (forward). Typical of the dUTP/UDG sequencing method.
- **Minimum Contig Length:** Minimum assembled contig length to report. Trinity uses 200 bp as default value.
- **Assess the Read Content:**To assess the read composition of the assembly, input RNA-Seq reads are aligned to the transcriptome assembly using Bowtie2. Reads that map to the assembled transcript are captured and counted, including the properly paired and those that are not. Check this option to obtain the read representation charts and table.

Note that it is an expensive operation, so the process will take more time.

- **Construct Super Transcripts:**SuperTranscripts provide a gene-like view of the transcriptional complexity of a gene. A SuperTranscript is constructed by collapsing unique and common sequence regions among splicing isoforms into a single linear sequence.

NORMALIZATION

- **Do Not Normalize Reads:**Trinity normalizes input reads to optimize the assembly procedure. Set this option to skip this step.

Normalization is highly recommended to deal with large datasets. Turning off normalization is not recommended for most applications.

- **Normalization Maximum Read Coverage:**Set the maximum read coverage to which the data will be normalized.

PAIRED-END CONFIGURATION

- **Minimizing Falsely Fused Transcripts:**If the transcriptome RNA-seq data under study are derived from a gene-dense compact genome, fusion transcripts can be minimized. This option is only available for paired-end data. In compact fungal genomes, it is highly recommended.

Note that it is an expensive operation, so avoid using it unless necessary.

- **Pair Distance:**Maximum length expected between fragment pairs (500 nucleotides by default). Reads outside this distance are treated as single-end.

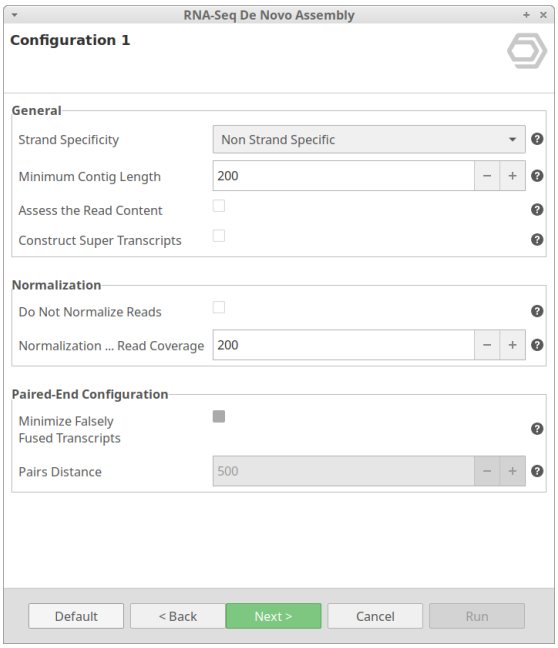


Figure 2. *Configuration Page 1*

INCHWORM

- **Minimum K-mer Coverage:** The minimum count for K-mers to be assembled by the Inchworm algorithm.

CHRYSALIS

- **Maximum Reads Per Graph:**The maximum number of reads to anchor within a single graph.
- **Minimum Glue:**The minimum number of reads needed to glue two Inchworm contigs.
- **Maximum Cluster Size:**The maximum number of Inchworm contigs to be included in a single Chrysalis cluster.

BUTTERFLY

- **Assembly Algorithm:**The assembly algorithm to use during the Butterfly step: the original algorithm or Pasafly. Pasafly is a PASA-like algorithm for maximally supported isoforms.
- **Path Reinforcement Distance:**The minimum overlap of reads with growing transcript path. Set to 1 for the most lenient path extension requirements.
- **No Path Merging:**By default, alternative transcript candidates are merged if they are found to be too similar. This is determined by taking into account similarity, mismatches, and gaps. If this option is checked, all final transcripts candidates are output (including SNP variations). Otherwise, if in a comparison between two alternative transcripts, they are found too similar, the transcript with the greatest cumulative compatible read (pair-path) support is retained, and the other discarded.
- **Minimum Percent Identity:**Minimum percent identity for two paths to be merged into single paths. The identity is calculated as the number of matches divided by the shorter length.
- **Maximum Allowed Differences:** Maximum allowed differences encountered between path sequences to combine them.
- **Maximum Internal Gap:**The maximum number of internal consecutive gap characters allowed for paths to be merged into single paths.

The parameters on this page are only for expert or experimental usage.

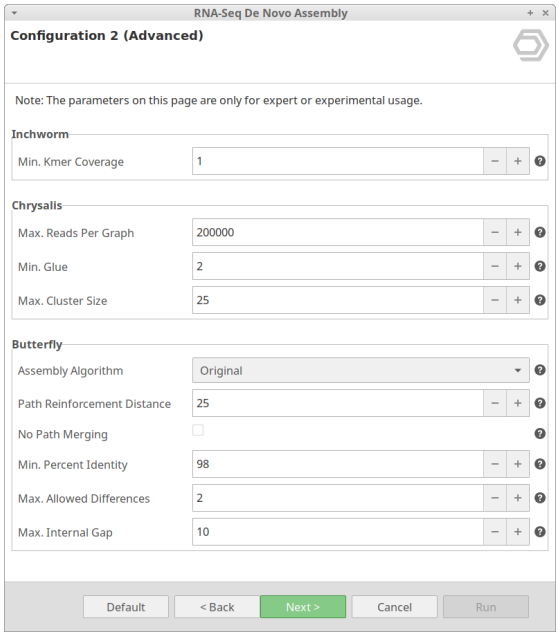


Figure 3. *Configuration Page 2*

OUTPUT

- **Transcripts**: Select a location to place the assembled transcripts in FASTA format.
- **Transcript to Gene Mapping:**Select a location to place the "transcript to gene" mapping file. It is a tab-delimited file with the information to map from transcript (isoform) identifiers to gene identifiers. It could be used in downstream analysis such as Transcript-level Quantification.
- **Super Transcripts**. Only available if the "Construct Super Transcripts" option is checked (Configuration 1 wizard page). Select the destination file to save the reconstructed super transcripts in FASTA format.
- **Super Transcripts GFF**. Only available if the "Construct Super Transcripts" option is checked (Configuration 1 wizard page). Select the destination file to save the super transcripts structures in GFF format.

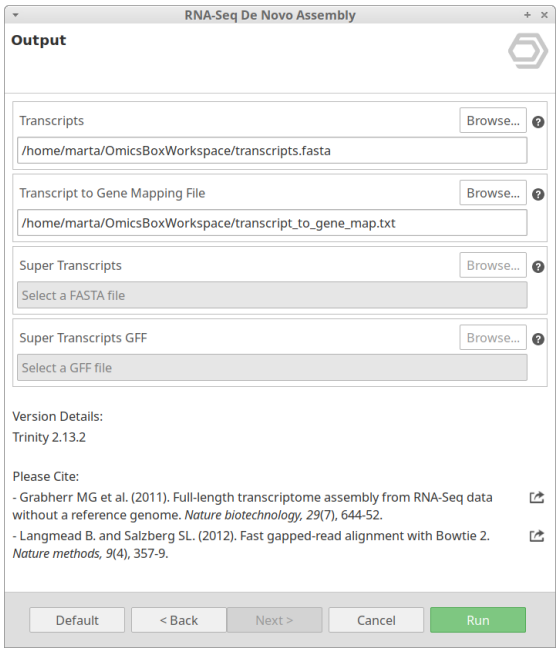


Figure 4:*Output Data Page*

Results

When the RNA-seq *de novo*assembly completes, it creates FASTA file containing the transcript sequences. This FASTA file can be loaded in OmicsBox as a sequence table (Figure 5). Trinity groups transcripts into clusters based on shared sequence content. Such a transcript cluster can be considered as a 'gene'. The tool also generates a transcript to gene mapping text file in the output folder.

This information is encoded in the Trinity FASTA accession as well. An example FASTA entry for one of the transcripts is formatted like so:

- Isoform 1: TRINITY_DN869_c0_g1_i1
- Isoform 2: TRINITY_DN869_c0_g1_i2

The accession encodes the Trinity 'gene' and 'isoform' information. In the example above, the accession 'TRINITY_DN869_c0_g1_i1' indicates Trinity read cluster 'TRINITY_DN869_c0, gene 'g1', and isoform 'i1' and 'i2'. Because a given run of trinity involves many clusters of reads, each of which are assembled separately, and because the 'gene' numbering is unique within a given processed read cluster, the 'gene' identifier should be considered an aggregate of the read cluster and corresponding gene identifier, which in this case would be 'TRINITY_DN869_c0_g1'.

If the "Construct Super Transcripts" option was checked, two additional outputs will be generated:

- SuperTranscripts in FASTA format.
- Transcript structure annotation in GFF format.

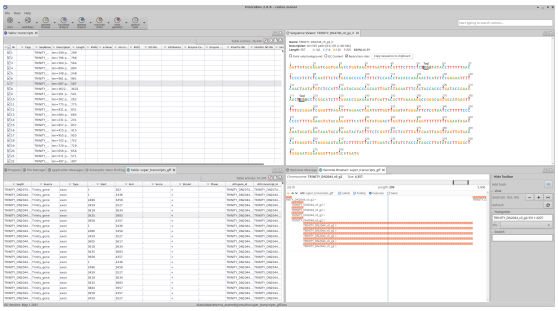


Figure 5:*Sequence table project containing the sequences of the assembled transcripts*

Furthermore, a result page will show a summary of the RNA-seq *de novo*assembly results (Figure 6). It contains the following information:

- Details of input FASTQ files.
- Results overview that informs about the number of total transcripts and genes detected, the percentage of GC, and the total assembled bases.
- Statistics based on the lengths of the assembled transcriptome contigs. The conventional Nx length statistic means that at least x% of the assembled transcript nucleotides are found in contigs that are at least of Nx length. For example, the N50 means that at least half of all assembled bases are in transcript contigs of at least the N50 length value.
- The RNA-Seq Read Representation, that allows assessing the read composition of the assembly. It shows the number of reads that map to the assembled transcripts, including the properly paired and those that are not (details below).

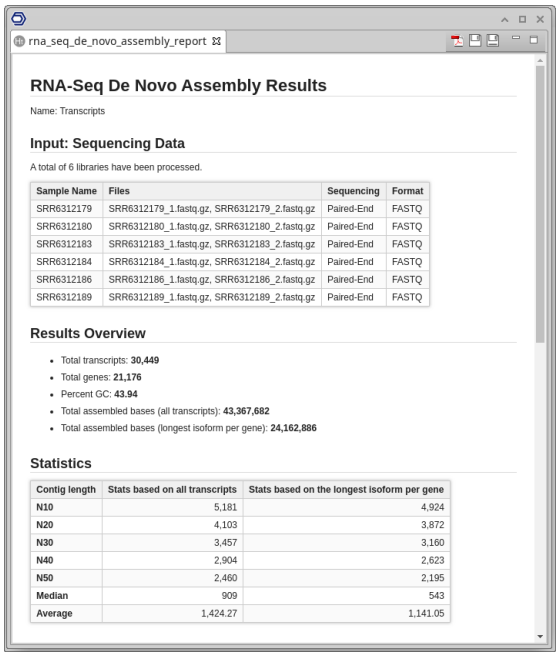


Figure 6:*Summary report*

Finally, two charts showing the read representation of the assembly are generated (Figure 7). These charts display the number of reads of each input file sorted by different categories (the second chart represents the same information in percentages). Bowtie2 is used to align the reads to the transcriptome and then the number of the single-end reads or proper pairs and improper or orphan read alignments are counted.

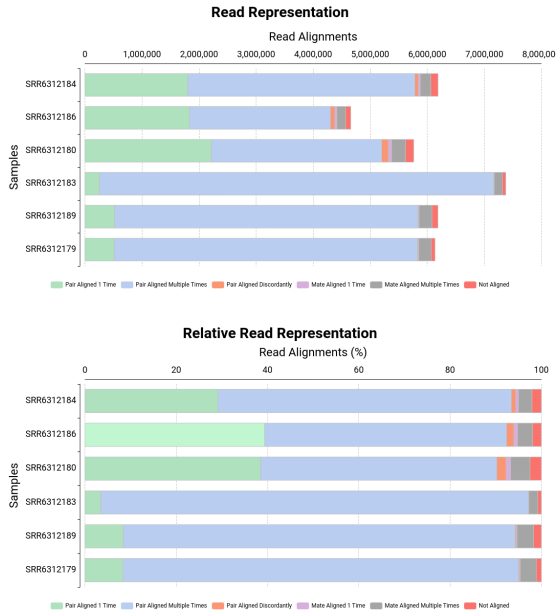


Figure 7:*Read Representation Chart*

4.4.4 Completeness Assessment

Introduction

The Completeness Assessment functionality provides quantitative measures for the assessment of transcriptome assembly completeness, based on evolutionarily-informed expectations of gene content from Benchmarking Universal Single-Copy Orthologs (BUSCO) selected from OrthoDB.

The Benchmarking Universal Single-Copy Orthologs are ideal for such quantifications of completeness, as the expectations for these genes to be found in a genome/transcriptome in single-copy are evolutionarily strong.

The application offers predefined BUSCO sets for six major phylogenetic clades. Sampling hundreds of genomes, orthologous groups with single-copy orthologs in >90% of species were selected. Importantly, this threshold accommodates the fact that even well-conserved genes can be lost in some lineages, as well as allowing for incomplete gene annotations and rare gene duplications.

OmicsBox offers predefined BUSCO datasets for six major phylogenetic clades:

- Bacteria
- Archaea
- Eukaryota
- Protists
- Fungi
- Plants

Please cite BUSCO and OrthoDB as:

Seppey M., Manni M. and Zdobnov EM. (2019). BUSCO: Assessing Genome Assembly and Annotation Completeness. Methods in molecular biology (Clifton, N.J.), 1962, 227-245.

Kriventseva EV., Kuznetsov D., Tegenfeldt F., Manni M., Dias R., Simao FA. and Zdobnov EM. (2019). OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial, and viral genomes for evolutionary and functional annotations of orthologs. Nucleic acids research, 47(D1), D807-D811.

Run Completeness Assessment

This functionality can be found under **Transcriptomics → RNA-Seq Assembly → Completeness Assessment.**The wizard allows to select input files and adjust analysis parameters (Figure 1 and Figure 2).

INPUT

- **Input Sequences**: Select the input file to be analyzed. Either a nucleotide FASTA file or a protein FASTA file (depending on the mode selected on the next page).

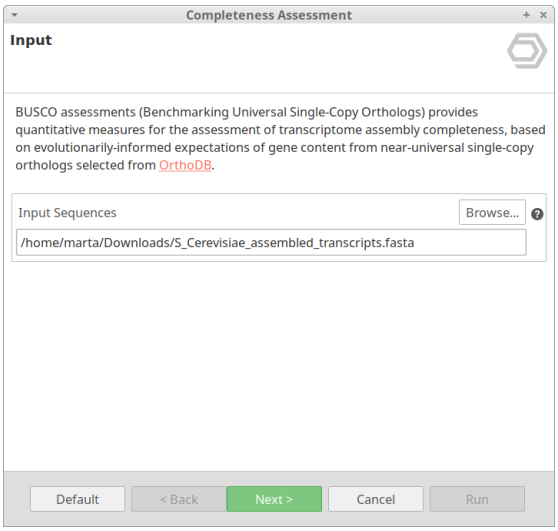


***Figure 1**. Input Wizard Page.*

CONFIGURATION

- **Lineage:**Choose the appropriate lineage-specific profile to classify matches, depending on the species to be assessed. Genes that make up the BUSCO sets for each major lineage are selected from orthologous groups with genes present as single-copy orthologs in at least 90% of species.
- **Mode:** Set the assessment mode according to the type of sequences to be analyzed.

- Transcriptome: nucleotide sequences (e.g. transcriptome *de novo*assembly).
- Proteome: Protein amino acid sequences.
- **Blast e-Value:**The statistical significance threshold for reporting matches against a sequence database. If the statistical significance of alignment is greater than the e-Value threshold, this hit will not be reported. Lower e-Value thresholds are more stringent, leading to fewer results. Increasing the threshold shows less stringent matches. The default e-Value used by BUSCO is 1e-03.

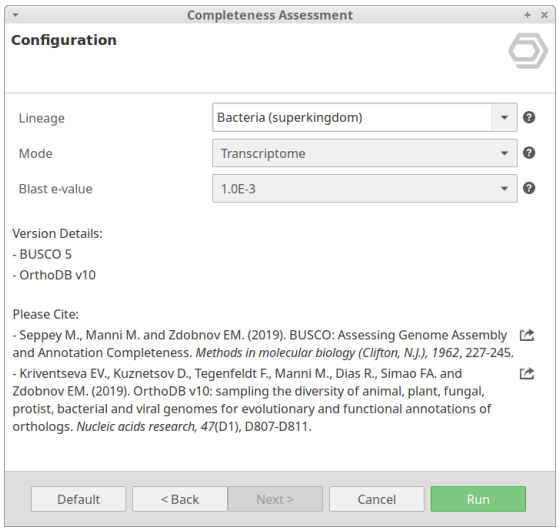


Figure 2.*Configuration Wizard Page*

Results

Once finished, a new tab is opened containing the results of the completeness assessment procedure (Figure 3). Each row corresponds to a BUSCO from the lineage database selected, and columns show the following information:

- BUSCO ID: Name of the BUSCO.
- Sequence ID: Name of the transcript/protein sequence matching the BUSCO.
- Score: Score of the alignment.
- Length: Length of the transcript/protein sequence matching the BUSCO.
- Tag: Result category.

The results are simplified into categories of Complete and single-copy, Complete and duplicated, Fragmented, or Missing BUSCOs:

- **Complete (single and duplicated):** The BUSCO matches have scored within the expected range of scores and within the expected range of length alignments to the BUSCO profile.
- **Fragmented:** The BUSCO matches have scored within the range of scores but not within the range of length alignments to the BUSCO profile. For transcriptomes or annotated gene sets, this indicates incomplete transcripts or gene models.
- **Missing:** There were either no significant matches at all, or the BUSCO matches scored below the range of scores for the BUSCO profile. For transcriptomes or annotated gene sets this indicates that these orthologous are indeed missing or the transcripts or gene models are so incomplete/fragmented that they could not even meet the criteria to be considered as fragmented.

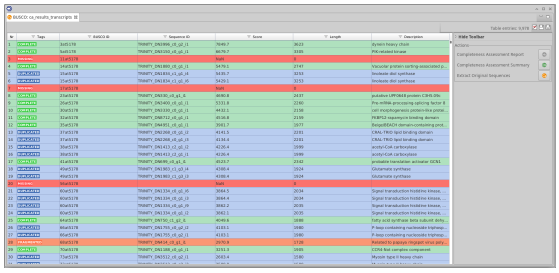


Figure 3.*BUSCO Project*

A result page will show a summary of the "Completeness Assessment" results (Figure 4). This page provides a quick evaluation of the results and provides ID lists containing BUSCO or transcript/protein identifiers assigned to the different categories. The result summary can be generated via **Side Panel → Actions → Completeness Assessment Report.**

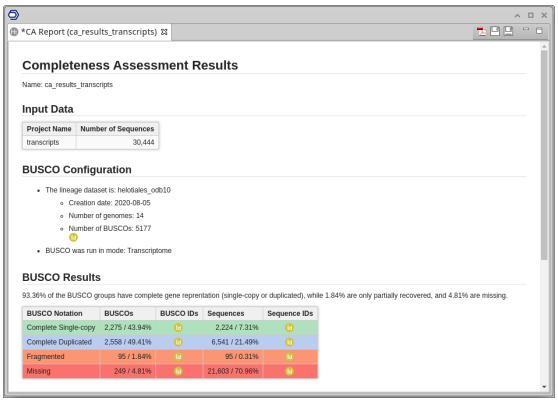


Figure 4.*Completeness Assessment Report*

Furthermore, the Completeness Assessment Summary chart (Figure 5) shows the percentage of lineage-specific BUSCOs assigned to each category. The pie chart can be generated via **Side Panel → Actions → Completeness Assessment Summary.**

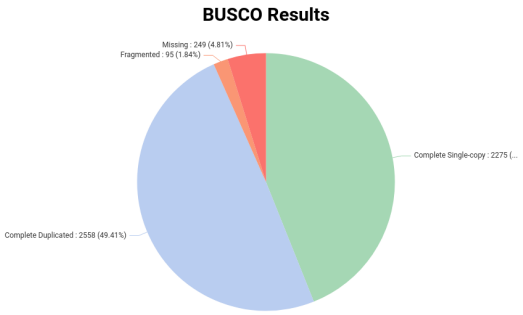


Figure 5.*Completeness Assessment Summary Chart*

Finally, the **Extract Original Sequences** utility (sidebar) allows extracting sequences from the original project based on its analysis status (Figure 6). For this, the original project containing the sequences that were assessed should be provided.

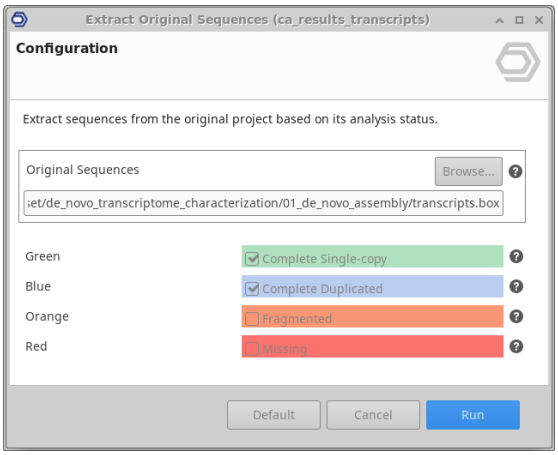


Figure 6.*Extract Original Sequences*

4.4.5 Predict Coding Regions

Introduction

The Predict Coding Regions functionality detects candidate coding regions within transcript sequences, such as those generated by *de novo* RNA-Seq transcript assembly. It is based on TransDecoder, a pipeline that recognizes likely coding sequences based on the following criteria:

- A minimum length open reading frame (ORF) is found in a transcript sequence.
- A log-likelihood score is computed and it should be > 0.
- The above coding score is higher when the ORF is scored in the 1st reading frame as compared to scores in the other 2 forward reading frames.
- If a candidate ORF is found fully encapsulated by the coordinates of another candidate ORF, the longer one is reported. However, a single transcript can report multiple ORFs (allowing for operons, chimeras, etc).
- A Position-Specific Scoring Matrix (PSSM) is built, trained and used to refine the start codon prediction.
- The putative peptide has a match to a Pfam domain above the noise cut-off score (optional).

Please cite TransDecoder as:

- Haas BJ et al. (2013). De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nature protocols, 8(8), 1494-512.
- TransDecoder 5.5.0. Haas, BJ. and Papanicolaou, A. 2019. https://github.com/TransDecoder/TransDecoder/wiki

Run Predict Coding Regions

This functionality can be found under **Transcriptomics → Assembly → Predict Coding Regions.** The wizard allows to provide input files and adjust analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

INPUT

- **Input Sequences**: Select a FASTA file containing input nucleotide sequences (e.g. assembled transcripts).

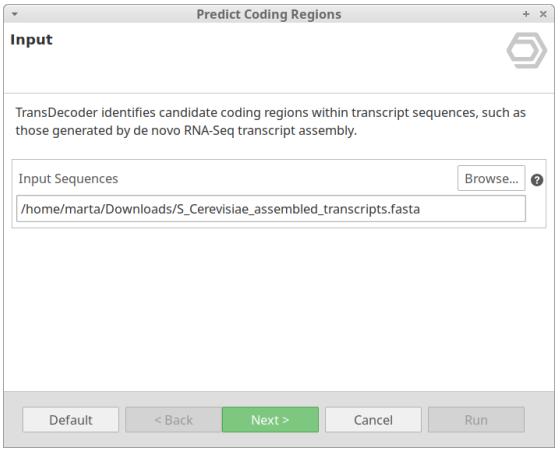


***Figure 1**. Input Wizard Page*

EXTRACT THE LONG ORFS CONFIGURATION

- **Genetic Code:** Select the genetic code of the organism under study. The available genetic codes are:

Available genetic codes

- Universal
- Acetabularia
- Candida
- Ciliate
- Dasycladacean
- Euplotid
- Hexamita
- Mesodinium
- Mitochondrial Ascidian
- Mitochondrial Chlorophycean
- Mitochondrial Echinoderm
- Mitochondrial Flatworm
- Mitochondrial Invertebrates
- Mitochondrial Protozoan
- Mitochondrial Pterobranchia
- Mitochondrial Scenedesmus obliquus
- Mitochondrial Thraustochytrium
- Mitochondrial Trematode
- Mitochondrial Vertebrates
- Mitochondrial Yeast
- Pachysolen tannophilus
- Peritrich
- SR1 Gracilibacteria
- Tetrahymena

- **Minimum Protein Length:** Minimum protein length to retain coding regions.
- **Strand Specific:** Only the top strand option is analyzed.
- **Provide Gene-Transcript relation:** Provide a tab-delimited file with the information to map from transcript (isoform) IDs to gene IDs. Each line should be of the form: Gene ID[tab]Transcript ID.

HOMOLOGY SEARCH CONFIGURATION

- **Pfam Search:** Identify ORFs with homology to known proteins via Pfam searches. Searching PFAM allows identifying common protein domains, that are included as ORF retention criteria. Note that this option will significantly increase the execution time.

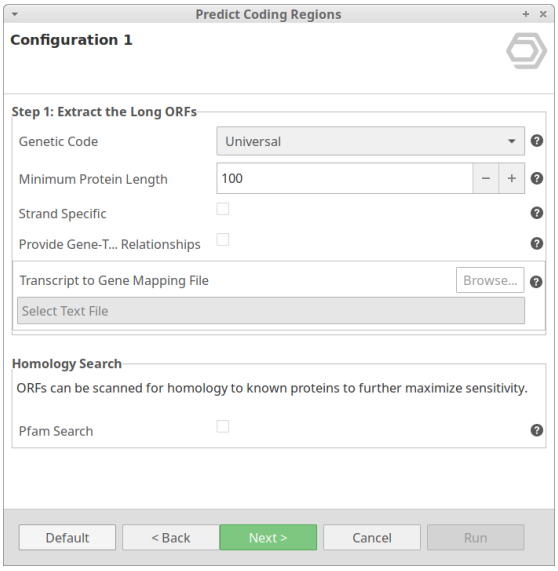


***Figure 2**. Configuration Page 1*

PREDICT THE LIKELY CODING REGIONS CONFIGURATION

- **Retain Long Orfs Mode:** Select the retain long ORFs strategy. The dynamic mode sets range according to 1% FDR in a random sequence of the same GC content. Under the strict mode, all ORFs found that are equal or longer to the Retain Long ORFs Length are kept, even if no other evidence marks it as coding.
- **Retain Long Orfs Length:** Select the minimum length to retain ORFs under the strict mode.
- **Single Best Only:** Retain only the single best ORF per transcript (prioritized by homology, then ORF length).
- **No Refine Starts:** By default, the predict coding regions strategy identifies potential start codons for 5' partial ORFs using a PWM (position weight matrix). Check this option to deactivate this process.
- **Top Longest ORF for Training:** Top longest ORFs to train Markov Model (hexamer stats). The default value is 500. Note, 10X this value is first selected for removing redundancies, and then the value of the longest ORF is selected from the non-redundant set.

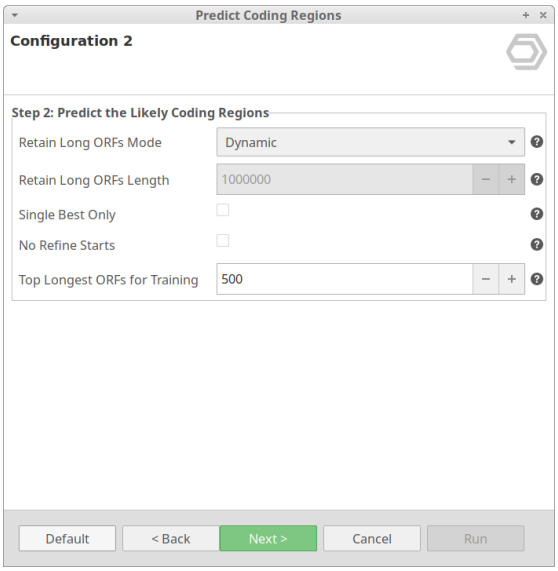


***Figure 3.** Configuration Page 2*

OUTPUT

- **Predicted CDSs**: select the destination file to save the predicted CDSs in FASTA format.

- **Predicted Proteins:** select the destination file to save the predicted proteins in FASTA format.
- **Coding Regions Coordinates:** select the destination file to save the predicted coding regions coordinates in GFF format.

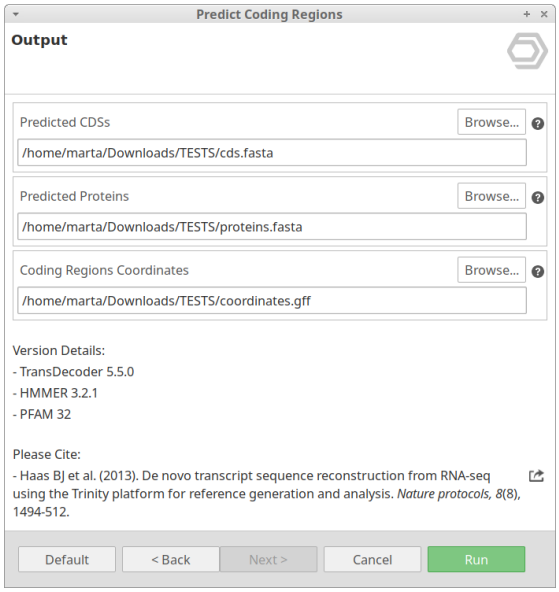


Figure 4. *Output Wizard Page.*

Results

Once finished, three files are generated. These files can be loaded in OmicsBox (Figure 5):

- **proteins.fasta**: FASTA file that contains peptide sequences for the final candidate ORFs.
- **cds.fasta**: FASTA file that contains nucleotide sequences for coding regions of the final candidate ORFs.
- **coordinates.gff**: GFF file that contains positions within the target transcripts of the final selected ORFs.

Note that in both FASTA files, CDSs and proteins, the description field contains details about the predicted ORF. This description includes:

- The protein identifier. It is composed of the original transcripts along with '|m.(number)'.
- The type attribute indicates whether the protein is:
- **Complete:** Contains a start and a stop codon.
- **5' partial**: It is missing a start codon and presumably part of the N-terminus.
- **3' partial**: It is missing the stop codon and presumably part of the C-terminus.
- **Internal**: It is both 5' and 3' partial.
- An indicator (+) or (-) to indicate in which strand the coding region was found, along with the coordinates of the ORF in that transcript sequence.

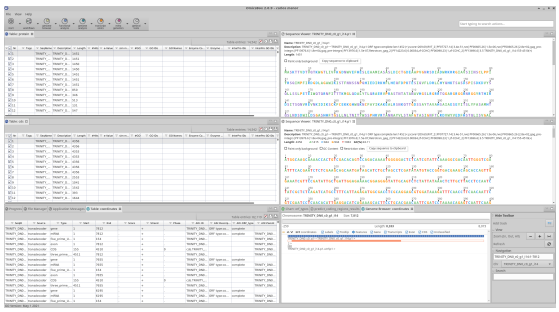


Figure 5. *Predict Coding Regions Results*

In addition, a result page will show a summary of the "Predict Coding Regions" results (Figure 6). This page provides a quick evaluation of the results and provides ID lists containing transcript identifiers assigned to the different categories.

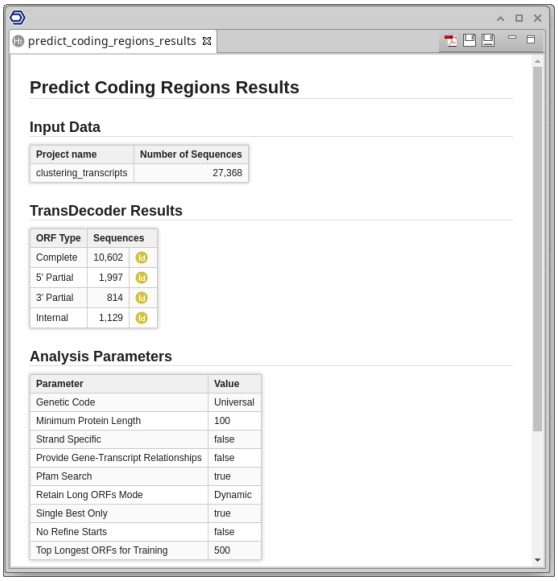


Figure 6. *Predict Coding Regions Report*

Furthermore, the Predict Coding Regions Summary chart (Figure 7) shows the percentage of ORFs that have been predicted as Complete, 5' Partial, 3' Partial, and Internal.

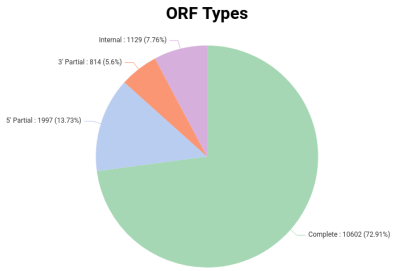


Figure 7. *Predict Coding Regions Summary*

4.4.6 Clustering

Introduction

With the advancement of next-generation sequencing technologies, the amount of available sequencing data is growing exponentially. Removing redundancy from such data could be crucial for reducing storage space, computational time, and noise interference in some analysis methods. The Clustering functionality allows to cluster sequence data to reduce this redundancy.

The clustering functionality is based on CD-HIT, a widely used program for clustering biological sequences. Basically, CD-HIT is a greedy incremental algorithm that starts with the longest input sequence as the first cluster representative and then processes the remaining sequences from long to short to classify each sequence as a redundant or representative sequence based on its similarities to the existing representatives. The similarities are estimated by common word counting using word indexing and counting tables to filter out unnecessary sequence alignments, which are used to compute exact similarities.

Run Clustering

This functionality can be found under **Transcriptomics → RNA-Seq Assembly → Clustering.**The wizard allows configuring analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

INPUT

- **Input Sequences**: Select a FASTA file containing input nucleotide sequences to be clustered (e.g. assembled transcripts).

Clustering Limitations

To ensure efficient and manageable computation times, CD-HIT clustering in OmicsBox has the following limitations:

1. **Sequence Identity Threshold Below 0.9**:
2. Datasets with a sequence identity threshold below 0.9 are not permitted if they contain more than 500,000 nucleotide sequences. Please adjust the threshold above 0.9 or reduce the dataset size.
3. **Sequence Identity Threshold Below 0.95**:
4. Datasets with a sequence identity threshold below 0.95 are not allowed if they contain more than 1,000,000 nucleotide sequences. Please set the threshold above 0.95 or use a smaller dataset.
5. **Maximum Dataset Size**:
6. Datasets exceeding 1,500,000 sequences are restricted to prevent excessive computation time. Please reduce the number of sequences in your dataset.

These limitations are in place due to the exponential nature of the clustering algorithm. Adhering to these guidelines helps ensure optimal performance and resource management.

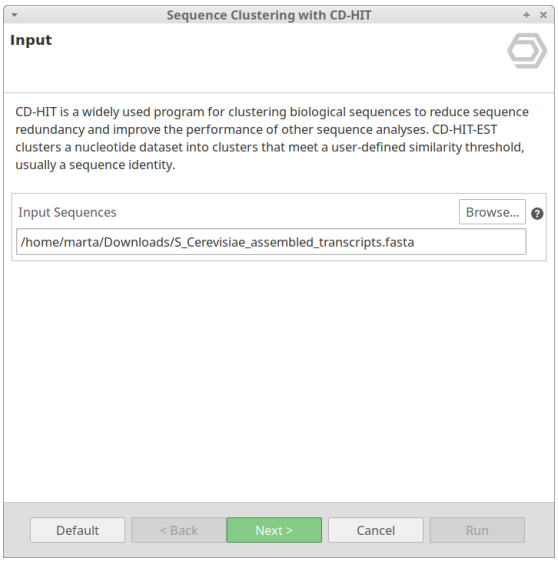


***Figure 1**. Input Page.*

ALGORITHM OPTIONS

- **Sequence Identity Type:** Sequence identity is calculated as:
- Global: number of identical bases in alignment divided by the length of the sorter sequence.

- Local: number of identical bases in alignment divided by the length of the alignment.

NOTE: The local option requires that the longer and shorter sequence coverage parameters are different from 0.

- **Sequence Identity Threshold:** Sequence identity threshold to consider clusters. Must be greater than or equal to 0.8
- **Band Width:** Band width of the alignment.
- **Word Length:** Word size for the alignments. Choose of word size:
- 10, 11 for thresholds 0.95 ~ 1.0.
- 8,9 for thresholds 0.90 ~ 0.95.
- 7 for thresholds 0.88 ~ 0.9.
- 6 for thresholds 0.85 ~ 0.88.
- 5 for thresholds 0.80 ~ 0.85.
- 4 for thresholds 0.8.
- **Length Cutoff:** Length of sequence to skip. Sequences below this length will be skipped.
- **Length Difference Cutoff:** Length difference cutoff. It is required as a proportion (0-1). If set to 0.9, the shorter sequences need to be at least 90% length of the representative of the cluster.
- **Accurate Mode:** By default, a sequence is clustered to the first cluster that meets the threshold (fast cluster). If this option is checked, the program will cluster it into the most similar cluster that meets the threshold (accurate but slow mode). This won't change the representatives of the final clusters.
- **Comparing Both Strands:** By default, the program does both, +/+ and +/- alignments. If this option is unchecked, the program only performs +/+ strand alignments.

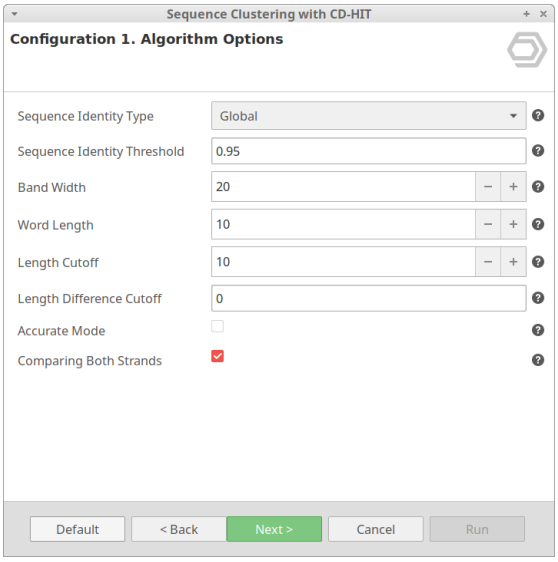


Figure 2. *Configuration Page 1*

ALIGNMENT COVERAGE OPTIONS

- **Adjust Longer Sequence Coverage:** Establish an alignment coverage for the longer sequence. This option is mandatory if the "Local" Sequence Identity Type will be used.
- **Longer Sequence Coverage:** Alignment coverage for the longer sequence. It is required as a proportion (0-1). If set to 0.9, the alignment must cover 90% of the longer sequence.
- **Adjust Shorter Sequence Coverage:** Establish an alignment coverage for the shorter sequence. This option is mandatory if the "Local" Sequence Identity Type will be used.
- **Shorter Sequence Coverage:** Alignment coverage for the shorter sequence. It is required as a proportion (0-1). If set to 0.9, the alignment must cover 90% of the shorter sequence.
- **Longer Sequence Unmatched %:** Maximum unmatched percentage for the longer sequence. If set to 0.1, the unmatched region (excluding leading and tailing gaps) must not be more than 10% of the longer sequence.
- **Shorter Sequence Unmatched %:** Maximum unmatched percentage for the shorter sequence. If set to 0.1, the unmatched region (excluding leading and tailing gaps) must not be more than 10% of the shorter sequence.

- **Alignment Position Constraints:** If it is checked, the program will force sequences to align at beginnings, and it only does +/+ alignment.

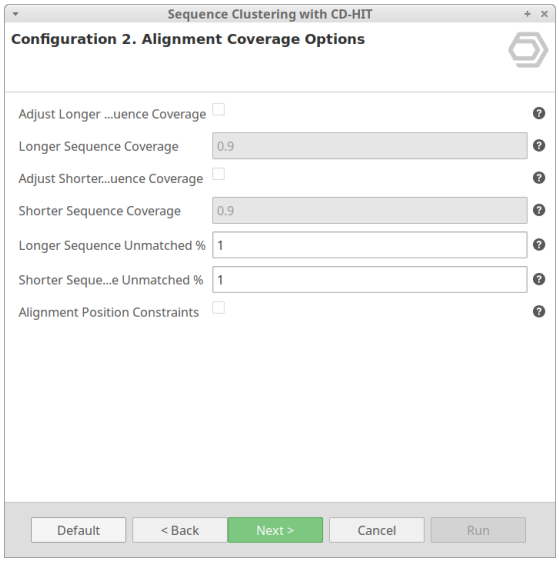


Figure 3. *Configuration Page 2*

OUTPUT DATA

- **Representative Sequences**: Select the destination file to save the representative sequence of each cluster in FASTA format.
- **Save Cluster File:** CD-HIT produces a file containing information about each cluster. Set this option to obtain this file.
- **Output Cluster File:** Select a file where the "cluster" file will be placed.

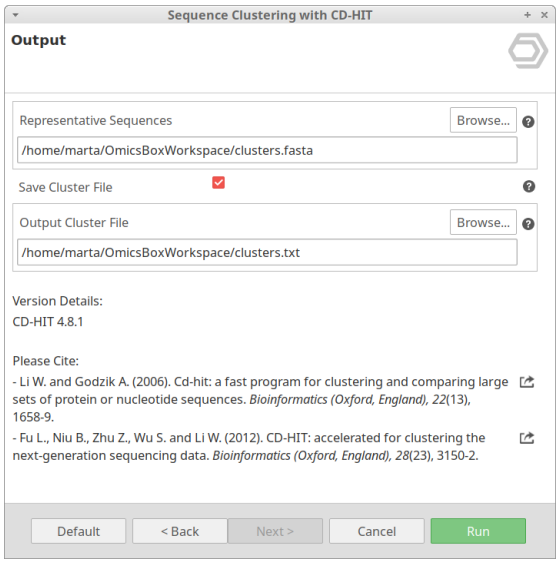


Figure 4. *Output Page*

Results

Once finished, results are returned in a project containing the representative sequence of each cluster. The SeqName field shows the identifier of the representative sequence. The Description field contains the sequence identifiers for the sequences that have been grouped into each cluster (Figure 4).

In addition, the "Cluster File" is a text file generated by CD-HIT. It contains information about each cluster, such as the sequences grouped in each cluster, what is the representative sequence and how similar are the sequences between them.

```
>Cluster 0 #Name of the cluster
0   227nt, >TRINITY_DN1539_c0_g1_i1... at 227:1:2041:2267/-/99.56% # Information about the similarity
```

```
14980nt, >TRINITY_DN1539_c0_g2_i1... * # Representative Sequence labeled with *
14977nt, >TRINITY_DN1539_c0_g2_i2... at 1:14977:1:14980/+/100.00%
```

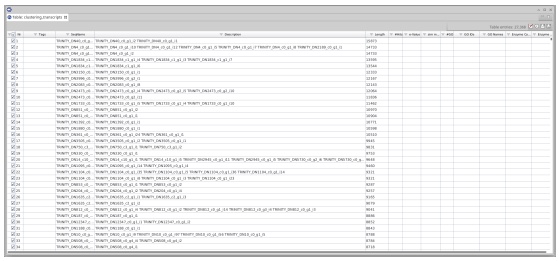


Figure 4: *Results Project*

Finally, a report page (Figure 5) will show a summary of the Clustering results. In the "CD-HIT Results" table, the number of clusters of different sizes is shown. In addition, a list of the representative sequences of each type of cluster can be obtained by clicking on the "Id" buttons.

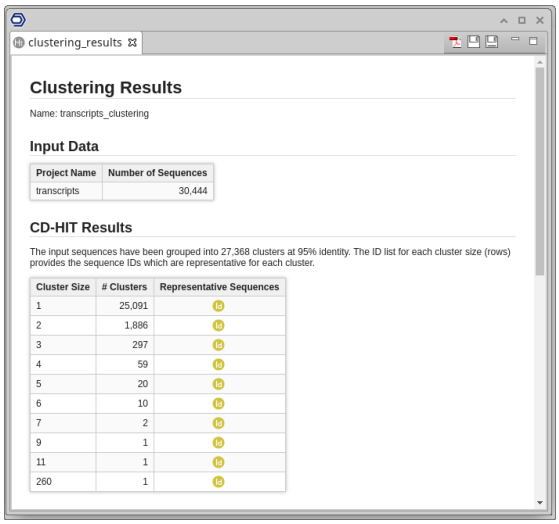


Figure 5: *Summary Report*

Furthermore, the Cluster Distribution chart (Figure 6) displays the number of clusters of different sizes that have been obtained.

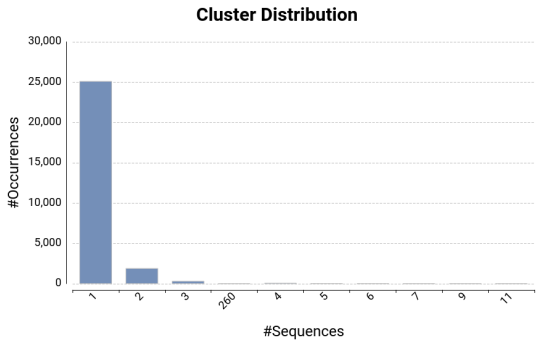


Figure 6: *Cluster Distribution Chart*

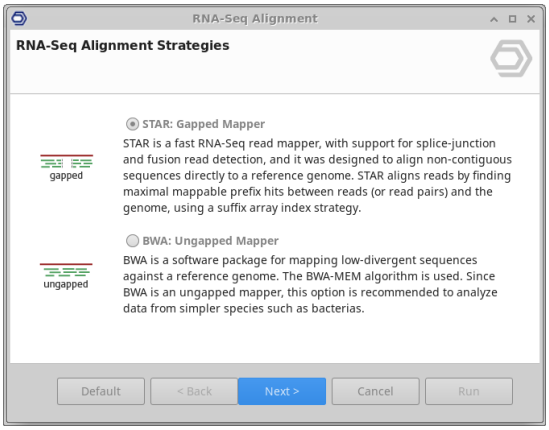
4.4.7 RNA-Seq Alignment

RNA-Seq Alignment

Read alignment is a common process applied to high-throughput sequencing data, being one of the first stages required for many different types of analysis. In the RNA-Seq scenario, this process is used to quantify gene expression. The goal of the read alignment is to map short sequencing reads efficiently to a large reference genome to identify the 'correct' genomic loci from which the read originated whilst taking into account errors in the sequence reads. This functionality can be found under **transcriptomics → RNA-Seq Alignments.**

Two alignment strategies are available:

- **STAR:**STAR is a gapped RNA-Seq read mapper, with support for splice-junction and fusion read detection, and it was designed to align non-contiguous sequences directly to a reference genome.
- **BWA:**BWA is a software package for mapping low-divergent sequences against a large reference genome. Since BWA is an ungapped mapper, this option is recommended to analyze data from simpler species such as bacteria.



RNA-Seq STAR

INTRODUCTION

STAR is a fast RNA-Seq read mapper, with support for splice-junction and fusion read detection, and it was designed to align non-contiguous sequences directly to a reference genome. STAR aligns reads by finding maximal mappable prefix hits between reads (or read pairs) and the genome, using a suffix array index strategy. Different parts of a read can be mapped to different genomic positions, corresponding to splicing or RNA-fusions. If genomic annotations are provided, the genome index includes known splice-junctions from annotated gene models, allowing for sensitive detection of spliced reads.

The binary nature of the suffix array search results in a favorable logarithmic scaling for the search time with the reference genome length. Since this approach has high memory requirements, it is executed in the Omics Cloud, which provides a high-performance computing environment, allowing fast alignments even against large genomes.

Please cite STAR as:

Dobin A, Davis CA, Schlesinger F, et al (2012). "STAR: ultrafast universal RNA-seq aligner." Bioinformatics, 29(1):15-21.

RUN RNA-SEQ ALIGNMENT (STAR)

This functionality can be found under **Transcriptomics → RNA-Seq Alignment → STAR.**The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, and Figure 3).

Input

- **Input Reads:**Provide the files containing RNA sequencing reads. These files are assumed to be in FASTQ format or compressed FASTQ format (.gz). Choose single-end or paired-end reads. Note that if the paired-end option is selected, two files per sample are required.
- **Paired-end configuration:**In the case of paired-end reads, a pattern to distinguish upstream files from downstream files is required. The provided patterns are searched in the filenames right before the extension. The beginning of the filenames should be the same for both files of each sample.
- **Upstream Files Pattern**: Establish the pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern**: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is SRR037717_1.fastq and the downstream SRR037717_2.fastq, "_1" should be established as the upstream pattern and "_2" as the downstream pattern.

- **Reference Genome:** Specify a FASTA file that contains the genome reference sequences. Multiple reference sequences, e.g. chromosomes or scaffolds, can be provided. It is strongly recommended to include major chromosomes as well as un-placed and un-localized scaffolds since a substantial number of reads may map to these scaffolds (e.g. ribosomal RNA). These reads would be reported as unmapped if the scaffolds are not included, or maybe aligned to the wrong loci on the chromosomes. On the other hand, patches and alternative haplotypes should not be included in the genome.

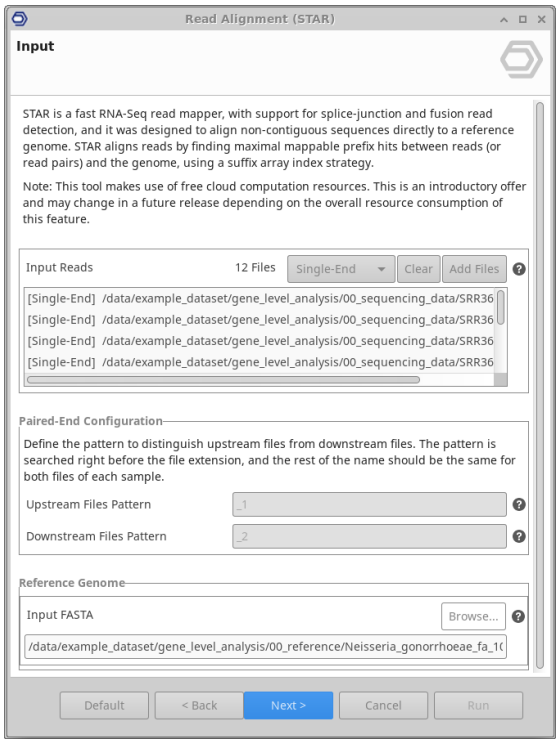


Figure 1: *Input Page*

Configuration

- **Provide annotations:**This option allows providing a file with annotated genes and transcripts in GTF/GFF format (GTF is recommended). The aligner will extract splice junctions from this file and use them to improve the accuracy of the alignment. While this is optional, using annotations is highly recommended. Chromosome names in the GTF annotation file have to match chromosome names in the FASTA genome sequences file.
- **Annotation File**: Select the file containing the annotated genes and transcripts in GTF/GFF format.
- **Overhang**: Establish the length of the genomic sequence around the annotated junction to be used in constructing the splice junctions database. This length should be equal to the length of the read -1. For instance, for 100 bp paired-end reads, the ideal value is 99. In the case of reads with varying lengths, the ideal value is the maximum read length -1.
- **2-pass Mapping:** This option allows a most sensitive novel junction discovery. The aligner algorithm is executed first to collect the junctions. These junctions are used for a second pass mapping.
- **Sort by Coordinate:** The aligner will output BAM files sorted by coordinates.
- **Minimum Intron Length:** Specify the minimum intron size. A genomic gap is considered an intron if its length is equal to or greater than the given value. Otherwise, it is considered a deletion.
- **Maximum Intron Length:**Specify the maximum intron size.
- **Maximum Number of Mismatches:** Set the maximum number of mismatches allowed per read or read pair.
- **Maximum Number of Multiple Alignments:**Establish the maximum number of multiple alignments allowed per read. If exceeded, the read is considered unmapped.
- **Include Chimeric Alignments:**This option allows to include the chimeric alignments together with normal alignments in the main BAM file. The format of chimeric alignments follows the latest SAM/BAM specifications.
- **Maximum Distance Between Mates:** Specify the maximum genomic distance between two mate pairs.
- **Add Read Group Information:**Include the 'Read Group' header (@RG) in output BAM files. This information may be required for downstream analysis of third-party tools. If this option is checked, the following read group tags will be included for each sample:
- Identifier (ID), automatically generated.
- The name of the sample (SM), inferred from file names.
- Sequencing Platform (PL), provided by the user.
- **Sequencing Platform:**Choose the sequencing platform which was used to obtain the input data. Consider that if this option is provided, all output BAMs will be tagged with the same platform.

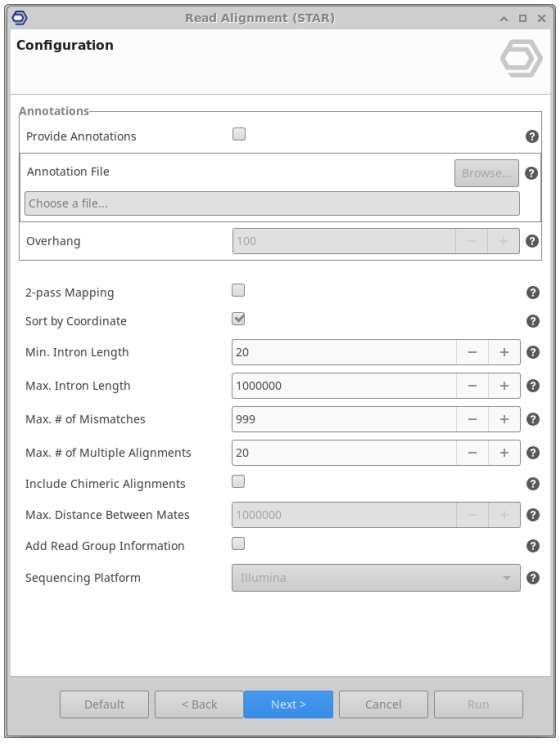


Figure 2: *Configuration Page*

Output

- **Save Splice Junctions:**Save high confidence collapsed splice junctions in tab-delimited format. Note that STAR defines the junction start/end as intronic bases. Files will be named as sample_name.Sj.out.tab, and will be placed in the "Alignment Files" destination folder.
- **Save Unmapped Reads:**Save unmapped and partially mapped. Files will be named as 'sample_name_Unmapped.fastq.gz' and will be placed in the "Alignment Files" destination folder.

- **Output BAMs Folder:**Select a folder to save the output BAM files. Take into account that one BAM file will be generated for each input FASTQ sample, so make sure there is enough disk space to store them.

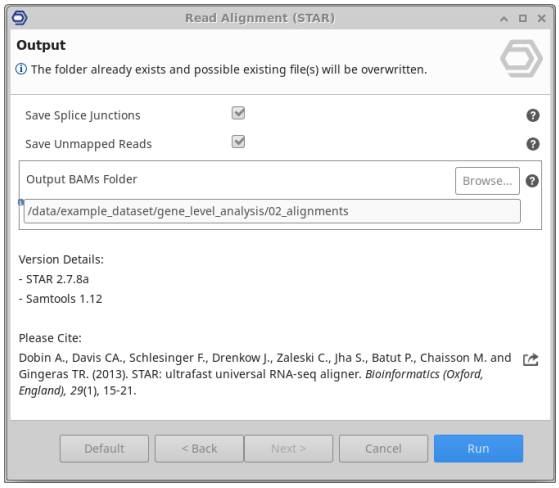


Figure 3: *Output Page*

RESULTS

The main outputs are the BAM files (Figure 4). A BAM file (*.bam) is a compressed binary version (BGZF format) of a SAM file that is used to represent aligned sequences. SAM is a TAB-delimited text format consisting of a header section and an alignment section. Header lines start with '@', while alignment lines do not. Each

alignment line has 11 mandatory fields for essential alignment information such as the mapping position, and a variable number of optional fields for flexible or aligner specific information:

SAM Format Description

1. **QNAME**: Query template (read) name.In a SAM file, a read may occupy multiple alignment lines, when its alignment is chimeric or when multiple mappings are given.
2. **FLAG**: SAM flags summarize many properties of reads, represented by flag bits, into a single number:
3. Read is paired.
4. Read is mapped in a proper pair.
5. Read is unmapped.
6. Mate is unmapped.
7. Read reverse strand.
8. Mate reverse strand.
9. Read is from the first pair.
10. Read is from the second pair.
11. Alignment isn't primary.
12. Read fails platform/vendor quality checks.
13. Read is PCR or optical duplicate.
14. **RNAME**: Reference sequence name. If @SQ header lines are present, RNAME must be present in one of the SQ-SN tag.
15. **POS**: 1-based leftmost mapping position of the first CIGAR operation. The first base in a reference sequence has coordinate 1.
16. **MAPQ**: Mapping quality. It equals −10 log10 Pr{mapping position is wrong}, rounded to the nearest integer. A value 255 indicates that the mapping quality is not available.
17. **CIGAR**: A string describing how the read aligns with the reference. It consists of one or more components. Each component comprises an operator and the number of bases which the operator applies to. Operators are:
18. M: Align match.
19. I: Insertion to the reference.
20. D: Deletion from the reference.
21. N: Skipped region from the reference.
22. S: Soft clipping.
23. H: Hard clipping.
24. P: Padding (silent deletion from padded reference).
25. =: Sequence match
26. X: Sequence mismatch
27. **RNEXT**: Reference sequence name of the primary alignment of the next read in the template. If all segments are mapped to the same reference, the unsigned observed template length equals the number of bases from the leftmost mapped base to the rightmost mapped base.
28. **PNEXT**: a 1-based position of the primary alignment of the next read in the template.
29. **TLEN**: Signed observed template length.
30. **SEQ**: Segment sequence.
31. **QUAL**: ASCII of base QUALity plus 33 (same as the quality string in the Sanger FASTQ format).

In addition to these 11 obligatory fields, optional fields may be included. All optional fields follow the TAG:TYPE:VALUE format where TAG is a two-character string.

For more information about the SAM format, visit the SAM Format Specification Page.

You can check the meaning of a FLAG number using the SAM Flag Translator.

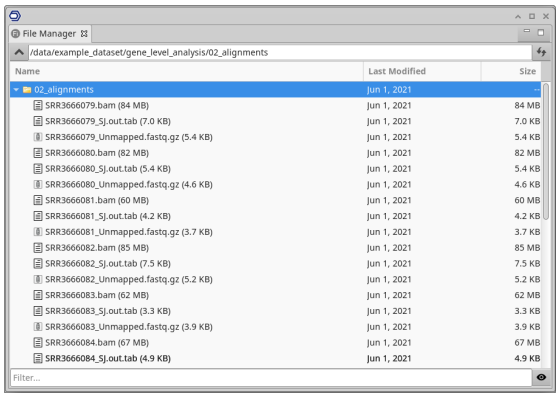


Figure 4: *RNA-Seq Alignment (STAR) Results*

In addition to the BAM files, splice junctions and unmapped reads are generated if the corresponding options were checked.

Splice junctions (Figure 5) are stored in tab-delimited format. The columns have the following meaning:

1. Chromosome.
2. The first base of the intron (1-based).
3. The last base of the intron (1-based).
4. Strand:
5. 0: undefined.
6. 1: +.
7. 2: -.
8. Intron motif:
9. 0: non-canonical.
10. 1: GT/AG.
11. 2: CT/AC.
12. 3: GC/AG.
13. 4: CT/GC.
14. 5: AT/AC.
15. 6: GT/AT.
16. Annotation:
17. 0: unannotated.
18. 1: annotated (only if splice junctions database is used)
19. The number of unique mapping reads crossing the junction.
20. The number of multi-mapping reads crossing the junction.
21. Maximum spliced alignment overhang.

The unmapped files contain the unmapped reads and partially mapped reads (i.e. mapped only one mate of a paired-end read). These reads are stored following the FASTQ specification. Note that if paired-end data were provided, two unmapped files are expected for each sample, one containing upstream unmapped reads and the other containing downstream unmapped reads.

```
                                    ...
   Chromosome  56415   56694   2   2   0   0   43  40
   Chromosome  68864   74367   1   1   0   0   1   37
   Chromosome  69823   75528   1   1   0   0   3   38
   Chromosome  69823   75533   1   1   0   0   10  39
   Chromosome  69823   75538   1   1   0   0   21  38
   Chromosome  69823   75543   1   1   0   0   24  33
   Chromosome  69823   75548   1   1   0   0   26  28
   Chromosome  69823   75553   1   1   0   0   29  23
Chromosome  75559   1000577 1   1   0   0   3795    25
   Chromosome  88501   1000534 1   1   0   0   110 12
   Chromosome  105942  121404  1   1   0   0   1   13
                                    ...
```

Figure 5: *Splice Junctions File*

In addition, a report and a chart are generated with complementary information. The report shows a summary of the RNA-Seq Alignment results (Figure 5). This page contains information about the reference genome sequences, the input FASTQ files, and a results overview. The last section is divided into four subsections: unique reads, multi-mapping reads, chimeric reads, and unmapped reads.

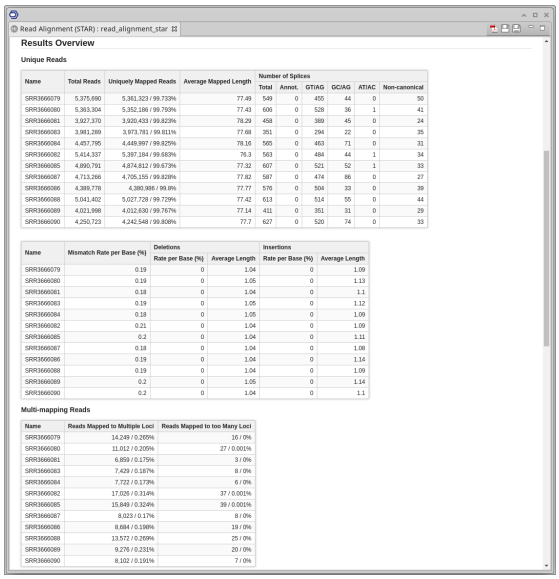


Figure 5: *Summary Report*

The bar chart (Figure 6) shows the number of reads of each input file sorted by different categories, according to how the read was aligned to the reference sequence:

- Unique reads: Reads that have been assigned once to a location of the reference sequence.
- Multi-mapping reads: Reads that have been assign to more than one location of the reference sequence.
- Chimeric reads: Reads that have been aligned to two distinct portions of the reference sequence.
- Unmapped Reads: Reads that have not been assigned to any reference transcript.

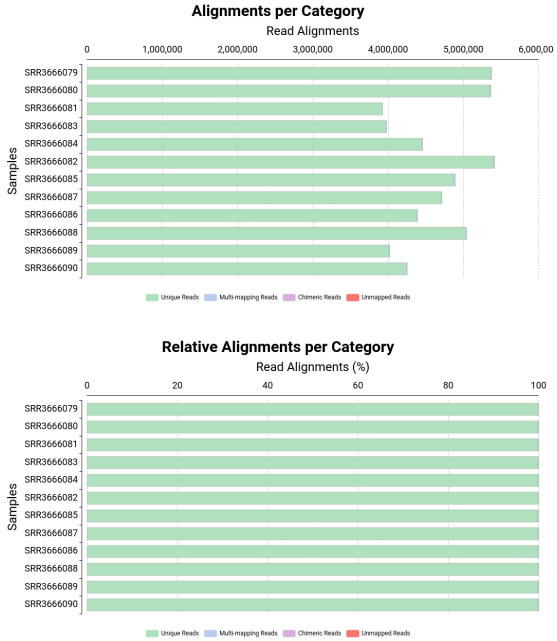


Figure 6: *Alignments per Category Charts*

Finally, the Genome Browser allows you to visualize genomic coordinates (GFF/GTF) in a side-scrolling way. Several tracks can be added to the browser, the currently supported tracks are VCF, DNA Fasta, and BAM. The BAM track (Figure 7) shows the reads of a BAM file and if the sequence track is active, it will also highlight the differences between the read sequence and the sequence track.

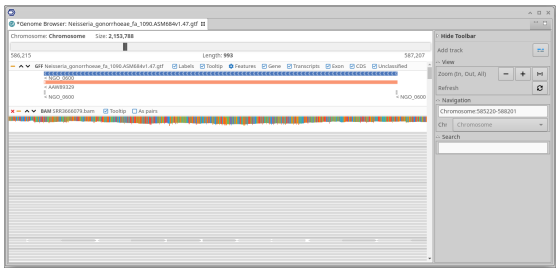


Figure 7: *Genome Browser*

RNA-Seq BWA

INTRODUCTION

The Burrows-Wheeler Alignment algorithm (BWA) is a read alignment package that is based on a backward search with Burrows-Wheeler Transform (BWT), to efficiently align short sequencing reads against a large reference sequence such as the human genome, allowing mismatches and gaps. BWA supports both base space reads, e.g. from Illumina sequencing machines, and color space reads from AB SOLiD machines. The BWA-MEM algorithm is used, which performs local alignment. It may produce multiple primary alignments for different parts of a query sequence.

Please cite BWA as:

Li H. and Durbin R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics (Oxford, England), 25(14), 1754-60.

RUN DNA/RNA-SEQ ALIGNMENT (BWA)

This functionality can be found under **Transcriptomics → RNA-Seq Alignment → BWA**. The wizard allows to select input files and adjust analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

Input

- **Input Reads:** Select the files containing sequencing reads. These files are assumed to be in FASTQ/FASTA format. Both, single and paired-end data are accepted.
- **Paired-end Configuration:** If paired-end reads are provided, a pattern to distinguish upstream files from downstream files is required. The provided patterns are searched in the filenames right before the extension. The beginning of the filenames should be the same for both files of each sample.
- **Upstream Files Pattern**: Establish the pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern**: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is SRR037717_1.fastq and the downstream SRR037717_2.fastq, "_1" should be established as the upstream pattern and "_2" as the downstream pattern.

- **Reference Genome:** Specify a FASTA file with the genome reference sequences. Multiple reference sequences (e.g. chromosomes or scaffolds) are allowed.

It is not recommended to provide masked genome sequences since the algorithm will force those reads that originate in repeats to map (falsely) somewhere else in the genome.

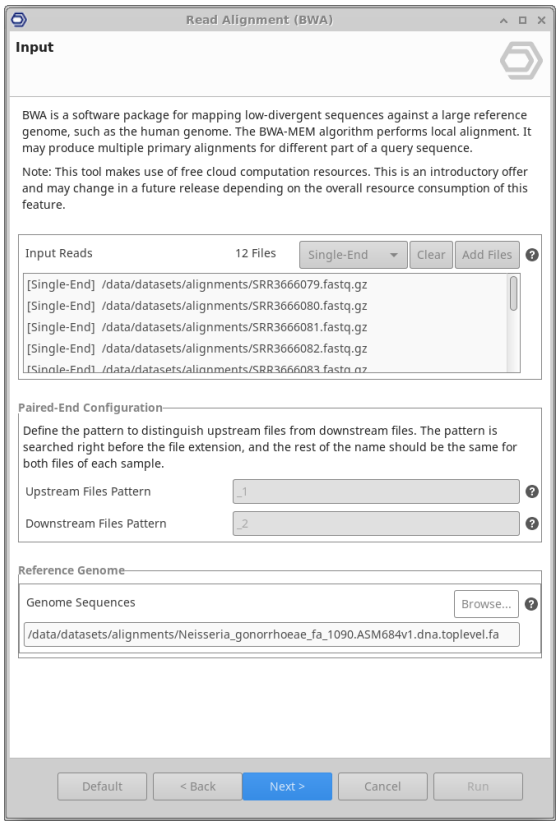


Figure 1: *Input Page*

Algorithm Options

- **Minimum Seed Length:** Matches shorter than this value will be missed. The alignment speed is usually intensive to this value unless it significantly deviates 20.

- **Band Width:** Essentially, baps longer than this value will not be found. Note that the maximum gap length is also affected by the scoring matrix and the hit length, not solely determined by this option.
- **Z-dropoff:** Also known as Off-diagonal X-dropoff. Stop extension when the difference between the best and the current extension score is above |i-j|*A+Z-dropoff, where *i* and *j* are the current positions of the query and reference, respectively, and A is the matching score. Z-dropoff is similar to BLAST's X-dropoff except that it does not penalize gaps in one of the sequences in the alignment. Z-dropoff not only avoids unnecessary extension but also reduces poor alignments inside a long good alignment.
- **Trigger Re-seeding:** Look for internal seeds inside a seed longer than {Minimum Seed Length} * Trigger Re-seeding. This is a key heuristic parameter for tuning the performance. Larger values yield fewer seeds, which leads to faster alignment speed but lower accuracy.
- **Seed Occurrence:** Seed occurrence for the 3rd round seeding.
- **Skip Seeds:** Discard a seed if it has more than this number of occurrences in the genome.
- **Drop Chains:** Drop chains shorter than this fraction of the longest overlapping chain.
- **Discard Chains:** Discard a chain if seeded bases shorter than this value.
- **Mate Rescue Rounds:** Perform at most this number of rounds of mate rescues for each read.
- **Skip Mate Rescue:** Skip the mate rescue procedure.
- **Skip Pairing:** In the paired-end mode, perform SW to rescue missing hits only but do not try to find hits that fit a proper pair. The mate rescue is performed, unless the Skip Mate Rescue option is also in use.

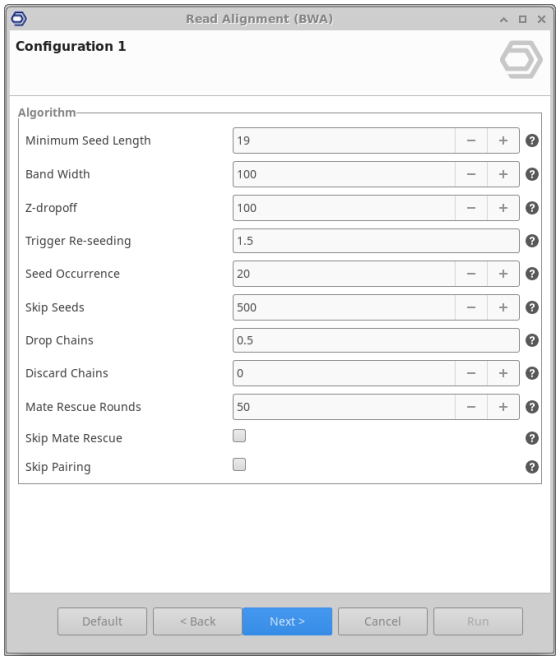


Figure 2: *Configuration Page 1*

Scoring Options

- **Matching Score:** Score for a sequence match.
- **Mismatch Penalty:** Penalty for a mismatch. The sequence error rate is approximately: {.75 * exp[-log(4) * Mismatch Penalty/Matching Score]}.
- **Gap Open Penalty for Deletions:** Gap open penalty for deletions.
- **Gap Open Penalty for Insertions:** Gap open penalty for insertions.
- **Gap Extension Penalty for Deletions:** Gap extension penalty for deletions. A gap of size k cost '{-O} + {-'Gap Extension Penalty'}*k'.
- **Gap Extension Penalty for Insertions:** Gap extension penalty for insertions. A gap of size k cost '{-O} + {-'Gap Extension Penalty'}*k'.
- **5'-end Clipping Penalty:** Penalty for 5'-end clipping. When performing SW extension, BWA-MEM keeps track of the best score reaching the end of the query. If this score is larger than the best SW score minus the clipping penalty, clipping will not be applied. Note that in this case, the SAM AS tag reports the best SW score; the clipping penalty is not deducted.
- **3'-end Clipping Penalty:** Penalty for 3'-end clipping.
- **Unpaired Read Penalty:** Penalty for an unpaired read pair.

Output Options

- **Minimum Score:** Minimum score to output.
- **Mark Split Alignments as Primary:** For split alignment, take the alignment with the smallest coordinate as primary.
- **Not Modify mapQ of Supp. Alignments:** Do not modify the mapping quality of supplementary alignments.

- **Output All SE/Unpaired PE Alignments:** Output all alignments for Single-End or unpaired Paired-End reads.
- **Soft Clipping for Supplementary:** Use soft clipping for supplementary alignments.
- **Mark Shorter Split Hits as Secondary:** Mark shorter split hits as secondary.
- **Sort BAM File:** Establish how output BAM files should be sorted.
- **Add Read Group Information:** Include the 'Read Group' header (@RG) in output BAM files. This information may be required for downstream analysis of third-party tools. If this option is checked, the following read group tags will be included for each sample:
- Identifier (ID), automatically generated.
- The name of the sample (SM), inferred from file names.
- Sequencing Platform (PL), provided by the user.
- **Sequencing Platform:** Choose the sequencing platform which was used to obtain the input data. Consider that if this option is provided, all output BAMs will be tagged with the same platform.

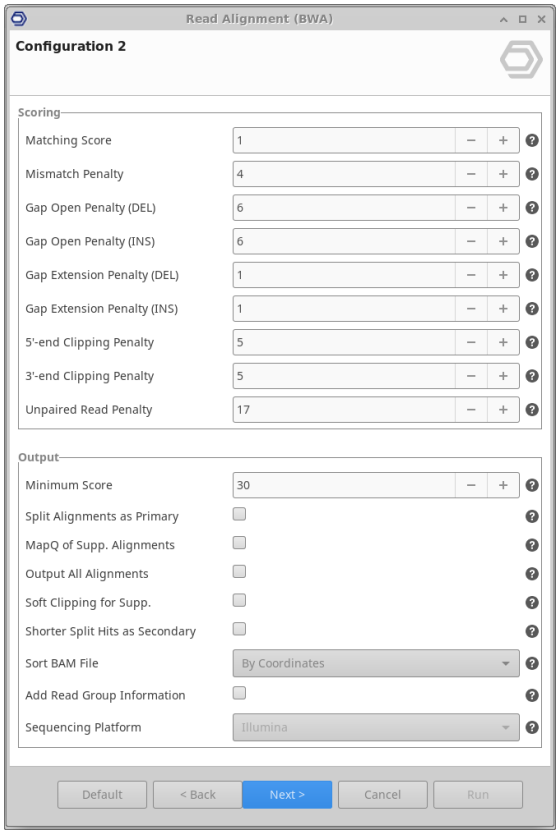


Figure 3: *Configuration Page 2*

Output

- **Alignment Files:** Select a destination folder to save output BAM files.

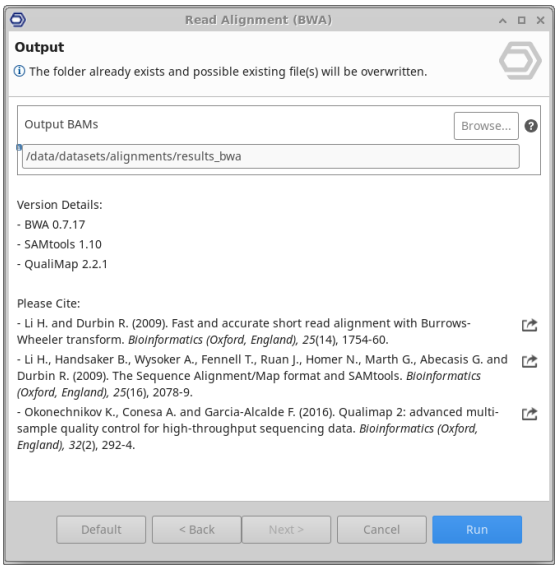


Figure 4: *Output Page*

RESULTS

The main outputs are the BAM files. A BAM file (*.bam) is a compressed binary version (BGZF format) of a SAM file that is used to represent aligned sequences. SAM is a TAB-delimited text format consisting of a header section and an alignment section. Header lines start with '@', while alignment lines do not. Each alignment line has

11 mandatory fields for essential alignment information such as the mapping position, and a variable number of optional fields for flexible or aligner-specific information.

SAM Format Description

1. **QNAME**: Query template (read) name.In a SAM file, a read may occupy multiple alignment lines, when its alignment is chimeric or when multiple mappings are given.
2. **FLAG**: SAM flags summarize many properties of reads, represented by flag bits, into a single number:
3. Read is paired.
4. Read is mapped in a proper pair.
5. Read is unmapped.
6. Mate is unmapped.
7. Read reverse strand.
8. Mate reverse strand.
9. Read is from the first pair.
10. Read is from the second pair.
11. Alignment isn't primary.
12. Read fails platform/vendor quality checks.
13. Read is PCR or optical duplicate.
14. **RNAME**: Reference sequence name. If @SQ header lines are present, RNAME must be present in one of the SQ-SN tag.
15. **POS**: 1-based leftmost mapping position of the first CIGAR operation. The first base in a reference sequence has coordinate 1.
16. **MAPQ**: Mapping quality. It equals −10 log10 Pr{mapping position is wrong}, rounded to the nearest integer. A value 255 indicates that the mapping quality is not available.
17. **CIGAR**: A string describing how the read aligns with the reference. It consists of one or more components. Each component comprises an operator and the number of bases which the operator applies to. Operators are:
18. M: Align match.
19. I: Insertion to the reference.
20. D: Deletion from the reference.
21. N: Skipped region from the reference.
22. S: Soft clipping.
23. H: Hard clipping.
24. P: Padding (silent deletion from padded reference).
25. =: Sequence match
26. X: Sequence mismatch
27. **RNEXT**: Reference sequence name of the primary alignment of the next read in the template. If all segments are mapped to the same reference, the unsigned observed template length equals the number of bases from the leftmost mapped base to the rightmost mapped base.
28. **PNEXT**: a 1-based position of the primary alignment of the next read in the template.
29. **TLEN**: Signed observed template length.
30. **SEQ**: Segment sequence.
31. **QUAL**: ASCII of base QUALity plus 33 (same as the quality string in the Sanger FASTQ format).

In addition to these 11 obligatory fields, optional fields may be included. All optional fields follow the TAG:TYPE:VALUE format where TAG is a two-character string.

For more information about the SAM format, visit the SAM Format Specification Page.

You can check the meaning of a FLAG number using the SAM Flag Translator.

In addition, a report and two charts are generated with complementary information. The report (Figure 5) shows a summary of the DNA-Seq Alignment results. This page contains information about the reference genome sequences, the input FASTQ files, and a results overview. The last section is divided into several subsections: globals, paired information, ACTG content, coverage, mapping quality, insert size, mismatches, and indels.

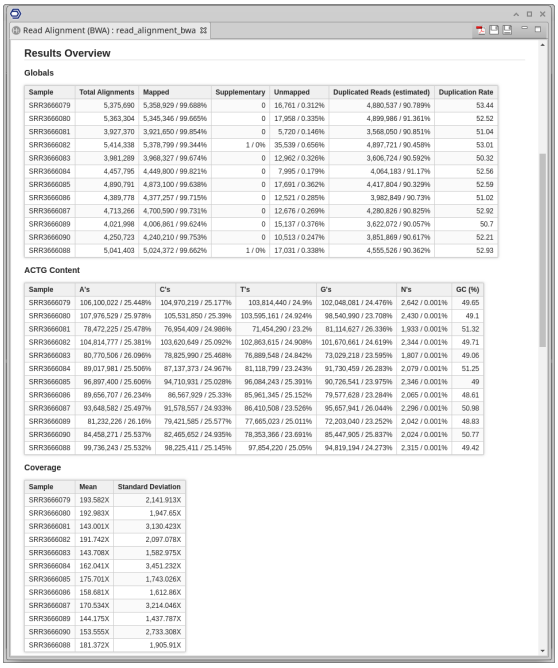


Figure 5: *Summary Report*

The bar charts (Figure 6) show the number of mapped and unmapped reads of each input file.

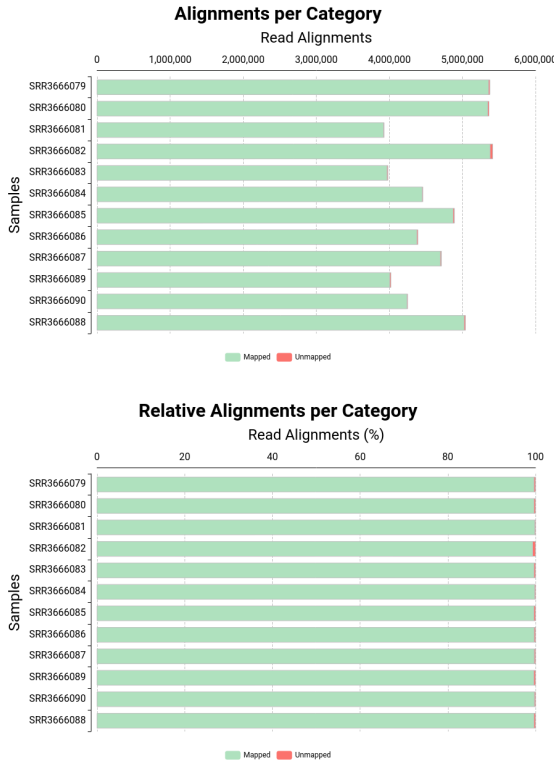


Figure 6: *Alignments per Category Charts*

Finally, the Genome Browser allows you to visualize genomic coordinates (GFF/GTF) in a side-scrolling way. Several tracks can be added to the browser, the currently supported tracks are VCF, DNA Fasta, and BAM. The BAM track (Figure 7) shows the reads of a BAM file and if the sequence track is active, it will also highlight the differences between the read sequence and the sequence track.

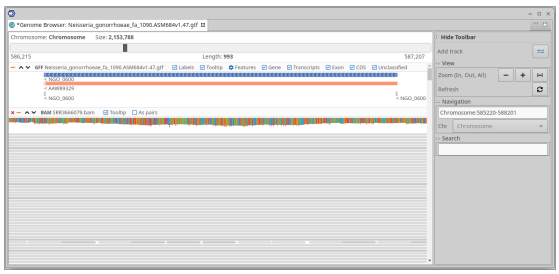


Figure 7: *Genome Browser*

4.4.8 BAM File Quality Control

Introduction

This tool allows the evaluation of alignment files of RNA-Seq datasets comprehensively. It makes use of the R package RSeQC which provides a number of modules that quickly inspect sequence quality, nucleotide composition bias, PCR bias, and GC bias. The RNA-Seq specific modules allow to evaluate:

- sequencing saturation
- mapped reads distribution
- coverage uniformity
- strand specificity
- transcript level RNA integrity
- and more.

Please cite RSeQC and Samtools as:

- Wang L., Wang S. and Li W. (2012). RSeQC: quality control of RNA-seq experiments. Bioinformatics (Oxford, England), 28(16), 2184-5.
- Li H., Handsaker B., Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G. and Durbin R. (2009). The Sequence Alignment/Map format and SAMtools. Bioinformatics (Oxford, England), 25(16), 2078-9.

Run Bam Quality Control for RNA-Seq alignment data

It can be found in the Transcriptomics Module of OmicsBox under **Bam File Quality Control**. The wizard allows providing various input BAM files, an optional BED, GFF, or GTF reference, and several parameters that depend on the input data (Figure 1 and Figure 2)

INPUT

Aligned RNA-Seq short-reads in BAM format (single or paired-end) can be provided as input.

CONFIGURATION

- **Gene Models**
  A BED file with gene models can be provided, that has to match the version of the reference genome during the previous mapping step. Chromosome names have to match and the BED file is expected to have 12 tab-separated columns.
  Providing a BED file is optional and recommended because many statistics and plots are not available without this.
- **Minimum Mapping Quality**
  Establish the minimum mapping quality (Phred-scaled) for an alignment to be considered "uniquely mapped".
- **Read Alignment Length**
  Set this to the original read length. For example, all these cigar strings ("101M", "68M140N33M", "53M1D48M") suggest the read alignment length is 101.
- **Read Sample Rate**
  The number of aligned reads will be used to calculate the mismatch and deletion profiles. The default value is 1000000.
- **Minimum Intron Length**
  Minimum intron length in base pairs. The default value is 50.

- **Min Reads for Junction Calls**
  The minimum number of supporting reads necessary to call a junction. The default value is 1.

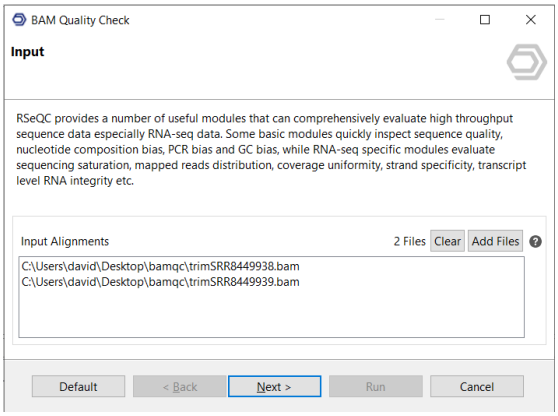


Figure 1: *Input Page*

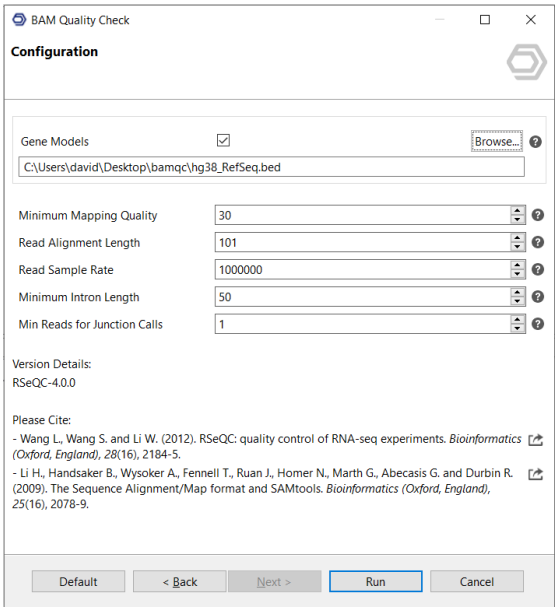


Figure 2: *Configuration Page*

Results

- **Table** with the main information about all the analyzed samples (Figure 4).
- **Report** with specific information about each sample (Figure 3).
- **Charts** can be created from the main table side panel.

CHARTS

In the side panel of the table (Figure 4) there are different action buttons with different chart categories (some examples below):

- **General Charts**
  These charts are related to general characteristics.
- **Annotation Based Charts**
  These charts are related to the distance of Full Splice Match (FSM) and Incomplete Splice Match (ISM) transcripts to annotated Transcription Start Sites (TSS) and Transcription Termination Sites (TTS).

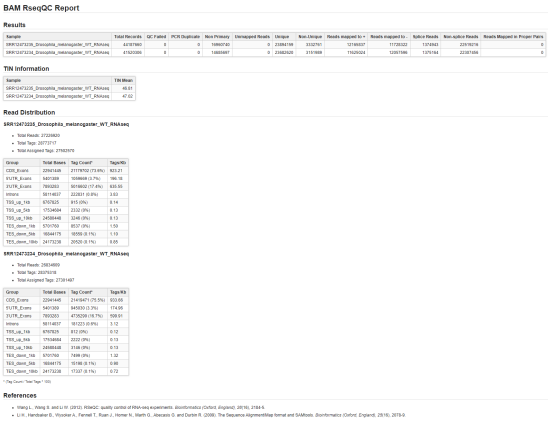


Figure 3: *Report Page*

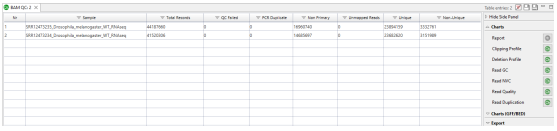


Figure 4: *Results Table*

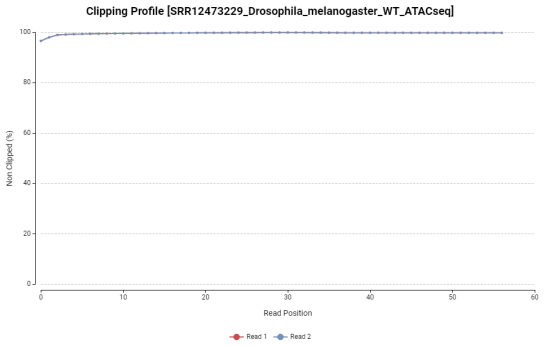


Figure 5: *Clipping Profile Chart*

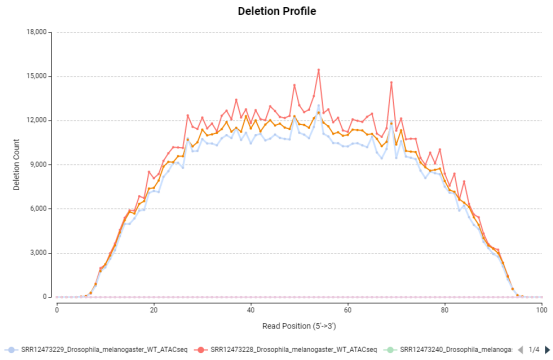


Figure 6: *Deletion profile Chart*

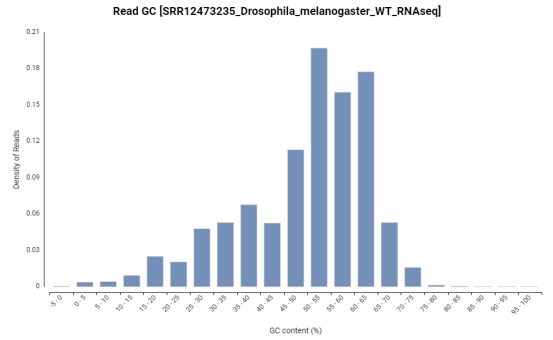


Figure 7: *Read GC Content Distribution Chart*

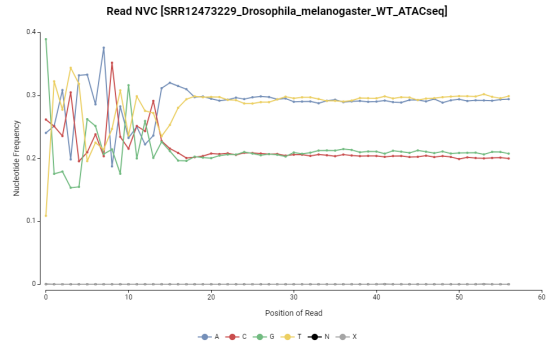


Figure 8: *Read NVC Distribution*

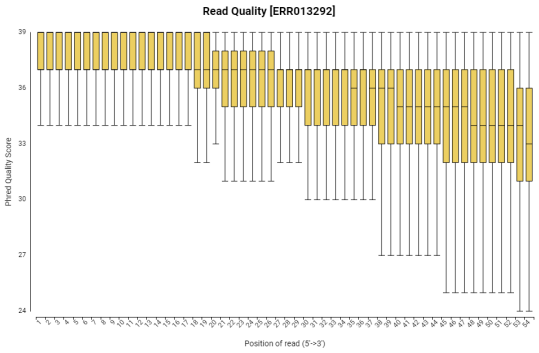


Figure 9: *Read Quality Chart*

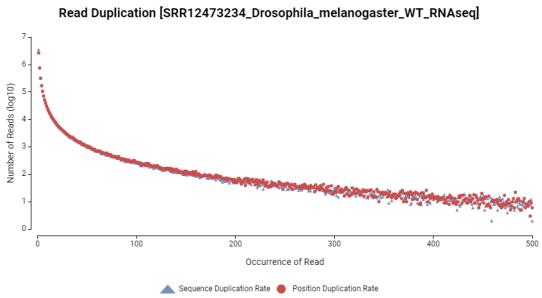


Figure 10: *Read Duplication Rate*

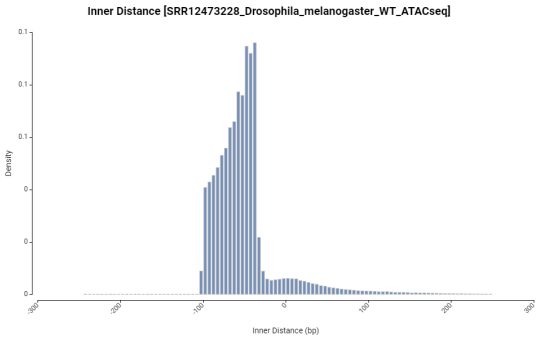


Figure 11: *Inner Distance Chart*

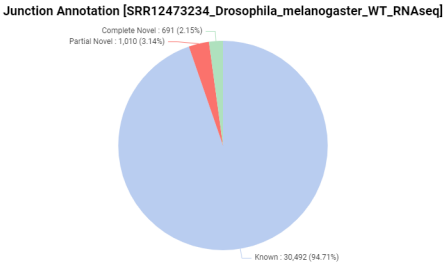


Figure 12: *Junction Annotation Pie Chart*

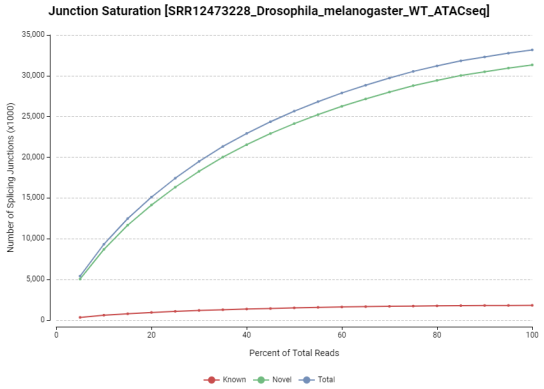


Figure 13: *Junction Saturation Chart*

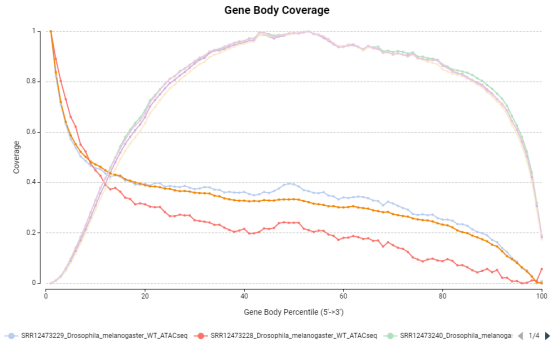


Figure 14: *Gene Body Coverage Distribution*

4.4.9 Create Count Table

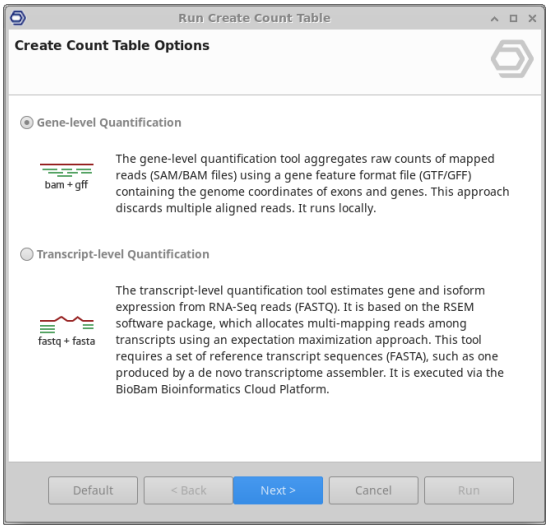
Create Count Table

One of the most common applications of RNA-seq is to estimate gene and transcript expression, and it is a required step before differential expression analysis. It starts with the alignment or mapping of reads and there are two possible alternatives: mapping to the genome when a reference sequence is available (RNA-Seq Alignment) or mapping to the transcriptome (RNA-Seq de novo Assembly). After the mapping, the read quantification is performed. The simplest approach is toaggregate raw counts of mapped reads taking into account gene coordinates. Other cases require more sophisticated algorithms, capable of allocating multi-mapping reads among similar transcripts (isoforms).

This functionality can be found under **transcriptomics → RNA-Seq Read Quantification.**

Two strategies are available:

- **Gene-level Quantification:** The gene-level quantification tool aggregates raw counts of mapped reads (SAM/BAM files) using a gene feature format file (GTF/GFF) containing the genome coordinates of exons and genes. This approach discards multiple aligned reads. It runs locally.
- **Transcript-level Quantification:** The transcript-level quantification tool estimates gene and isoform expression from RNA-Seq reads (FASTQ). It is based on the RSEM software package, which allocates multi-mapping reads among transcripts using an expectation-maximization approach. This tool requires a set of reference transcript sequences (FASTA), such as one produced by a de novo transcriptome assembler. It is executed via the BioBam Bioinformatics Cloud Platform.



Gene-level Quantification

INTRODUCTION

The "Create Count Table" tool is designed to estimate gene expression from RNA-sequencing experiments. This tool expects files with aligned sequencing reads in SAM/BAM format and a GTF/GFF file with coordinates of genomic features. It counts how many reads map to each feature of interest (e.g. genes, exons...). A Count Table is obtained that can be used to perform a differential expression analysis within OmicsBox.

Only reads mapping unambiguously to a single genomic feature are considered. On the other hand, reads aligned to more than one position or overlapping with more than one feature are discarded. This is convenient because if there are two or more genes that overlap or have some sequence similarity but they have different expression levels, counting common reads for all of them could provide inaccurate results. If paired-end data is provided fragments (read-pairs) instead of single reads are counted.

This module is based on the popular **HTSeq package**. Please cite HTSeq as:
Anders S, Pyl PT and Huber W (2014). "HTSeq — A Python framework to work with high-throughput sequencing data." Bioinformatics, 31(2), 166-9.

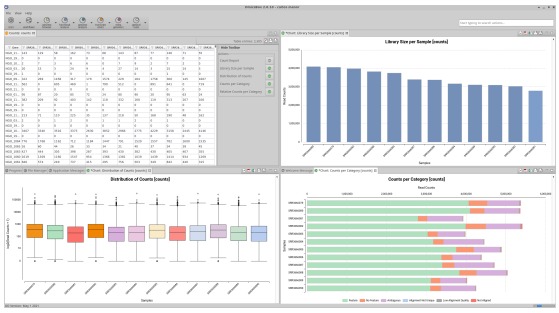


Figure 1:*Create Count Table Interface*

RUN CREATE COUNT TABLE

This functionality can be found under **transcriptomics → RNA-Seq Read Quantification → Gene Level Quantification**. The wizard allows to select input files and adjust analysis parameters (Figure 2 and Figure 3).

Input

- **Alignment Files:**Select files containing the sequencing alignment data. It must be in the "Sequence Alignment/Map" format (SAM) or in its compressed format (BAM).
- **Annotation File:**Select the file containing the list of genomic features in GFF/GTF format. GFF objects from OmicsBox are accepted too.

The GFF/GTF must belong to the same genome as the one used for the alignments.

Figure 2: *Input Page*

Configuration

- **Quantification Level:** Choose the feature type (3rd column in GFF/GTF file, "Type" column in GFF object) for which expression will be quantified (e.g. gene, exon...). Features coordinates (range of positions) will be extracted from annotation using the provided value and all features of other types are ignored.
- **Name/Group By:**Specify the attribute type (9th column in GFF/GTF file, "Attr" columns in GFF object) to be used as feature ID. The feature ID is used to identify counts in the output Count Table. Attribute types tagged with "*" (e.g. *gene_name) are not present in all features of the selected type and only those containing it will be extracted. Several GFF lines with the same feature ID will be considered as parts of the same feature. Figure 4 illustrates how "Quantification Level" and "Name/Group By" parameters work.
- **Strand Specificity:**Indicate how the strand is taken into account.
- Non-Strand Specific: A read is considered overlapping with a feature regardless of the strand in which the read has been mapped.
- Strand Specific Forward: For single-end reads, the read has to be mapped to the same strand as the feature to be counted. For paired-end reads, the first read on the pair must be mapped to the same strand as the feature and the second read on the opposite strand.
- Strand Specific Reverse: For single-end reads, the read has to be mapped to the opposite strand of the feature to be counted. For paired-end reads, the first read on the pair must be mapped to the opposite strand of the feature and the second read on the same strand.
- **Overlap Mode:**Modes to handle reads overlapping more than one feature. Consider that for each position in the read, a set of all features overlapping is defined. If the resulting set for a read (or read pair) contains precisely one feature, the read is counted for this feature. If it contains more than one feature, the read is counted as "ambiguous" (and not counted for any features), and if it is empty, the read is counted as "no feature". The three overlap modes join these sets as follows (Figure 5 illustrates the effect of these three modes):
- Union: The union of all sets.
- Intersection Strict: The intersection of all sets.
- Intersection Non-Empty: The intersection of all non-empty sets.

- **Minimum Mapping Quality:**Set a filter to discard all reads with alignment quality (MAPQ) lower than the given minimum value.

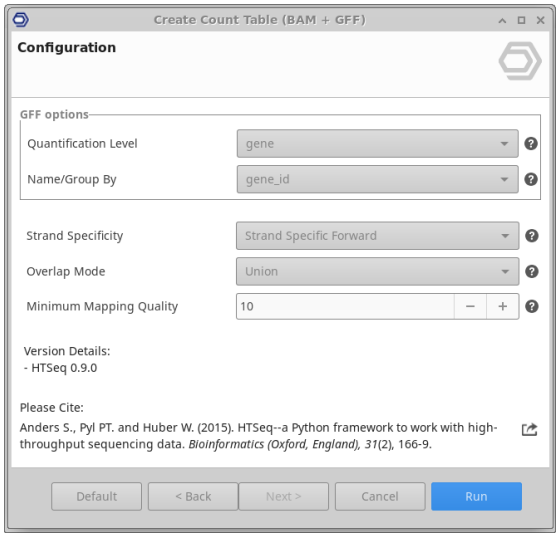


Figure 3: *Configuration Page*

SEQNAME	SOURCE	FEATURE	START	END	SCORE	STRA
chrom_1	RefSeq	gene	200	3150	·	+
chrom_1	RefSeq	mRNA	200	3150	·	+
chrom_1	RefSeq	exon	200	1520	·	+
chrom_1	RefSeq	exon	1900	3150	·	+

QUANTIFICATION LEVEL	NAME/ GROUP BY	FEATURE ID (IN COUNT TABLE)
gene	locus_tag	**gene_one**
exon	gene_id	**Gene_1**
exon	ID	**exon1 exon2**

Figure 4:*Example of a simple GFF and usage of "Quantification Level" and "Name/Group By" parameters*

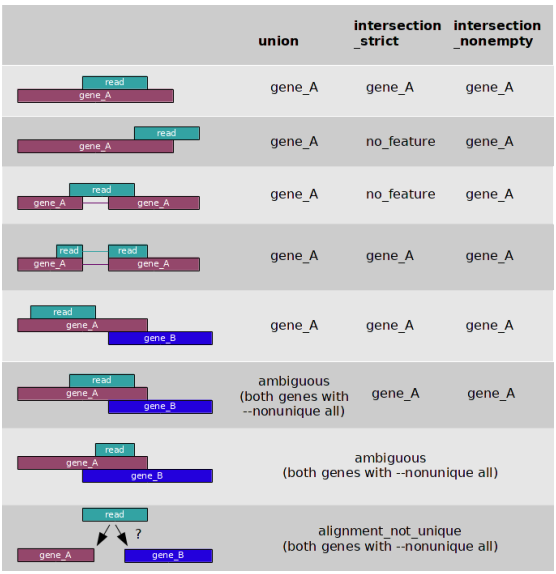


Figure 5: *Scheme of overlap modes from HTSeq manual*

RESULTS

Once the analysis has been finished, a new tab containing the resulting Count Table is opened (Figure 6). Rows correspond to genomic features and columns to samples (one by analyzed file). Counts represent the total number of reads aligned to each genomic feature. Results can be saved as a Count Table object.

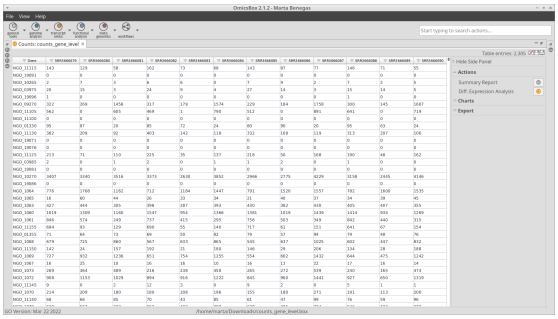


Figure 6: *Count Table*

Furthermore, a result page will show a summary of the "Create Count Table" results (Figure 7). On this page information about the extraction of genomic features from GFF, alignment files, and obtained results are provided. The result summary can be generated via **Side Panel → Actions → Summary Report**and it can be exported as pdf.

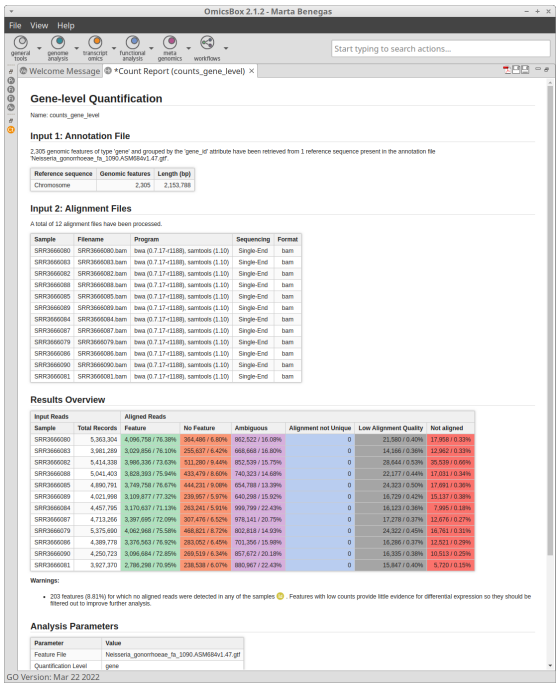


Figure 7:*Result Summary*

Actions

All the actions available for this type of object are on the **Side Panel → Actions**.

Summary Report

It generates the Summary Report explained above (Figure 7).

Rename Features

This option allows modifying the sequence IDs in the Feature column using different methods:

- **Add**: Add a prefix or sufix to all IDs in the table.
- **Replace**: Replace specific text within the IDs. The text to be replaced must be defined in the Find parameter using a regular expression (regex).
- **Mapping**: Use a mapping file to rename features. The mapping file must be a tab-separated text file with two columns: the first column contains the original feature IDs from the dataset, and the second column contains the new feature names. If duplicate IDs occur during renaming, you can define how they are handled:
  - Sum Rows: Combine counts for all matching features.
  - First Row: Retain only the counts of the first occurrence.

Diff. Expression Analysis

This feature performs a Differential Expression Analysis as explained here.

Charts

Different statistical charts of the obtained results can be generated. These provide additional information about the process of quantifying expression, as well as a quality assessment of the resulting counts. All these charts can be found under the **Side Panel → Charts** of the Count Table viewer.

Library Size per Sample

Bar chart showing the number of read counts aligned to genomic features contained in each sample (Figure 8).

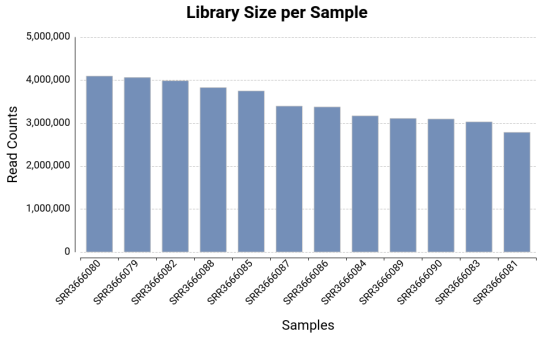


Figure 8: *Library Size per Sample*

Distribution of Counts

Box plot that allows seeing how counts are distributed within each sample for all the features (Figure 9). Features with 0 counts in all samples will be discarded for this chart. The binary logarithm of raw counts is represented.

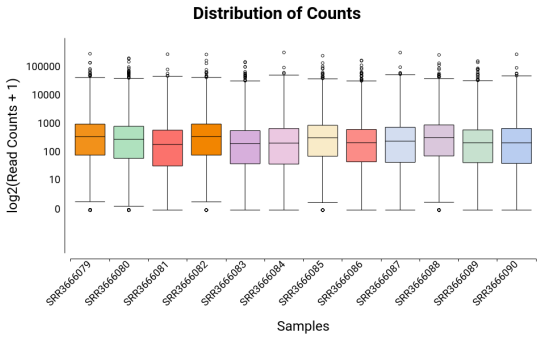


Figure 9: *Distribution of Counts*

Counts per Category

Bar chart showing the number of reads of each input file sorted by different categories (Figure 10). This chart and the next one are only available for count tables created by the "Create Count Table" tool within OmicsBox.

- Feature: The sum of all reads that have been assigned to any features.
- No Feature: Reads which could not be assigned to any feature (the resulting set for the read is empty as mentioned above).
- Ambiguous: Reads which have been assigned to more than one feature (the resulting set for the read has more than one feature). These reads are not counted for any feature.
- Alignment Not Unique: Reads with more than one reported alignment. These reads are identified from the NH optional SAM field tag. If the program that was used to obtain alignments does not set this field, multiple aligned reads will be counted multiple times.
- Low Alignment Quality: Reads which were skipped due to the "Minimum Mapping Quality" filter set on the main wizard page.
- Not Aligned: Reads in the SAM/BAM file without alignment.

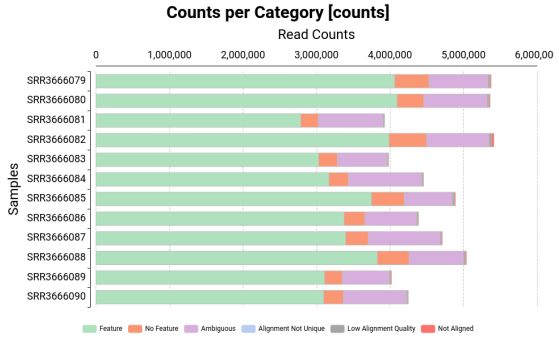


Figure 10: *Counts per Category*

Counts per Category (%)

The same chart as explained above but in percentages (Figure 11).

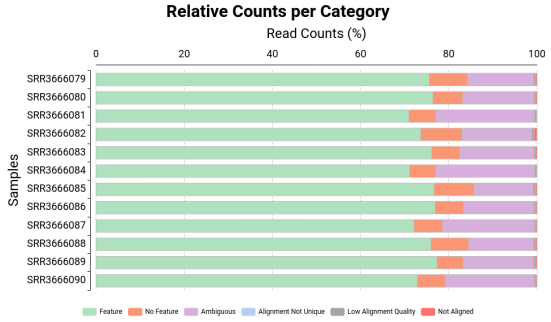


Figure 11: *Counts per Category (%)*

PCA Plot

This feature performs a Principal Component Analysis and generates a 2D (Figure 12) or 3D (Figure 13) with the two and three first Principal Components, respectively. This chart helps to identify which samples are similar to each other in terms of gene expression. Ideally, samples belonging to the same condition should appear closer in the plot.

PCA Plot in 3 Dimensions is only available for datasets with 3 or more samples.

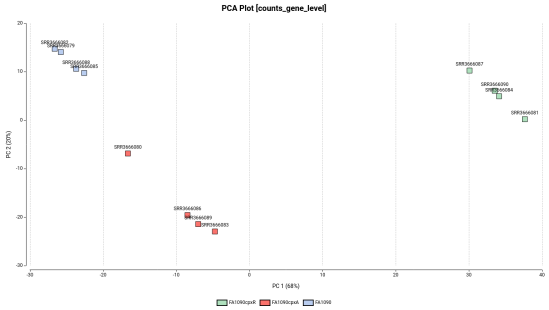


Figure 12: *2 Dimensions PCA plot.*

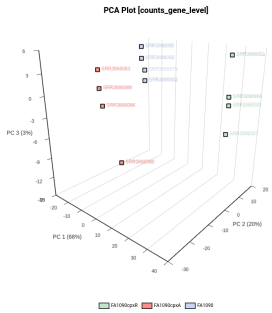


Figure 13: *3 Dimensions PCA plot.*

Context Menu

Besides the generic context menu options, the available actions for this object depend on whether one or multiple rows are selected.

With one row selected:

- **Remove Sample**: Remove a specific sample from the dataset by right-clicking on the corresponding column in the selected row.
- **Rename Sample**: Change the name of a sample by right-clicking on the corresponding column in the selected row.

With multiple rows selected:

- **Extract Selection to New Tab**: Extract the data from the selected rows and open it in a new tab.

Transcript-level Quantification

INTRODUCTION

The transcript-level quantification tool is designed for estimating gene and isoform expression levels from RNA-Seq data. It expects the sequencing reads in FASTQ format (so a prior alignment is not necessary), and it supports both single-end and paired-end data. In addition, a set of transcript sequences in FASTA format is required, such as one produced by a de novo transcriptome assembler. Therefore it lacks the requirement of a reference genome. A Count Table is obtained and it can be used to perform a differential expression analysis within OmicsBox.

The application is based on **RSEM**, a software package that quantifies expression from transcriptome data. This program handles both the alignment of reads against the reference transcript sequences and the calculation for relative abundances. RSEM uses the Bowtie2 aligner to align reads, with parameters specifically chosen for RNA-Seq quantification. Since RNA-Seq reads do not always map uniquely to a single gene or isoform, this method is able to allocate multi-mapping reads among transcripts using an expectation-maximization approach.

This feature uses RSEM and Bowtie2. Please cite RSEM and Bowtie2 as:

- Li B and Dewey CN (2011). "RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome." BMC Bioinformatics, 12:323
- Langmead B, Salzberg S (2012). "Fast gapped-read alignment with Bowtie 2." Nature Methods, 9:357-359

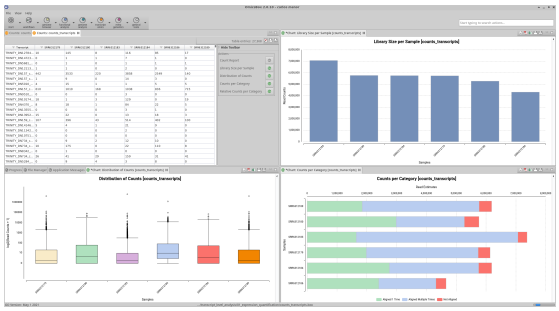


Figure 1:*Create Count Table Interface*

RUN CREATE COUNT TABLE

This functionality can be found under **trasncriptomics → RNA-Seq Read Quantification → Transcript-level Quantification** option. The wizard allows to provide input files and adjust analysis parameters (Figure 2, Figure 3, and Figure 4).

Input Data

- **Sequencing Data:** Choose the type of data to be preprocessed: single-end or paired-end reads. Note that if paired-end is selected, two files per sample are required.
- **Input Reads:**Provide the files containing sequencing reads. These files are assumed to be in FASTQ format.
- **Paired-end configuration:**In the case of paired-end reads, the pattern to distinguish upstream files from downstream files is required. The provided patterns are searched right before the extension, and the start of the name should be the same for both files of each sample.
- Upstream Files Pattern: Establish the pattern to recognize upstream FASTQ files.
- Downstream Files Pattern: Establish the pattern to recognize downstream FASTQ files.

For example, if the upstream file is named SRR037717_1.fastq and the downstream one SRR037717_2.fastq, you should establish "_1" as the upstream pattern and "_2" as the downstream pattern.

- **Transcript References:**This tool works with a set of transcript sequences instead of a genome, such a file could be obtained from a reference genome database or a *de novo* transcriptome assembler. A FASTA file containing the sequences of reference transcripts should be provided.
- **Gene-level Estimations:** This option allows estimating expression both at gene-level and isoform-level. In this way, the gene's expression estimates are just the sum of its transcripts' expression estimates, and results will be provided separately. Otherwise, the program assumes that each transcript provided as a reference sequence is a separated gene.
- **Transcript to Gene Map File:** Provide a file with the information to map from transcript (isoform) identifiers to gene identifiers. Each line should be of the form: gene id transcript id, with the two columns separated by a tab character.

Transcript to Gene Map File Example

```
TRINITY_DN14992_c1_g1   TRINITY_DN14992_c1_g1_i1
TRINITY_DN14992_c1_g1   TRINITY_DN14992_c1_g1_i2
TRINITY_DN14943_c0_g1   TRINITY_DN14943_c0_g1_i1
TRINITY_DN14948_c0_g1   TRINITY_DN14948_c0_g1_i1
TRINITY_DN14902_c0_g1   TRINITY_DN14902_c0_g1_i1
TRINITY_DN14902_c0_g1   TRINITY_DN14902_c0_g1_i2
TRINITY_DN14921_c0_g1   TRINITY_DN14921_c0_g1_i1
TRINITY_DN14921_c0_g1   TRINITY_DN14921_c0_g1_i2
TRINITY_DN14987_c0_g1   TRINITY_DN14987_c0_g1_i1
TRINITY_DN14965_c0_g1   TRINITY_DN14965_c0_g1_i1
```

```
TRINITY_DN14965_c0_g2   TRINITY_DN14965_c0_g2_i1
TRINITY_DN14965_c0_g2   TRINITY_DN14965_c0_g2_i2
```

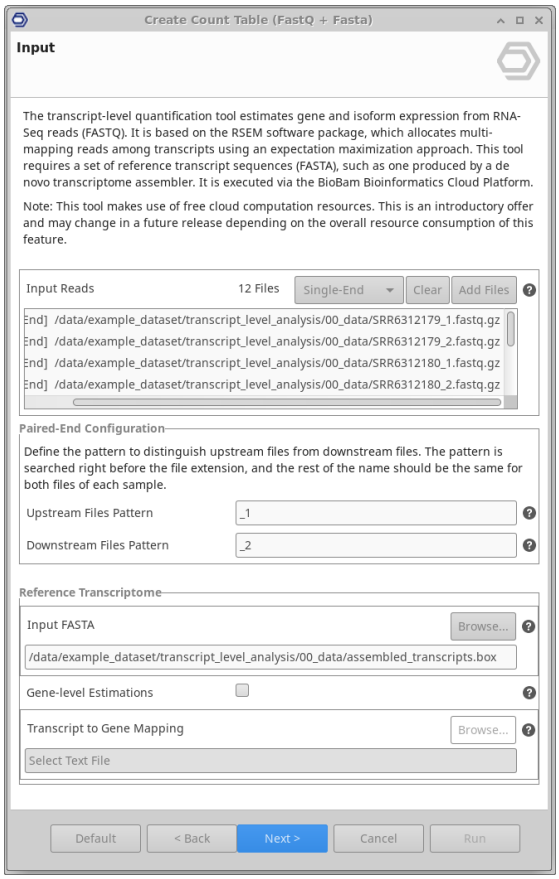


Figure 2:*Input Page*

Advanced Configuration

- **Estimate RSPD:**This option allows to estimate a read start position distribution (RSPD), which increases the accuracy of expression estimates. Highly recommended if the protocol produces read position distributions that are highly 5' or 3' biased. Otherwise, the program will use a uniform RSPD.
- **Append Poly(A) Tails:**For poly(A) mRNA analysis, the program will append poly(A) tail sequences to reference transcripts to allow more accurate read alignment.
- **Poly(A) Tails Length:** Establish the length of the poly(A) tails to be added.
- **Strand Specificity:**This option defines the strandedness of the RNA-Seq reads:
- Non-Strand Specific: Refers to non-strand-specific protocols.
- Strand Specific Forward: This means all (upstream) reads are derived from the forward strand.
- Strand Specific Reverse: This means all (upstream) reads are derived from the reverse strand.
- **Provide Fragment Length Distribution:** For single-end samples, the fragment length distribution can be provided via the Fragment Length mean and the Fragment Length Standard Deviation parameters. The specification of an accurate fragment length distribution is important for the accuracy of expression level estimates from single-end data. If this option is not checked, the fragment length distribution will not be taken into consideration.
- **Fragment Length Mean:** Establish the mean of the fragment length distribution, which is assumed to be a Gaussian.

- **Fragment Length Standard Deviation:** Establish the standard deviation of the fragment length distribution, which is assumed to be a Gaussian.

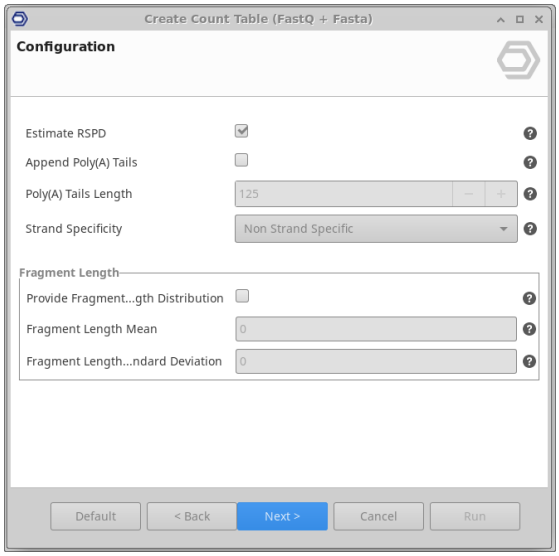


Figure 3:*Configuration Page*

Output Data

- **Alignments.** Decide if alignments files in bam format are saved and select a location to place them. These files can be used for downstream analyses.

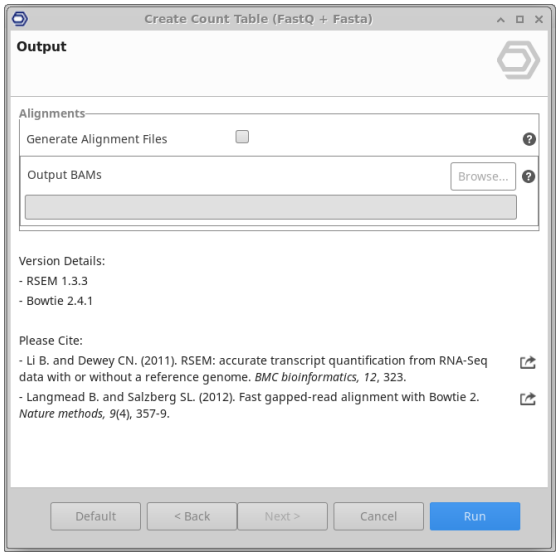


Figure 4:*Output Page*

RESULTS

Once the analysis has been finished results will be returned in two different ways, depending on the option chosen in the "Gene-level Estimations" parameter:

- **Isoform-level and gene-level estimations:**Two Count Tables are returned. One shows the expression level of each transcript or isoform (input sequences) and the other shows the expression level of each gene (Figure 5 and Figure 6). They have an additional column that shows the gene or transcript identifiers (respectively) associated with each record.
- **Transcript-level estimations only:** One Count Table is returned that shows the expression level of each transcript sequence provided as input.

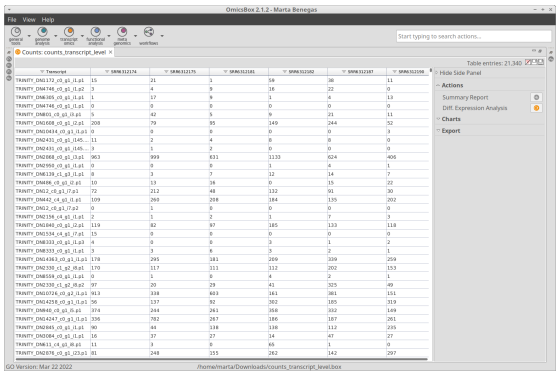


Figure 5:*Count table of isoform-level estimations. The Gene column shows the identifier of the parent gene of each isoform*

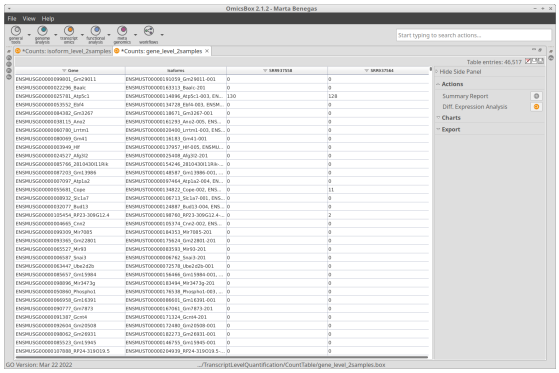


Figure 6:*Count table of gene-level estimations. The Isoform column shows the ids of all isoforms associated with each gene*

Furthermore, a result page will show a summary of the "Create Count Table" results (Figure 7). This page contains information about the reference transcript sequences, input FASTQ files, and obtained results. The results summary can be generated via **Side Panel → Actions → Result Summary**and it can be exported in pdf.

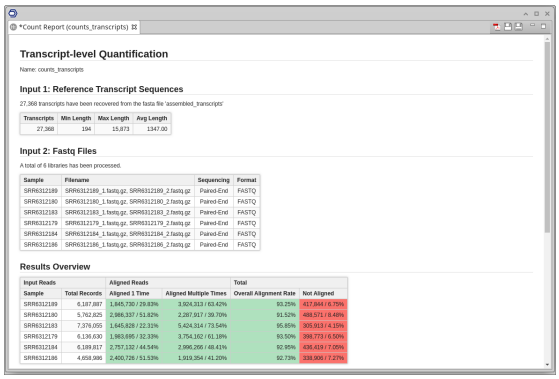


Figure 7: *Summary Report*

Actions

All the actions available for this type of object are on the **Side Panel → Actions**.

Summary Report

It generates the Summary Report explained above (Figure 7).

Rename Features

This option allows modifying the sequence IDs in the Feature column using different methods:

- **Add**: Add a prefix or sufix to all IDs in the table.
- **Replace**: Replace specific text within the IDs. The text to be replaced must be defined in the Find parameter using a regular expression (regex).
- **Mapping**: Use a mapping file to rename features. The mapping file must be a tab-separated text file with two columns: the first column contains the original feature IDs from the dataset, and the second column contains the new feature names. If duplicate IDs occur during renaming, you can define how they are handled:
  - Sum Rows: Combine counts for all matching features.
  - First Row: Retain only the counts of the first occurrence.

Diff. Expression Analysis

This feature performs a Differential Expression Analysis as explained here.

Charts and Statistics

Different statistical charts can be generated from the results. These provide additional information about the process of quantifying expression, as well as a quality assessment of the resulting counts. All these charts can be found under the **Side Panel → Charts** of the Count Table Viewer.

Library Size per Sample

Bar chart showing the number of read counts aligned to genomic features contained in each sample (Figure 8).

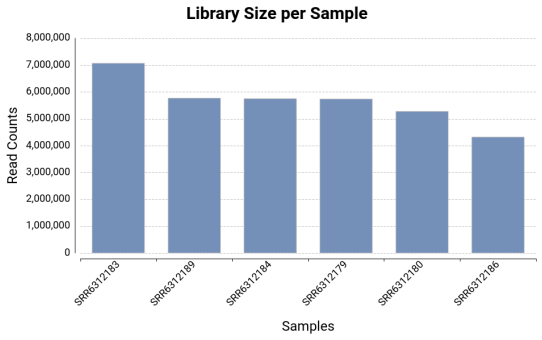


Figure 8: *Library Size per Sample*

Distribution of Counts

Box plot that allows seeing how counts are distributed within each sample for all the transcripts (Figure 9). Features with 0 counts in all samples will be discarded for this chart. The binary logarithm of raw counts is represented.

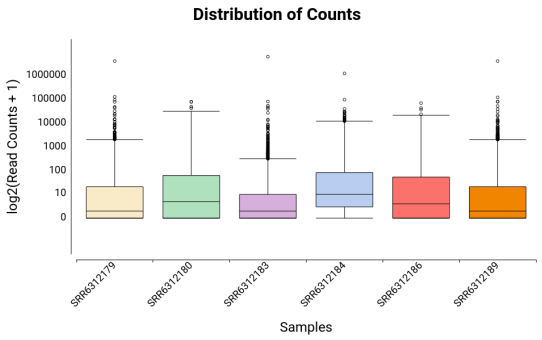


Figure 9: *Distribution of Counts*

Counts per Category

Bar chart showing the number of reads of each input file sorted by different categories (Figure 10). This chart and the next one are only available for count tables created by the "Create Count Table" tool within OmicsBox.

- Aligned Concordantly Exactly 1 Time: Reads that have been assigned once to a reference transcript.
- Aligned Concordantly > Time: Reads that have been assigned to more than one reference transcript.
- Not Aligned: Reads that have not been assigned to any reference transcript.

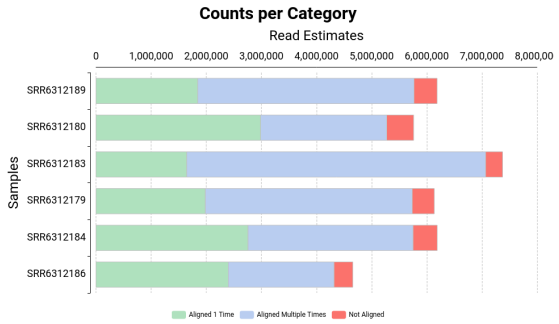


Figure 10: *Counts per Category*

Counts per Category (%)

The same chart is explained above in percentages (Figure 11).

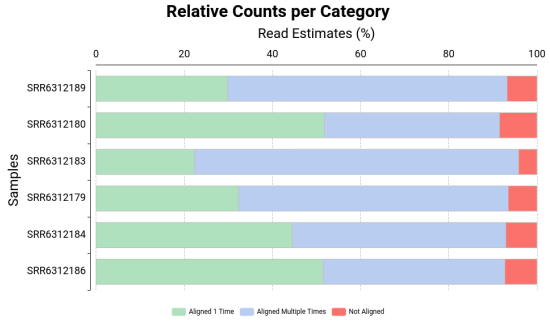


Figure 11: *Relative Counts per Category*

PCA Plot

This feature performs a Principal Component Analysis and generates a 2D (Figure 12) or 3D (Figure 13) with the two and three first Principal Components, respectively. This chart helps to identify which samples are similar to each other in terms of gene expression. Ideally, samples belonging to the same condition should appear closer in the plot.

PCA Plot in 3 Dimensions is only available for datasets with 3 or more samples.

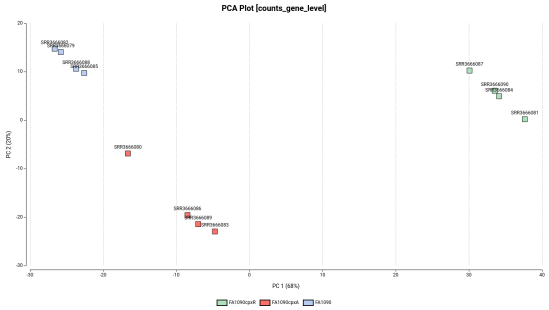


Figure 12: *2 Dimensions PCA plot.*

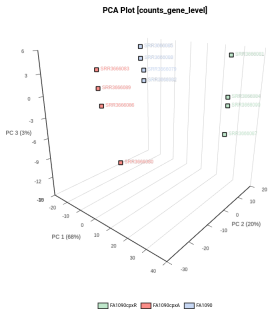


Figure 13: *3 Dimensions PCA plot.*

Context Menu

Besides the generic context menu options, the available actions for this object depend on whether one or multiple rows are selected.

With one row selected:

- **Remove Sample**: Remove a specific sample from the dataset by right-clicking on the corresponding column in the selected row.
- **Rename Sample**: Change the name of a sample by right-clicking on the corresponding column in the selected row.

With multiple rows selected:

- **Extract Selection to New Tab**: Extract the data from the selected rows and open it in a new tab.

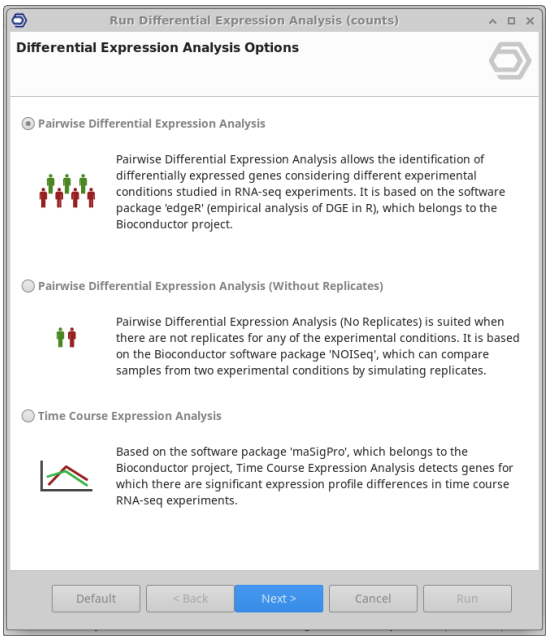
4.4.10 Differential Expression Analysis

Differential Expression Analysis

Differential expression analysis means taking RNA-Seq expression data and performing statistical analysis to discover quantitative changes in gene expression levels between experimental groups. In other words, differential expression analysis consists of the identification of genes (or other types of genomic features, such as transcripts or exons) that are expressed in significantly different quantities in distinct groups of samples. There are several statistical methodologies that allow addressing different experimental designs: biological conditions, diseased vs healthy, different tissues, different development stages, different gender, time series…

Three strategies are available:

- **Pairwise Differential Expression Analysis:** Pairwise differential expression analysis allows the identification of differentially expressed genes considering different experimental conditions studied in RNA-Seq experiments. It is based on the software package "edgeR" (empirical analysis of DGE in R), which belongs to the Bioconductor project.
- **Pairwise Differential Expression Analysis (Without Replicates):** Pairwise differential expression analysis (without replicates) is suited when there are not replicates for any of the experimental conditions. It is based on the Bioconductor software package "NOISeq", which can compare samples from two experimental conditions by simulating replicates.
- **Time Course Expression Analysis:** Based on the software package "maSigPro", which belongs to the Bioconductor project, time-course expression analysis detects genes for which there are significant expression profile differences in time course RNA-Seq experiments.



Pairwise Differential Expression Analysis

INTRODUCTION

This tool is designed to perform differential expression analysis of count data arising from RNA-seq technology. This application, based on the edgeR program, allows the identification of differentially expressed genomic features (e.g. genes) in a pairwise comparison of two different experimental conditions. The software package **edgeR** (empirical analysis of DGE in R), which belongs to the Bioconductor project, implements quantitative statistical methods to evaluate the significance of individual genes between two experimental conditions.

Please cite edgeR as: Robinson MD, McCarthy DJ and Smyth GK (2010). "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data." Bioinformatics, 26, pp. -1.

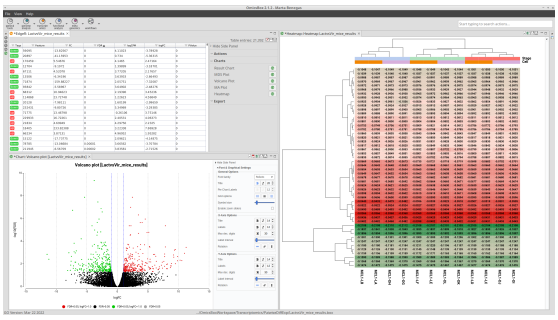


Figure 1: *Differential Expression Analysis Interface*

Expression Data

The pairwise differential expression analysis application expects gene expression levels in the form of a count table. In OmicsBox, count tables can be generated via the **Create Count Table** application.

Count tables can also be imported from a text file. Go to **transcriptomics → Load → Load RNA-Seq Count Table (expression data)** (Figure 2) and select your .txt file containing the count table.

Notes:

- This application only accepts raw counts without any type of normalization.
- Replicates for each experimental condition are necessary.

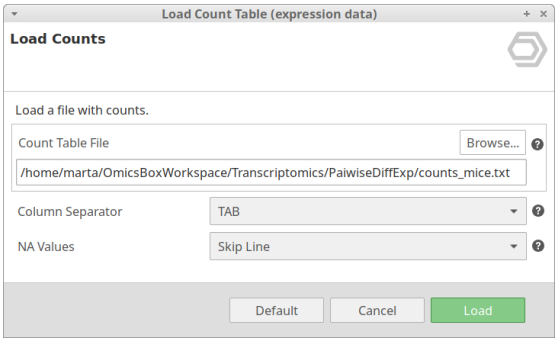


Figure 2: *Load Count Table from File*

RUN PAIRWISE DIFFERENTIAL EXPRESSION ANALYSIS

Go to **transcriptomics → Differential Expression Analysis**. If there's no count table project opened, the first wizard page (Figure 3) will ask to upload either a Count Table Project (.box file) or a Count Table File (.txt, .csv, or .tsv file). On the second wizard page, choose the "Pairwise Differential Analysis" option. If a count table is already loaded in OmicsBox (see above section), this one will be used to perform the analysis. In this case, the first wizard page will be to select the type of differential

expression analysis (Figure 4). In the next pages, it is possible to specify different analysis parameters, which are divided into three different sections: Preprocessing Data (Figure 5), Experimental Design (Figure 6), and Comparison and Test (Figure 7).

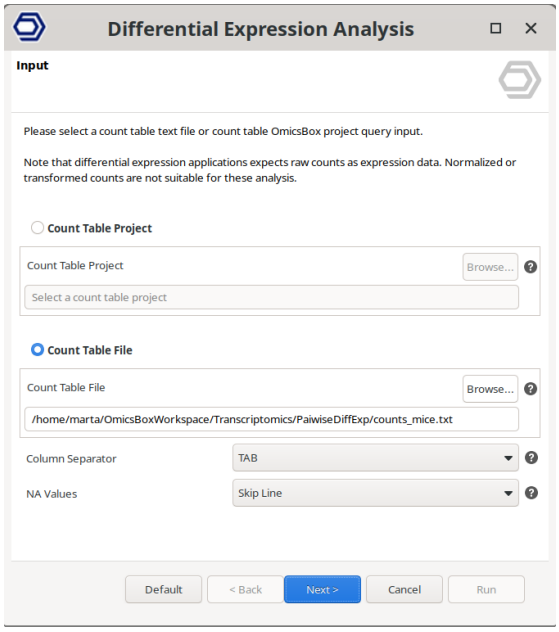


Figure 3: *Input wizard page.*

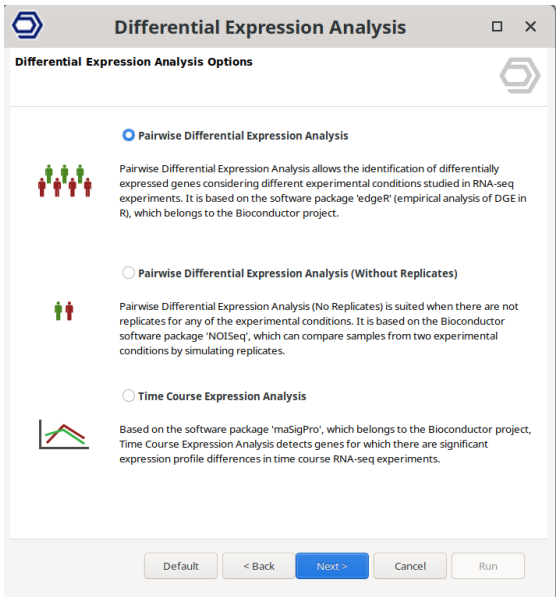


Figure 4: *Differential Expression Analysis Options wizard page.*

Preprocessing Data Page

Filter Genes with low counts. Establish a filter to exclude genes with low counts across libraries, as those genes may interfere with the subsequent statistical approximations:

- **Counts Per Million**: Filtering is performed on a count-per-million (CPM) basis to account for differences in library size between samples (e.g. a CPM of 1 corresponds to a count of 6 in a sample with 6 million reads).
    - **CPM Filter:** Minimum CPM. Genes with a CPM lower than this number will be filtered out. Set to 0 to not apply this filter.
    - **Samples reaching CPM Filter:** Set a minimum number of samples in which the gene's CPM is above the filter level (is expressed). If this value is set to e.g. 5, the gene's CPM has to be greater than the "CPM Filter" value in at least 5 of the samples. The number of samples in the smallest experimental group is usually used. That is, in an experiment that has two replicates for one condition (or group) and three for the other one, a gene should be expressed in at least two samples. Set to 0 to not apply this filter.
- **Automatic Filers:** This function automatically calculates the filtering thresholds to keep genes with worthwhile counts in a minimum number of samples. For more information, please visit EdgeR's documentation in this link.

Calculate normalization factors to scale the raw library sizes.

- **Normalization Method:** Here the normalization takes the form of scaling factors for library sizes that enter into the statistical model. These correctional factors are used to compute the effective library sizes. For further details please refer to the edgeR User's Guide. You can select the normalization method to be used:
    - **TMM:** Weighted trimmed mean of M-values. In this method, weights are obtained from the delta method on Binomial Data (this method is recommended).
    - **TMM with Zero Pairing:**This is a variant of TMM that should perform better for data with a high proportion of zeros.
    - **RLE:** Relative log expression. Scale factors are the median ratio of each sample to the median library (geometric mean of all samples).
    - **Upper-quartile:** 75% quantile for the counts for each library is used to calculate the scale factors.
    - **None:** No normalization method is applied.

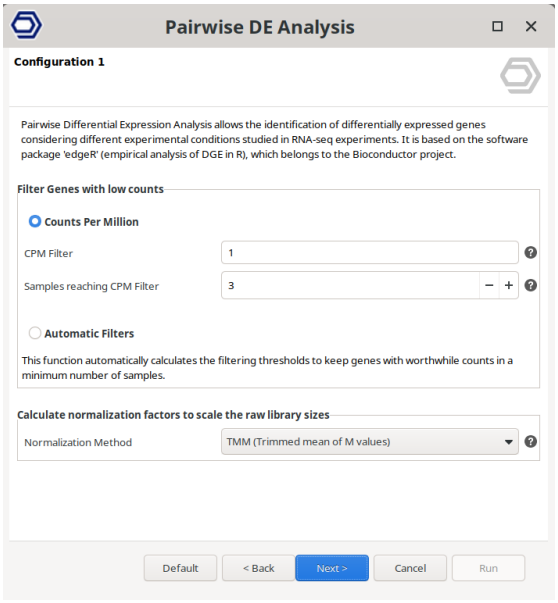


Figure 5: *Preprocessing Data Page*

Experimental Design Page

- **Experimental design file:** Select your .txt file containing your experimental factors with the experimental conditions associated with each sample in tab-delimited format. As shown below, rows correspond to samples and columns to experimental factors. Make sure that the names in the first column of the experimental design table are exactly the same as the sample names in the count table header. If your experimental design file has fewer samples than in the count table, only the samples contained in this file will be analyzed.

```
Name    Strain
SRR3666079  FA1090
SRR3666080  FA1090cpxA
SRR3666081  FA1090cpxR
SRR3666082  FA1090
SRR3666083  FA1090cpxA
SRR3666084  FA1090cpxR
SRR3666085  FA1090
SRR3666086  FA1090cpxA
SRR3666087  FA1090cpxR
SRR3666088  FA1090
```

```
SRR3666089  FA1090cpxA
SRR3666090  FA1090cpxR
```

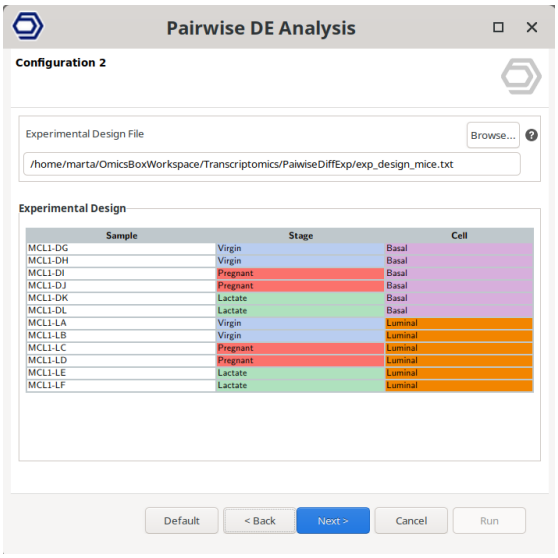


Figure 6: *Experimental Design Page*

Comparison and Test Page

- **Design Type:** Choose the design type to adjust the analysis
  - Simple design: Makes a pairwise comparison between samples belonging to two experimental conditions. You only have to select the experimental factor of interest and establish the comparison by selecting the reference and contrast conditions in "Primary Target".
  - Paired design: Makes a pairwise comparison between samples belonging to two experimental conditions, adjusting for baseline differences of other experimental factors. In this design, you have to establish the conditions for the comparison in "Primary Target" and the experimental factor for baseline difference in "Secondary Target". This design type is appropriate for paired or blocking design, or experiments with batch effects.
  - Multifactorial Design: Makes a pairwise comparison between samples belonging to two experimental conditions with two experimental factors. For this design, you have to select the two experimental factors of interest and establish the reference and contrast group for each in "Primary Target" and "Secondary Target". This design type is appropriate if you want to analyze the effects of combined experimental conditions on gene expression.
- **Statistical Test:** Select a statistical test.
  - Exact Test: Based on the quantile-adjusted conditional maximum likelihood (qCML) methods (similar to Fisher's exact test). It is only applicable to datasets with a single factor design (simple design).
  - GLM (Likelihood Ratio Test): Based on fitting negative binomial Generalized Linear Models (GLMs) with the Cox-Reid dispersion estimates. It is a good choice for inferences with GLMs.
  - GLM (Quasi Likelihood F-Test): The empirical Bayes quasi-likelihood F-test is an alternative to the Likelihood Ratio Test and provides a more robust and reliable error rate control when the number of replicates is small.

• **Robust**: Estimation is strengthened against potential outlier genes.

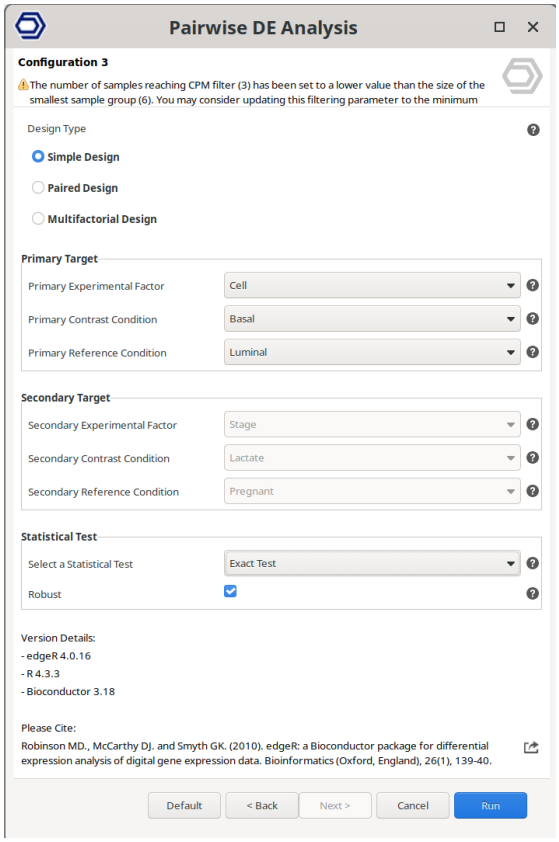


Figure 7: *Comparison and Test Page*

RESULTS

Once the input counts have been processed and analyzed via the "Pairwise Differential Expression Analysis" tool, a new tab is opened containing the results (Figure 8). The results table contains the differential expression statistics, where each row corresponds to a feature:

- **logFC:** A measure that describes how much the expression changes between conditions (log2-fold-changes are shown).
- **logCPM:** The average log2-counts-per-millions.
- **LR:** Likelihood ratio statistic for the GLM (Likelihood Ratio Test).
- **F:** Quasi-likelihood F-statistic for the GLM (Quasi Likelihood F-test).
- **FDR:** False Discovery Rate calculated by the Benjamini-Hochberg method (multiple hypothesis testing corrections).
- **Tags:** Indicate whether a gene is upregulated (FDR ≤ 0.05, logFC ≥ 0) or downregulated (FDR ≤ 0.05, logFC ≥ 0).

Genes that have not passed the filtering step are not shown in the new tab.

Results can be saved as a Pairwise Results object. Note that it is not possible to perform the analysis on this object. For this purpose, you have to open the Count Table object.

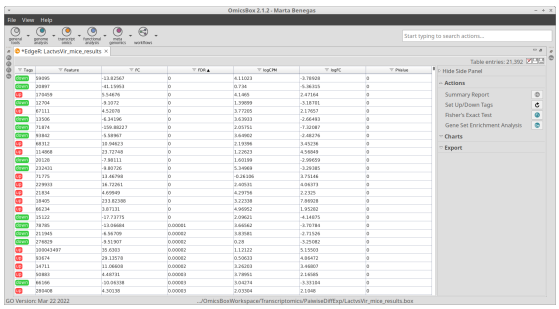


Figure 8: *Pairwise Differential Expression Results*

A result page will show a summary of the pairwise differential expression analysis results (Figure 9).

Side Panel

Actions

The actions are available in the **Side Panel→ Actions**.

Summary Report

It generates the Summary Report explained above.

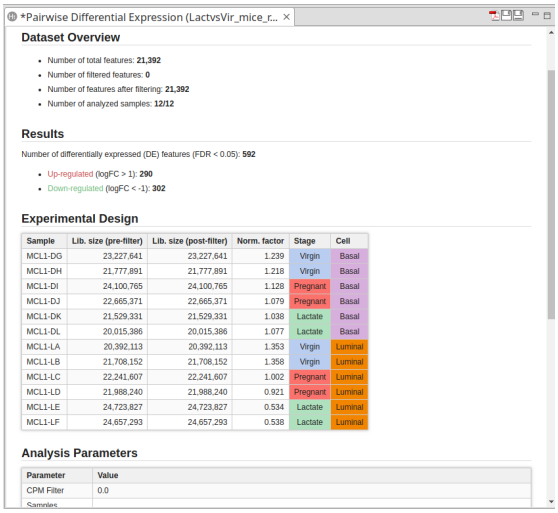


Figure 9: *Results Summary*

Rename Features

This option allows modifying the sequence IDs in the Feature column using different methods:

- **Add**: Add a prefix or sufix to all IDs in the table.
- **Replace**: Replace specific text within the IDs. The text to be replaced must be defined in the Find parameter using a regular expression (regex).
- **Mapping**: Use a mapping file to rename features. The mapping file must be a tab-separated text file with two columns: the first column contains the original feature IDs from the dataset, and the second column contains the new feature names. If duplicate IDs occur during renaming, you can define how they are handled:
    - Sum Rows: Combine counts for all matching features.
    - First Row: Retain only the counts of the first occurrence.

Set Up/Down Tags

It re-assigns the UP and DOWN labels based on different filtering cutoffs (Figure 10). Tags will be updated, and the result section of the Result Summary and statistical charts will change according to the new cutoffs.

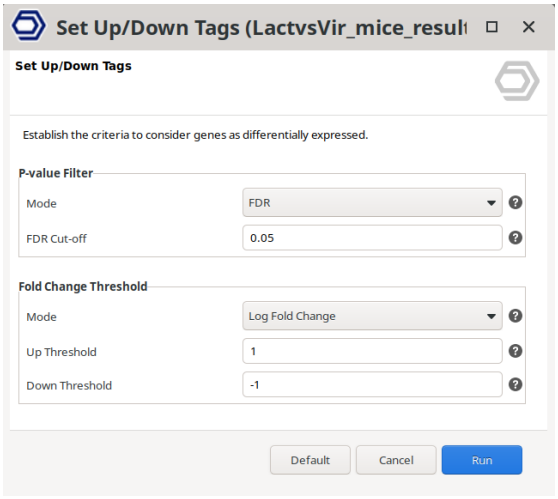


Figure 10: *Set Up/Down Tags*

Fisher's Exact Test

Fisher's Exact Test can be used to find GO terms that are over and under-represented in a set of genes (test set) with respect to a reference group (reference set). Fisher's Exact Test uses a contingency table-based method to examine the association between two kinds of classification. There's more information about the analysis and the parameters in the Fisher's Exact Test section.

With this tool, the subset of genes that will be considered as a Test-set will be the genes labeled as UP or DOWN regulated (Figure 11). Up-regulated and down-regulated genes are those that are tagged according to the criteria established by the option "Set Up/Down Tags".

The project containing the functionally annotated sequences that will be used as a reference background set should be provided in the "Reference Annotation" box.

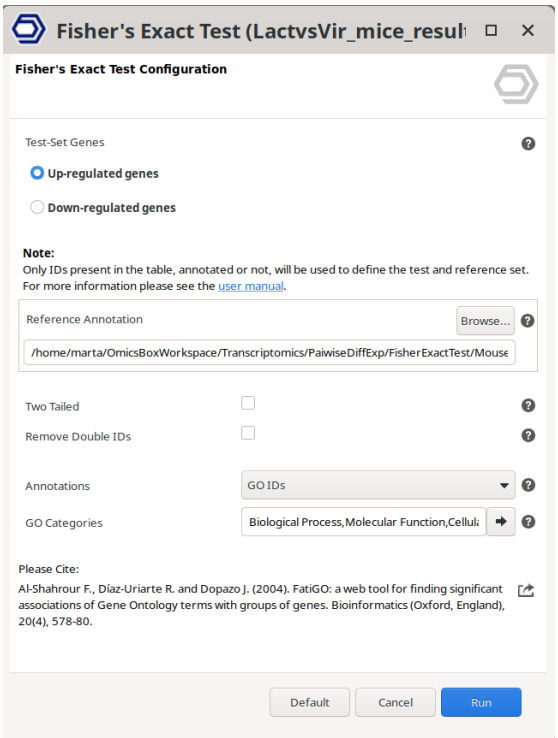


Figure 11: *Fisher's Exact Test*

Gene Set Enrichment Analysis

Gene Set Enrichment Analysis (GSEA) is a computational method that determines whether an a priori defined set of genes shows statistically significant, concordant differences between two biological states (e.g. phenotypes). There's more information about the analysis and the parameters in the Gene Set Enrichment Analysis section.

This analysis needs a ranked gene list, which will be automatically computed using the following formula:

Rank = sign(logFC) * -log10(P-Value)

The project containing the functionally annotated sequences that will be used as a reference background set should be provided (Figure 12).

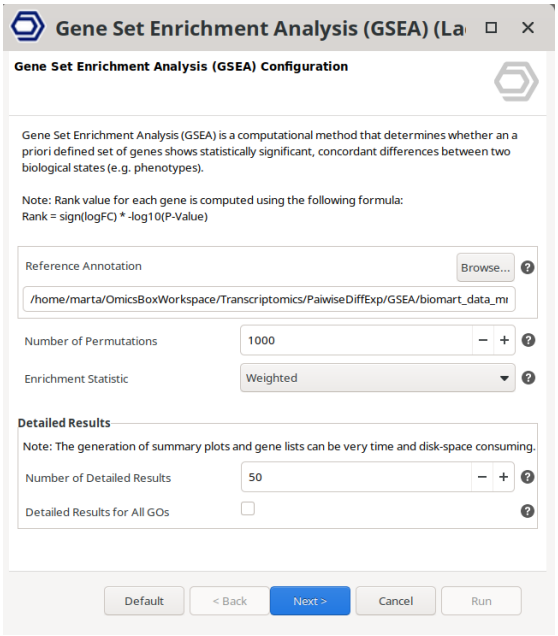


Figure 12: *Gene Set Enrichment Analysis*

Charts

Different statistics charts can be generated for a global visualization of the results. These charts can be found under the **Side Panel → Charts**of the Pairwise Results viewer.

Results Chart

Bar chart which shows the number of total features, kept features (those who have passed the filtering step), differentially expressed features, up-regulated features, and down-regulated features (Figure 13).

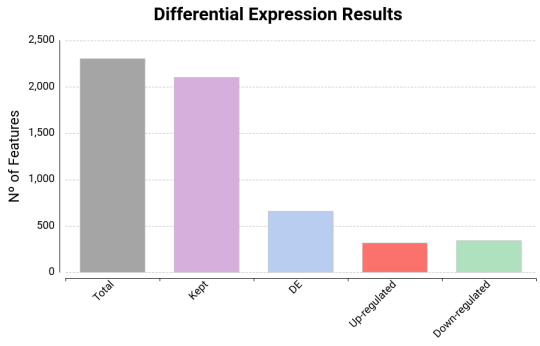


Figure 13: *Result Summary*

MDS Plot

Generates a two-dimensional scatterplot in which the distances represent the typical log2 fold changes between samples. You can select an experimental factor by which you want to color the MDS graphic (Figure 14).

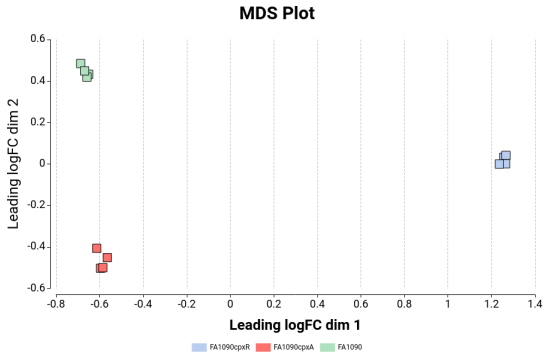


Figure 14: *MDS Plot*

Volcano Plot

A scatter plot constructed by plotting the negative log of the adjusted p-values (FDR) on the y-axis versus the log of the fold changes on the x-axis (Figure 15). Upregulated and downregulated genes are shown in green and red respectively.

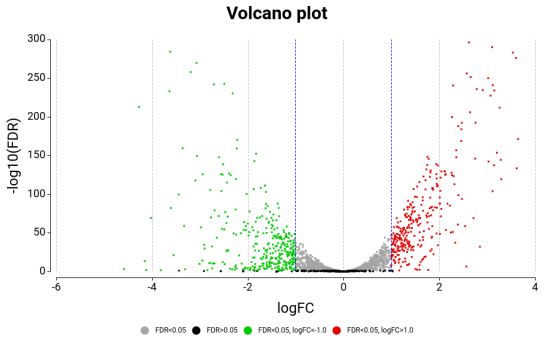


Figure 15: *Volcano Plot*

MA Plot

A scatter plot showing the log of the fold changes on the y-axis versus the average of the log of the CPM on the x-axis. Differentially expressed genes are highlighted (Figure 16).

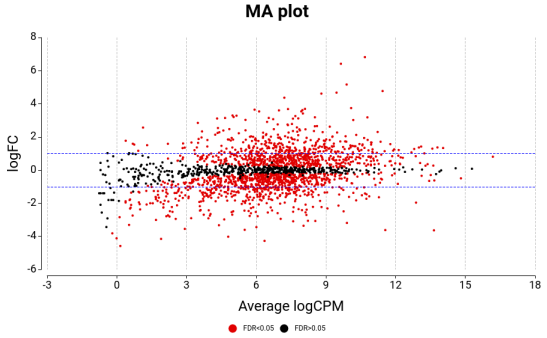


Figure 16: *MA Plot*

Heatmap

A heatmap is a two-dimensional visual representation of data in which numerical values of points are represented by a range of colors (Figure 17). The dendrograms added to the left and top sides are produced by a hierarchical clustering method that takes as input the Euclidean distance computed between genes (left) and samples (top).

The heatmap supports zooming by keeping clicked a node of either of the two dendrograms. The first bars contain the experimental design of the data showing the association between samples and experimental covariates.

Genes that will be displayed can be selected in the wizard. There are three options:

- The Top 50 differentially expressed genes (ranked by FDR).
- All differentially expressed genes.
- Provide an ID list containing the genes to represent.

Differentially expressed genes are those that are labeled as UP or DOWN in the table project ("Tags" column). The criteria for considering a gene as differentially expressed can be adjusted using the option "Set Up/Down Tags".

Furthermore, the wizard allows adjusting the type of expression data that will be represented, as well as the transformation that can be applied to this data.

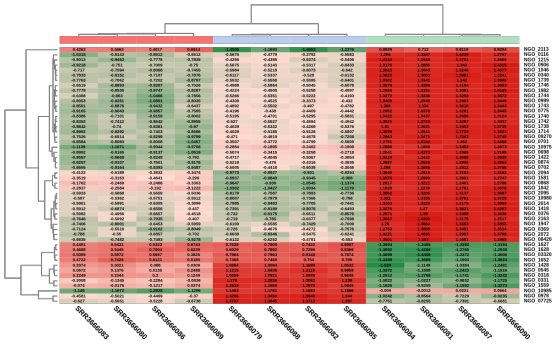


Figure 17:*Heatmap*

Export

Besides the generic Export Table, this object contains the following export options.

Export Raw Counts

Export the raw counts to a text file. It will not contain the genes discarded during the filtering step.

Export Normalized Counts

Export the normalized counts to a text file. It will not contain the genes discarded during the filtering step.

Export Experimental Desing

Esport the experimental design to a tab-separated file. The first column will contain the samples, whereas the rest will be the experimental factors.

Export Ranked List

Export a Ranked List with genes in one column and the rank in another column. Rank value for each gene is computed using the following formula: Rank = sign(logFC) * -log10(P-Value)

Context Menu

Besides the generic context menu options, the available action for this object is:

- **Extract Selection to New Tab**: Extract the data from the selected rows and open it in a new tab.

Time Course Expression Analysis

INTRODUCTION

This tool is designed to perform time-course expression analysis of count data arising from RNA-seq technology. Based on the maSigPro program, this application allows the detection of genomic features (e.g. genes) with significant temporal expression changes and significant differences between experimental groups. The software package **maSigPro**, which belongs to the Bioconductor project, implements a two steps regression strategy to find genes for which there are significant expression profile differences in time course RNA-seq experiments.

Please cite maSigPro as:
Conesa A, Nueda MJ, Ferrer A, Talón M. maSigPro: a method to identify significantly differential expression profiles in time-course microarray experiments. Bioinformatics. 2006 May 1;22(9):1096-102. doi:10.1093/bioinformatics/btl056

Nueda MJ, Tarazona S, Conesa A. Next maSigPro: updating maSigPro bioconductor package for RNA-seq time series. Bioinformatics. 2014;30(18):2598-2602. doi: 10.1093/bioinformatics/btu333

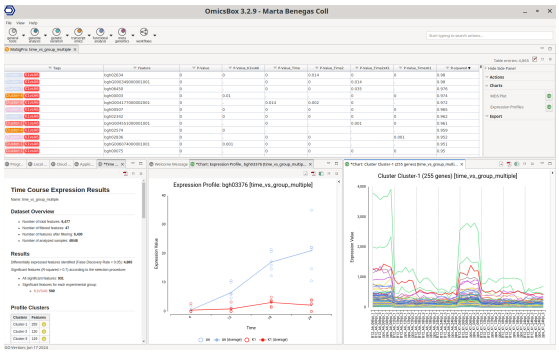


Figure 1: *Time Course Expression Interface*

Expression Data

The pairwise differential expression analysis application expects gene expression levels in a count table. In OmicsBox, count tables can be generated via the **Create Count Table** application.

Count tables can also be imported from a text file. Go to **transcriptomics → Load → Load RNA-Seq Count Table** (Figure 2) and select your .txt file containing the count table.

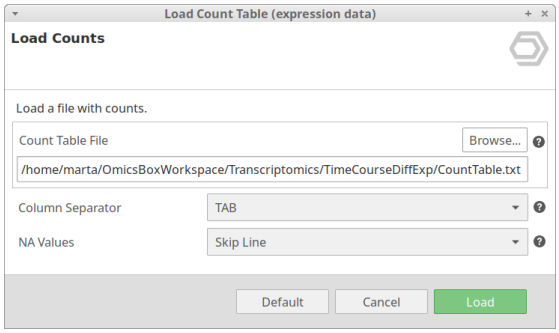


Figure 2:*Count Table File*

RUN ANALYSIS

Go to **transcriptomics → Differential Expression Analysis**. If there's no count table project opened, the first wizard page (Figure 3) will ask to upload either a Count Table Project (.box file) or a Count Table File (.txt, .csv, or .tsv file). On the second wizard page, choose the "**Time Course Expression Analysis**" option (Figure 4).

If a count table is already loaded in OmicsBox (see above section), this one will be used to perform the analysis. In this case, the analysis can be run by both clicking on the "**Diff. Expression Analysis**" in the Side Panel or by going to transcriptomics → Differential Expression Analysis. Now the first wizard page will ask to select the type of differential expression analysis (Figure 4).

In the next pages, it is possible to specify different analysis parameters, which are divided into three distinct sections: Preprocessing Data (Figure 5), Experimental Design (Figure 6), and Analysis Options (Figure 7).

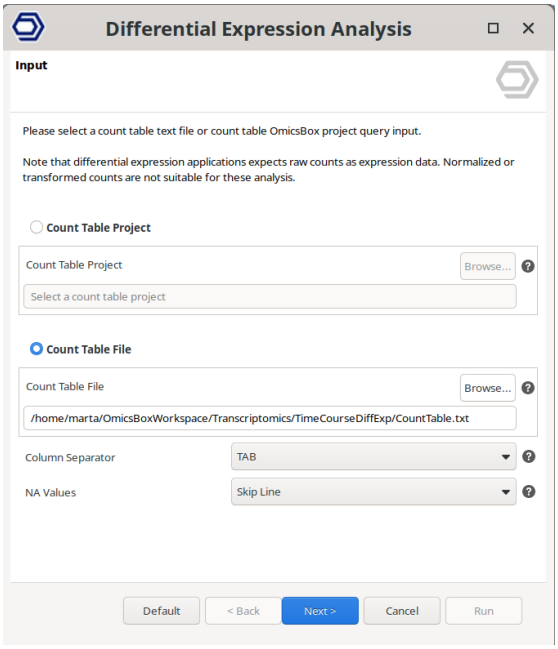


***Figure 3**. Input Wizard Page.*

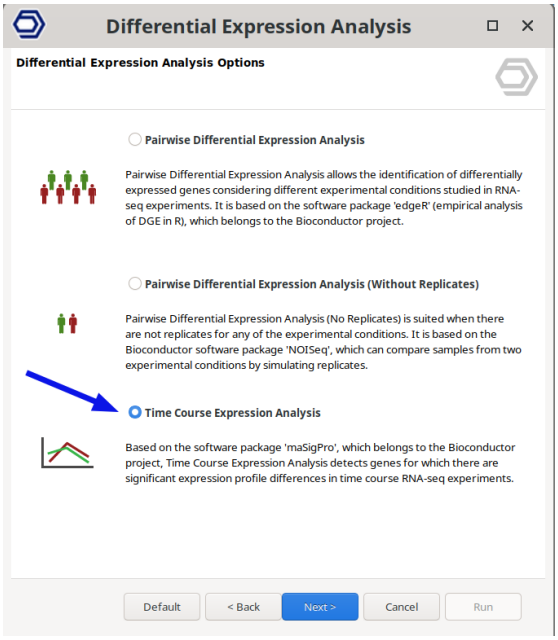


***Figure 4:** Differential Expression Analysis Options wizard page.*

Preprocessing Data Page

- **Filter low count genes:**
- **CPM Filter:** Establish a filter to exclude genes with low counts across libraries, as those genes may interfere with the subsequent statistical approximations. Filtering is performed on a count-per-million (CPM) basis to account for differences in library size between samples (e.g. a CPM of 1 corresponds to a count of 6 in a sample with 6 million reads).
- **Samples reaching CPM Filter:** Set a minimum number of samples in which the gene's CPM is above the filter level (is expressed). If this value is set to e.g. five, at least 5 of the samples have to be above the given CPM. The number of samples of the smallest group is usually taken (e.g. in an experiment that has two replicates for each condition (or group), a gene should be expressed in at least two samples). Set value to 0 if no filter is desired.

- **Normalization procedure:**
- **Normalization Method**: Normalization is an important step to make the samples comparable and to remove possible biases (as sequencing depth bias) in count data. You can select the normalization method to be used:
  - **TMM**: Weighted trimmed mean of M-values. In this method, weights are obtained from the delta method on Binomial Data (this method is recommended).
  - **RPKM**: Reads Per Kilobase per Million mapped reads. This method corrects for gene length and the number of sequencing reads (gene length is required).
  - **Upper-quartile**: 75% quantile for the counts for each library is used to calculate the scale factors for normalization.
  - **None**: No normalization method is applied.
  - **Feature Length File**: For RPKM normalization load a tab-delimited file (or ID-Value object) with two columns containing the name and length of each gene or genomic feature.

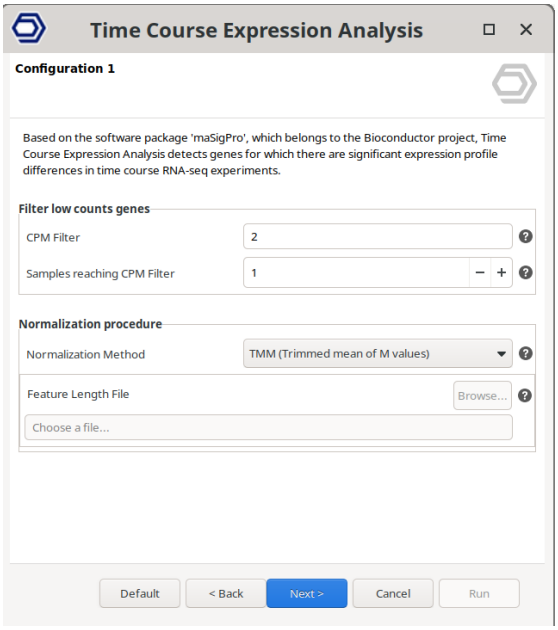


Figure 5:*Preprocessing Data Page*

Experimental Design Page

- **Experimental design file:** Select your .txt file containing your experiment descriptors associated with each sample in tab-delimited format. As shown below, rows correspond to samples and columns to experimental descriptors. A column must contain the associated time points for each sample, and another column should show the assignment of samples to experimental groups. Make sure that the names in the first column of the experimental design table are exactly the same as the sample names in the count table header. If your experimental design file has fewer samples than the count table, only the samples contained in this file will be analyzed.

Click here to expand ...

```
Sample  Time    Group
B12_A6_06hpi_1  6   A6
B12_A6_06hpi_2  6   A6
B12_A6_06hpi_3  6   A6
B12_A6_12hpi_1  12  A6
B12_A6_12hpi_2  12  A6
B12_A6_12hpi_3  12  A6
B12_A6_18hpi_1  18  A6
B12_A6_18hpi_2  18  A6
B12_A6_18hpi_3  18  A6
B12_A6_24hpi_1  24  A6
B12_A6_24hpi_2  24  A6
B12_A6_24hpi_3  24  A6
B12_K1_06hpi_1  6   K1
B12_K1_06hpi_2  6   K1
B12_K1_06hpi_3  6   K1
B12_K1_12hpi_1  12  K1
B12_K1_12hpi_2  12  K1
B12_K1_12hpi_3  12  K1
B12_K1_18hpi_1  18  K1
B12_K1_18hpi_2  18  K1
B12_K1_18hpi_3  18  K1
B12_K1_24hpi_1  24  K1
B12_K1_24hpi_2  24  K1
B12_K1_24hpi_3  24  K1
pps_A6_06hpi_1  6   A6
pps_A6_06hpi_2  6   A6
pps_A6_06hpi_3  6   A6
pps_A6_12hpi_1  12  A6
pps_A6_12hpi_2  12  A6
pps_A6_12hpi_3  12  A6
pps_A6_18hpi_1  18  A6
pps_A6_18hpi_2  18  A6
pps_A6_18hpi_3  18  A6
pps_A6_24hpi_1  24  A6
pps_A6_24hpi_2  24  A6
pps_A6_24hpi_3  24  A6
pps_K1_06hpi_1  6   K1
pps_K1_06hpi_2  6   K1
pps_K1_06hpi_3  6   K1
pps_K1_12hpi_1  12  K1
pps_K1_12hpi_2  12  K1
pps_K1_12hpi_3  12  K1
pps_K1_18hpi_1  18  K1
pps_K1_18hpi_2  18  K1
pps_K1_18hpi_3  18  K1
pps_K1_24hpi_1  24  K1
pps_K1_24hpi_2  24  K1
pps_K1_24hpi_3  24  K1
```

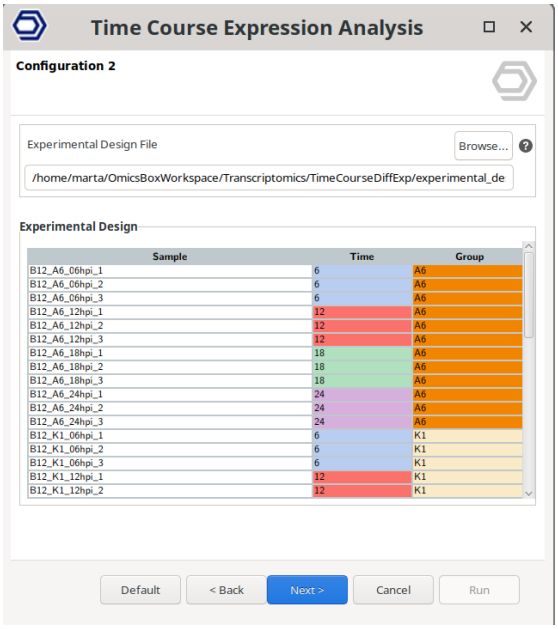


Figure 6:*Experimental Design Page*

Analysis Options

- **Design Type:** Choose the design type to adjust the analysis.
- **Single Series Time Course**: Detects genes that show significant expression changes over time. You only have to select the time factor of your experimental design in "Targets".
- **Multiple Series Time Course**: Find genes with significant temporal expression changes and significant differences between experimental groups. You have to establish the time and experimental factors, and select the control condition of your experimental design in "Targets".
- **Statistical Settings:**
- Significance Level (Alfa): The level of FDR control used for variable selection in the stepwise regression.
- R-squared Cutoff: Cutoff value for the R-squared of the regression model.
- **Visualization of Results:**
- **Number of Clusters**: Establish a number of clusters to group genes by similar expression profiles.
- **Clustering Method**: Choose a clustering method for data partitioning.
    - Hierarchical Clustering: Performs a hierarchical cluster analysis using a set of dissimilarities for the features being clustered.
    - K-Means Clustering: This is intended to divide the points into K clusters such that the sum of squares of the points to the centers of the clusters assigned is minimized.
    - Model-Based Clustering: The optimal model according to BIC for EM is initialized by hierarchical clustering for Gaussian mixture models. This method computes an optimal number of clusters. Keep in mind that this method requires more time.

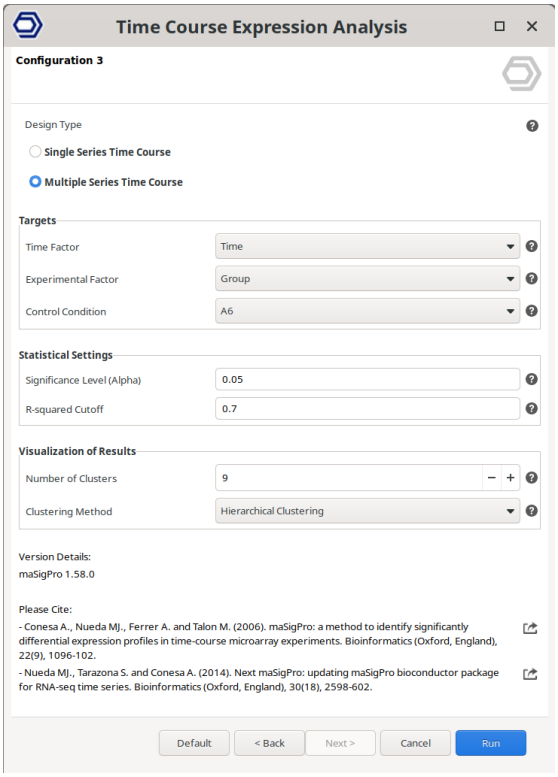


Figure 7:*Analysis Options*

RESULTS

Once the input counts have been processed and analyzed via the "Time Course Expression Analysis" tool, a new tab is opened containing statistical results obtained by the stepwise regression statistical test (Figure 8):

- **Tags**: genes labeled with a tag are found significantly expressed over time and/or between conditions (R-squared ≥ R-squared Cutoff).
- Cluster tags. There will be as many cluster tags as the number specified in the "Number of Clusters" parameter. Each cluster is assigned a different color. Genes found significantly expressed over time are then grouped by their expression profile by the clustering algorithm. Thus, genes labeled with i.e. "Cluster-1" tag, change their expression over time and, moreover, with a similar trend.
- Condition tags. Only available when the "Multiple Series Time Course" option is chosen. Genes labeled with this tag are found to be differentially expressed between conditions. The condition tags are always red and the text is the comparison made. For example, if we have three conditions A (control), B, and C, we will have two read condition tags: "AvsB" and "AvsC".
- **Feature**: feature or gene name.
- **P-Value**: if it's significant, it indicates that the gene expression changes over time.
- **P-Value_beta0**: if it's significant, it means that the gene's expression at time point 0 is different from 0.
- **P-Value_Time**: if it's significant, it means that the gene expression follows a linear trend, especially at the beginning. That is, that it increases or decreases linearly.
- **P-Value_Time2**: if it's significant, that means that the gene expression profile has a curvature. That is, it changes the expression behavior at some point (i.e. first the expression increases and then decreases).
- **R-squared**: how well the data fits the model obtained for that gene's expression.

For the "Multiple Series Time Course", additional p-values are calculated, one for each control vs condition combination. For two conditions named "A" (control) and "B" :

- **P-Value_BvsA**: if it's significant, it means that the gene's expression at time point 0 in condition B is different than in A.
- **P-Value_TimexB**: if it's significant, it means that the linear gene expression is different between conditions A and B. That is, that the gene expression in one condition increases or decreases more than in the other condition.
- **P-Value_Time2xB**: if it's significant, that means that the change in expression is different for A and B (i.e. in condition A it increases and then decreases, but in condition B it never decreases).

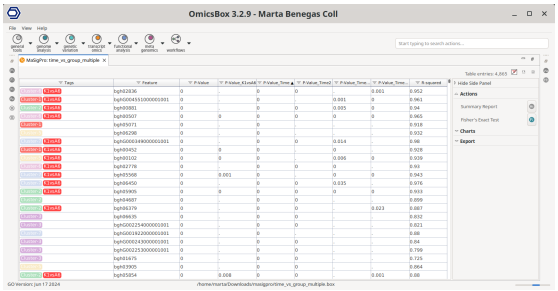


Figure 8:*Table Viewer*

Only the genes that have passed the established Significance Level are shown in the new tab. For further details please refer to the maSigPro User's Guide.

There could be missing p-values. That means that this characteristic is not significant in the gene expression profile. So it is not considered for constructing the gene's expression model, and thus its value is not stored.

Additionally, a report will be open that shows a summary of the time-course expression analysis results, including the cluster of features with similar expression profiles (Figure 9).

Side Panel Options

Actions Summary Report

Generate a summary of the results as shown in Figure 9.

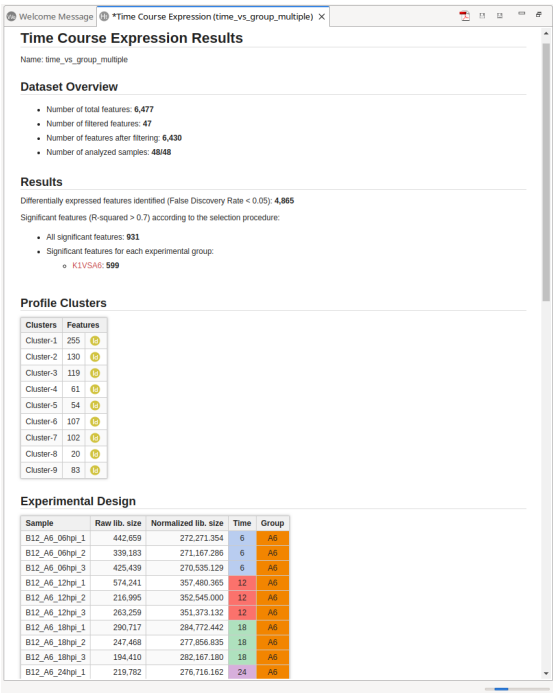


Figure 9: *Summary Report*

Rename Features

This option allows modifying the sequence IDs in the Feature column using different methods:

- **Add**: Add a prefix or sufix to all IDs in the table.
- **Replace**: Replace specific text within the IDs. The text to be replaced must be defined in the Find parameter using a regular expression (regex).
- **Mapping**: Use a mapping file to rename features. The mapping file must be a tab-separated text file with two columns: the first column contains the original feature IDs from the dataset, and the second column contains the new feature names. If duplicate IDs occur during renaming, you can define how they are handled:
    - Sum Rows: Combine counts for all matching features.
    - First Row: Retain only the counts of the first occurrence.

Fisher's Exact Test

Fisher's Exact Test (FET) can be used to find biological functions (represented by GO terms or other annotations) over and under-represented in a set of genes (test set) with respect to a reference group (reference set). Roughly speaking, it tests if the proportion of genes annotated with a specific biological function in the test set is significantly higher or lower than the proportion in the reference set. For more details about the analysis and the **results**, please visit the Fisher's Exact Test section on the Functional Analysis module. In this case, the test set is made with the genes labeled with a certain tag(s).

- **Test Set**.
- Groups. Select the tags to test for functional enrichment. The genes labeled with the selected tag(s) will be used as the test set.
- **Reference Set**.
- Remaining Genes. If checked, use the remaining genes in the count table, that are not part of the test set, as the reference set.
- Groups. Only available if the "Remaining Genes" option is unchecked. The genes labeled with the selected tag(s) will be used as the reference set. They can't overlap with the tags selected in the test set.

If a gene is present both in the test and in the reference sets, it will be removed from the reference.

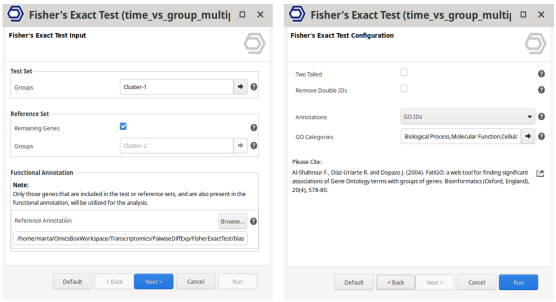


Figure 10: *Fisher's Exact Test wizard from timecourse results.*

Charts

Different statistics charts can be generated for a global visualization of the results. These charts can be found under the **Side Panel**of the TimeCourse Results viewer.

MDS Plot

Generates a two-dimensional scatterplot in which the distances represent the typical log2 fold changes between samples. You can select an experimental factor by which you want to color the MDS graphic (Figure 11).

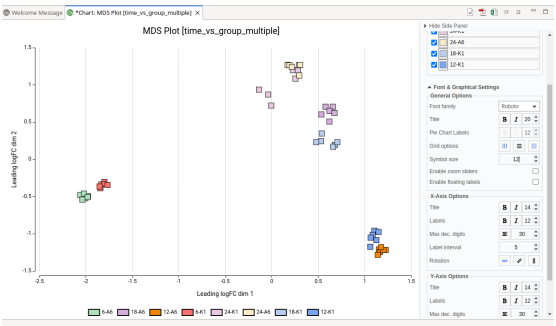


Figure 11: *MDS Plot*

Experiment-wide Expression Profiles

Plot showing the expression level levels across samples for each cluster of genes (Figure 12).

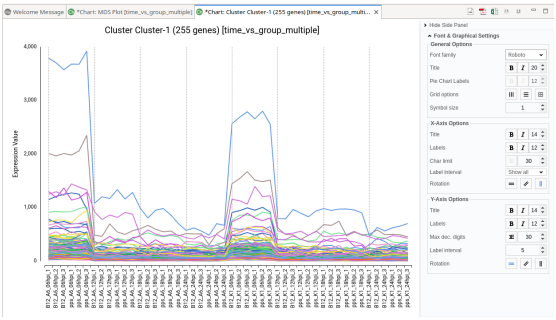


Figure 12: *Experiment-wide Expression Profile*

Summary Expression Profiles

Plot showing the median level expression of each cluster of genes across time (Figure 13).

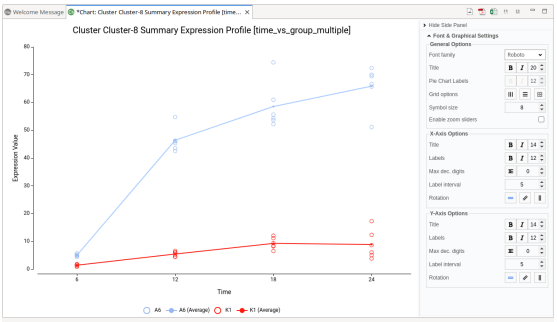


Figure 13: *Summary Expression Profile*

Export

Besides the generic Export Table, this object contains the following export options.

Export Raw Counts

Export the raw counts to a text file. It will not contain the genes discarded during the filtering step.

Export Normalized Counts

Export the normalized counts to a text file. It will not contain the genes discarded during the filtering step.

Export Experimental Desing

Esport the experimental design to a tab-separated file. The first column will contain the samples, whereas the rest will be the experimental factors.

Context Menu

Besides the generic context menu options, the available actions for this object depend on whether one or multiple rows are selected.

With one row selected:

- **Show Expression Profile**: Generates a plot with the average expression across samples of the given gene over time (Figure 14).

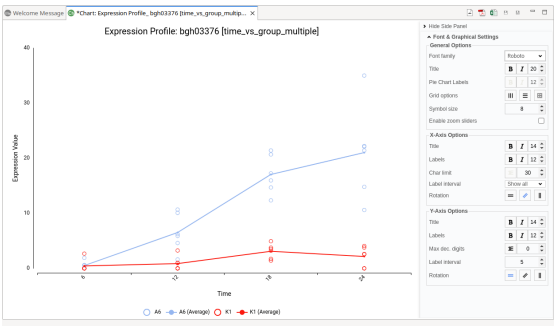


Figure 14: *Gene Expression Profile*

With multiple rows selected:

- **Extract Selection to New Tab**: Extract the data from the selected rows and open it in a new tab.

Pairwise Differential Expression Analysis (Without Replicates)

INTRODUCTION

Detecting genes that are differentially expressed between two experimental conditions (e.g. diseased vs healthy individuals) is a fundamental part of understanding the molecular basis of phenotypic variation. To carry out this task, there are statistical tools designed to perform differential expression analysis of the count data arising from RNA-seq technology (e.g. edgeR and maSigPro are statistical packages integrated into OmicsBox). However, these tools usually require the presence of replicates (both biological and technical) of each experimental condition that will be tested. This is a problem in cases where no replicates are available.

The **Pairwise Differential Expression Analysis (Without Replicates)** functionality offers a strategy for analyzing RNA-seq datasets that do not have replicates. It is based on the software package **NOISeq**, which belongs to the Bioconductor project. NOISeq is a novel nonparametric approach for the identification for differentially expressed genes from RNA-Seq count data. NOISeq creates a null or noise distribution of count changes by contrasting fold-change differences (M) and absolute expression differences (D) for all the genes in samples within the same condition. This reference distribution is then used to assess whether the M and D values computed between two conditions for a given gene are likely to be part of the noise or represent a true differential expression.

NOISeq method was designed to compute a differential expression on RNA-Seq data even when there are no replicates available for any of the experimental conditions. In this scenario, NOISeq can simulate technical replicates. The simulation relies on the assumption that read counts follow a multinomial distribution, where probabilities for each class (feature) in the multinomial distribution are the probability of a read to map to that feature. These mapping probabilities are approximated by using counts in the only sample of the corresponding experimental conditions. Given the sequencing depth (total amount of reads) of the unique available sample, the size of the simulated is a percentage of this sequencing depth, allowing a small variability.

Please remember that to obtain really reliable statistical results, biological replicates are needed.

Please cite NOISeq as:

Tarazona S, Furio-Tari P, Turra D, Di Pietro A, Nueda MJ, Ferrer A and Conesa, A (2015). "Data quality aware analysis of differential expression in RNA-seq with NOISeq R/Bioc package." Nucleic Acids Research, 43(21), e140.

Tarazona S, Garcia-Alcalde F, Dopazo J, Ferrer A and Conesa A (2011). "Differential expression in RNA-seq: a matter of depth." Genome Research, 21(12), 2213-2223.

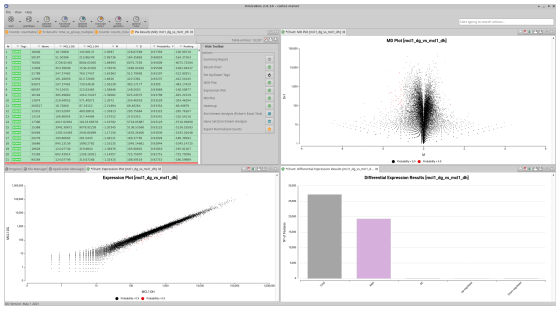


Figure 1: *Differential Expression Interface*

Expression Data

The pairwise differential expression analysis application expects gene expression levels in the form of a count table. In OmicsBox, count tables can be generated via the **Create Count Table** application.

Count tables can also be imported from a text file. Go to **transcriptomics → Load → Load RNA-Seq Count Table (expression data)** (Figure 2) and select your .txt file containing the count table.

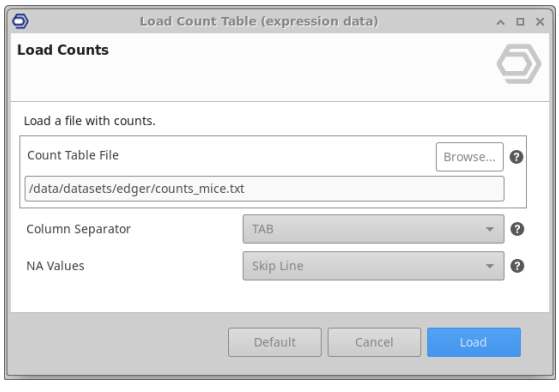


Figure 2:*Count Table File*

RUN PAIRWISE DIFFERENTIAL EXPRESSION ANALYSIS (WITHOUT REPLICATES)

Go to **transcriptomics → Run Differential Expression Analysis** and choose the "Pairwise Differential Expression Analysis (Without replicates) option. This application requires a **Count Table** object as input data. The wizard allows to adjust analysis parameters (Figure 3 and Figure 4).

Preprocessing Data

- **CPM FIlter:** Establish a filter to exclude genes with low counts across libraries, as those genes may interfere with the subsequent statistical approximations. Filtering is performed on a count-per-million (CPM) basis to account for differences in library size between samples (e.g. a CPM of 1 corresponds to a count of 6 in a sample with 6 million reads). To pass the filter, the gene's CPM should be above the filter level in at least one sample (contrast or reference sample).
- **Normalization Method:** Normalization is an important step to make the samples comparable and to remove possible biases (as sequencing depth bias) in count data. The normalization methods available for this analysis are:
  - **TMM:** Weighted trimmed mean of M-values. In this method, weights are obtained from the delta method on Binomial Data (this method is recommended).
  - **RPKM:** Reads Per Kilobase per Million mapped reads. This method corrects for gene length and the number of sequencing reads (gene length is required).
  - **Upper-quartile:** 75% quantile for the counts for each library is used to calculate the scale factors for normalization.
  - **None:** No normalization method is applied.
- **Feature Length File**: For RPKM normalization load a tab-delimited file (or ID-Value object) with two columns containing the name and length of each gene or genomic feature.

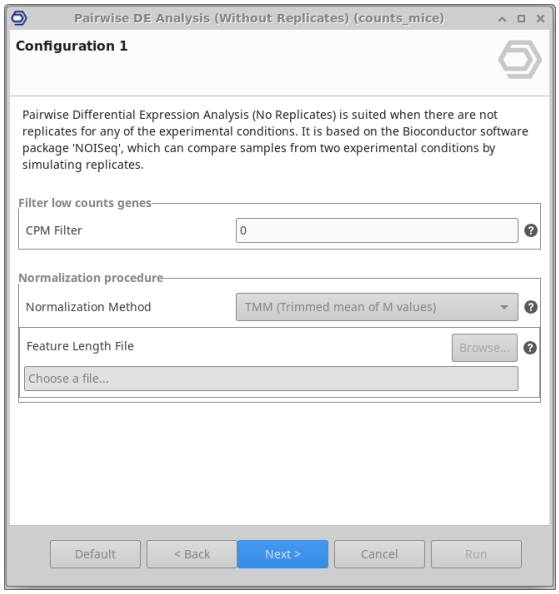


Figure 3: *Preprocessing Data Page*

Analysis Options

- **Replicates Simulation:**
- **Number of Simulated Replicates:** Set the number of replicates to be simulated for each condition.
- **Size of the Simulated Replicates:** Establish the percentage of the total reads used to simulate each sample.
- **Variability:** Variability in the simulated sample total reads.
- **Targets:**
- **Contrast Condition:** Choose the sample to be treated as contrast condition. Genes classified as UP will be upregulated in this sample.

- **Reference Condition:** Choose the sample to be treated as reference condition. Genes classified as DOWN will be upregulated in this sample.

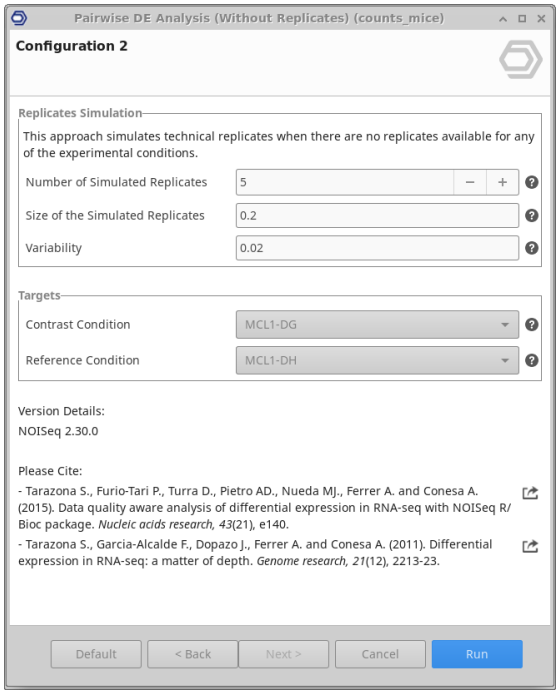


Figure 4: *Analysis Options Page*

RESULTS

Once the input counts have been processed and analyzed via the "Pairwise Differential Expression Analysis (Without Replicates)" feature, a new tab is opened containing the results (Figure 5). The results table contains the differential expression statistics, where each row corresponds to a feature:

- **Contrast Condition:** Normalized expression values for the contrast condition sample.
- **Reference Condition:** Normalized expression values for the reference condition sample.
- **M:** Is the log2-ratio of the two conditions.
- **D:** The value of the difference between the conditions.
- **Probability:** The probability of differential expression for each feature. It is obtained by comparing the M and D values of a given feature against the noise distribution. If the probability is higher than a given threshold (0.9 by default), the feature is considered to be differentially expressed between conditions.
- **Ranking:** Is a summary statistic of M and D values equal to -sign(M)*sqrt(M^2 + D^2), which can be used as a ranked value in gene set enrichment analysis (GSEA).

Genes that have not passed the filtering step are not shown in the results tab.

Results can be saved as a Pairwise Results object. Note that it is not possible to perform the analysis on this object. For this purpose, you have to open the Count Table object.

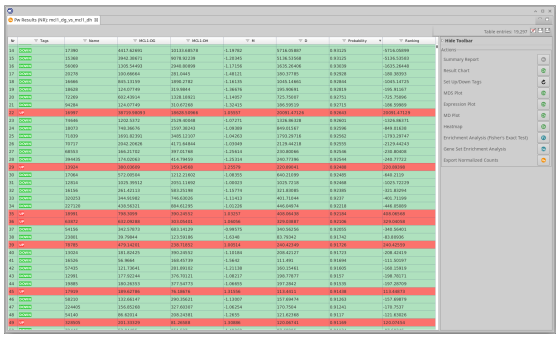


Figure 5: *Differential Expression Results*

A result page will show a summary of the pairwise differential expression analysis results (Figure 6).

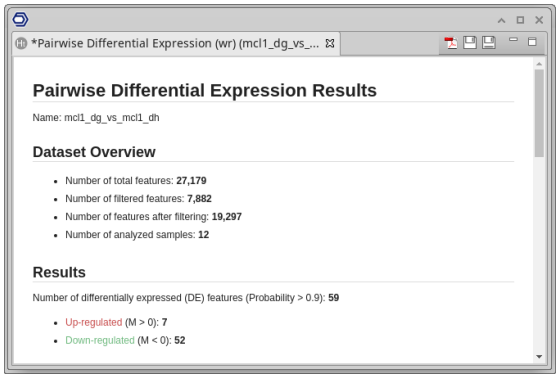


Figure 6: *Results Summary*

Side Panel

Actions Summary Report

It shows the Summary report previously explained in the above "Results" section (Figure 6).

Rename Features

This option allows modifying the sequence IDs in the Feature column using different methods:

- **Add**: Add a prefix or sufix to all IDs in the table.
- **Replace**: Replace specific text within the IDs. The text to be replaced must be defined in the Find parameter using a regular expression (regex).
- **Mapping**: Use a mapping file to rename features. The mapping file must be a tab-separated text file with two columns: the first column contains the original feature IDs from the dataset, and the second column contains the new feature names. If duplicate IDs occur during renaming, you can define how they are handled:
  - Sum Rows: Combine counts for all matching features.
  - First Row: Retain only the counts of the first occurrence.

Set Up/Down Tags

It re-assigns the UP and DOWN labels based on different filtering cutoffs (Figure 7). Tags will be updated, and the result section of the Result Summary and statistical charts will change according to the new cutoffs.

During the Pairwise Differential Expression Analysis (without replicates), raw counts are transformed according to the normalization method selected in the analysis configuration. Go to **Export Normalized Counts**(sidebar) to export normalized counts to a tabular text file.

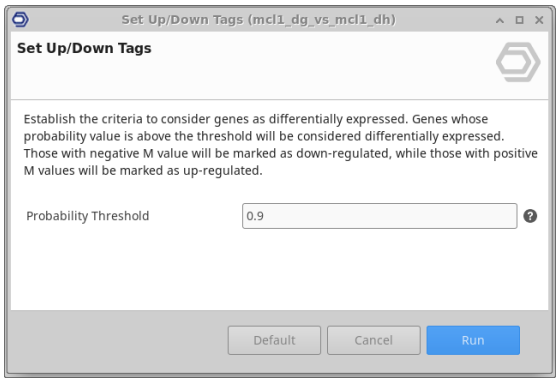


Figure 7: *Set Up/Down Tags*

Fisher's Exact Test

Fisher's Exact Test can be used to find GO terms that are over and under-represented in a set of genes (test set) with respect to a reference group (reference set). This set of genes can be the differentially expressed genes of differential expression analysis, a set of genes related to a phenotype of interest, etc. Fisher's Exact Test uses a contingency table-based method to examine the association between two kinds of classification.

The subset of genes that will be considered as a Test-set (Figure 8) has to be provided. Up-regulated and down-regulated genes are those that are tagged according to the criteria established by the option "Set Up/Down Tags".

The project containing the functionally annotated sequences that will be used as a reference background set should be provided.

The remaining parameters are explained in the Fisher's Exact Test section.

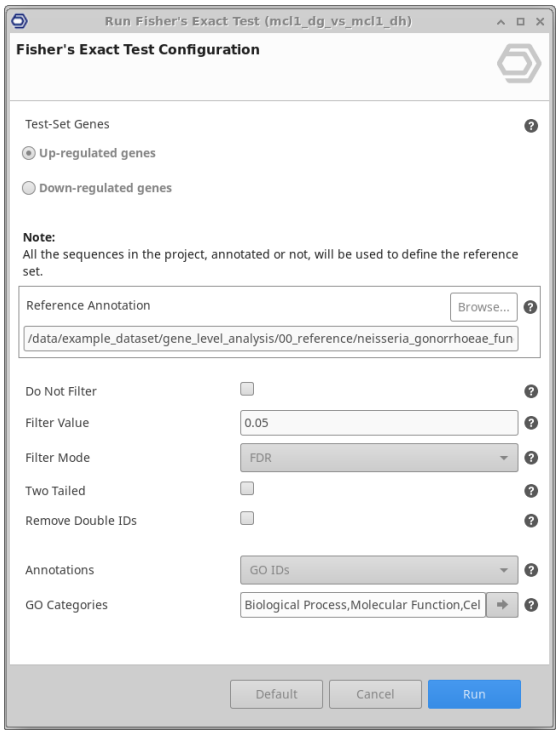


Figure 8: *Fisher's Exact Test*

Gene Set Enrichment Analysis

Gene Set Enrichment Analysis (GSEA) is a computational method that determines whether an a priori defined set of genes shows statistically significant, concordant differences between two biological states (e.g. phenotypes).

The "Probability Threshold for Ranked List" parameters allow setting a filter to exclude those genes whose probability value is not above it (Figure 9). The ranked gene list will be created using the "ranking" statistic.

The project containing the functionally annotated sequences that will be used as a reference background set should be provided.

The rest of the parameters are explained in the Gene Set Enrichment Analysis section.

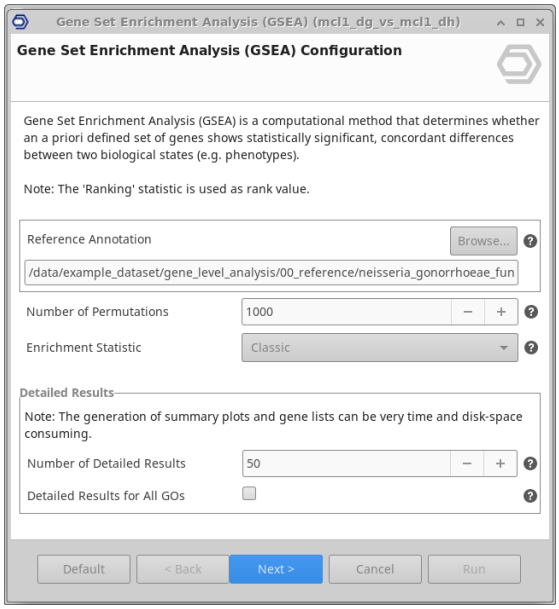


Figure 9: *Gene Set Enrichment Analysis*

Charts

Different statistics charts can be generated for a global visualization of the results. These charts can be found under the **Side Panel**of the Pairwise Results Viewer.

Results Chart

A bar chart shows the number of total features, kept features (those who have passed the filtering step), differentially expressed features, up-regulated features, and down-regulated features (Figure 10).

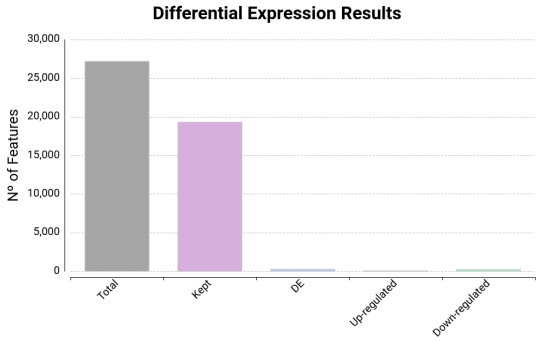


Figure 10: *Results Chart*

MDS Plot

It generates a two-dimensional scatterplot in which the distances represent the typical log2 fold changes between samples. You can select an experimental factor by which you want to color the MDS graphic (Figure 11).

This plot is only available if the input count table contains more than 2 samples (although only two of them are compared).

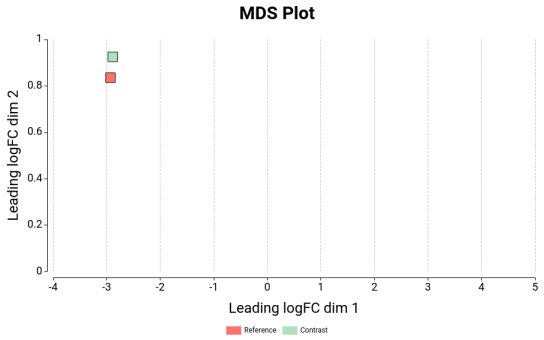


Figure 11: *MDS Plot*

Expression Plot

A scatter plot showing the average expression values of each condition (Figure 12). Differentially expressed features considering the probability threshold (0.9 by default) will be highlighted in red.

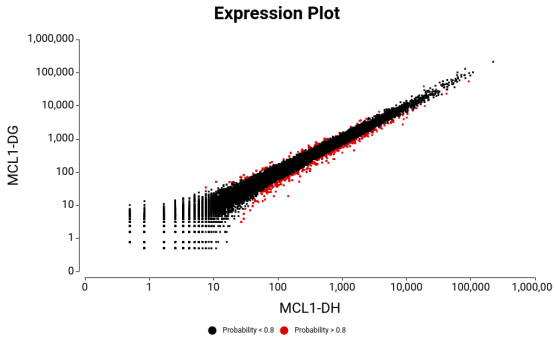


Figure 12:*Expression Plot*

MD Plot

A scatter plot showing the log-fold change (M) and the absolute value of the difference in expression between conditions (D). D values are displayed in a log scale (Figure 13).

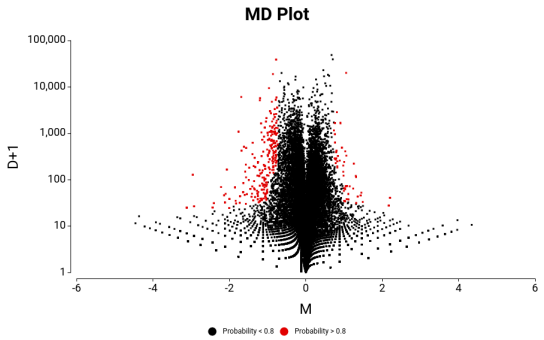


Figure 13: *MD Plot*

Heatmap

A heatmap is a two-dimensional visual representation of data in which numerical values of points are represented by a range of colors (Figure 14). The dendrograms added to the left and top sides are produced by a hierarchical clustering method that takes as input the Euclidean distance computed between genes (left) and samples (top).

The heatmap supports zooming by keeping clicked a node of either of the two dendrograms. The first bars contain the experimental design of the data showing the association between samples and experimental covariates.

Genes that will be displayed can be selected in the wizard. There are three options:

- The Top 50 differentially expressed genes (ranked by FDR).
- All differentially expressed genes.
- Provide an ID list containing the genes to represent.

Differentially expressed genes are those that are labeled as UP or DOWN in the table project ("Tags" column). The criteria for considering a gene as differentially expressed can be adjusted using the option "Set Up/Down Tags".

Furthermore, the wizard allows adjusting the type of expression data that will be represented, as well as the transformation that can be applied to this data.

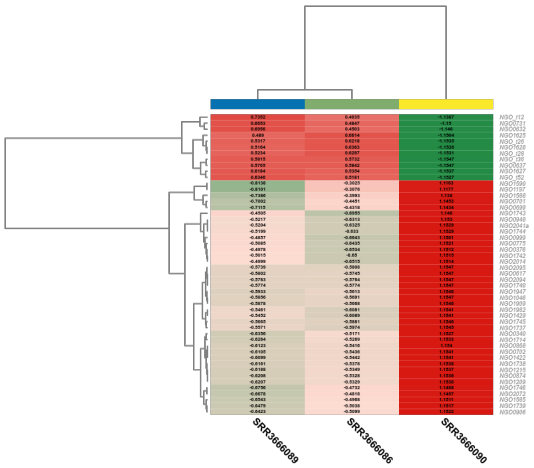


Figure 14: *Heatmap*

Export

Besides the generic Export Table, this object contains the following export options.

Export Raw Counts

Export the raw counts to a text file. It will not contain the genes discarded during the filtering step.

Export Normalized Counts

Export the normalized counts to a text file. It will not contain the genes discarded during the filtering step.

Export Experimental Desing

Esport the experimental design to a tab-separated file. The first column will contain the samples, whereas the rest will be the experimental factors.

Context Menu

Besides the generic context menu options, the available action for this object is:

- **Extract Selection to New Tab**: Extract the data from the selected rows and open it in a new tab.

4.4.11 Coding Potential

Introduction

Thanks to the Next Generation Sequencing methods, transcriptomes are becoming more and more abundant. Once the transcripts have been assembled and we dispose of the sequences that have been transcribed into RNA, we must distinguish between the transcripts that will be coding (mRNA) and the non-coding ones (ncRNA). This classification can be done by assigning to each transcript a score based on his nucleotide composition and patterns.

The "Coding Potential Assessment Tool" provides an easy and fast way to classify the transcripts according to their coding score. This tool integrates the **CPAT** algorithm within OmicsBox. The CPAT algorithm needs models in order to assign the coding potential scores to each sequence. OmicsBox incorporates the standard CPAT models and adds some of the most common organisms models used in molecular biology. In addition to the prebuilt models, this tool adds the option to create your species-specific model.

Run Coding Potential Assessment Tool

This tool can be found under **Functional Analysis → Coding Potential Assessment (CPAT).** The wizard allows adjusting analysis parameters (Figure 2).

- **Accuracy:** By default, the accuracy is set automatically in order to reduce the false positives and the false negatives, this means that the threshold equals the value where the sensitivity has the same value as the specificity.

If higher accuracy is desired the accuracy can be set manually. Raising the accuracy will allow classifying the sequences into three categories: coding, non-coding, and transcripts with unknown coding potential (Figure 1).

The accuracy can be set manually but it can never be lower than the default value. In this case, the accuracy value will automatically fall back to the default value.

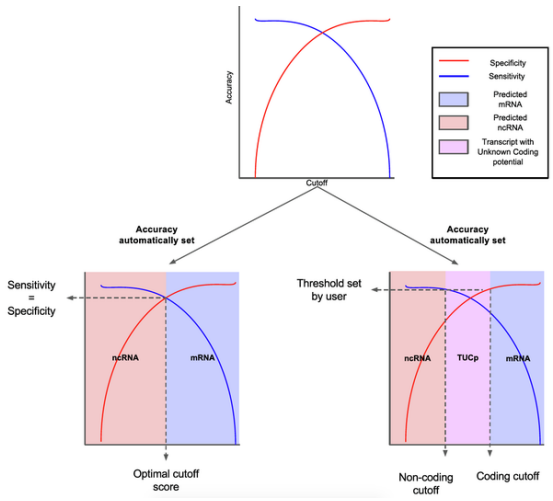


***Figure 1**: Accuracy: Interpretation of the double ROC Curve*

- **Models:** The algorithm needs models to calculate the coding potential for each transcript. Here we can choose the origin of these models:
- Prebuilt: Use one of the prebuilt models available. Selecting one of these prebuilt models, the algorithm will run faster.

Species	Accuracy	Coding Cutoff
Arabidopsis thaliana	0.984	0.415
Bos Taurus	0.953	0.359
Caenorhabditis elegans	0.998	0.523
Danio rerio	0.984	0.38
Drosophila melanogaster	0.963	0.39
Gallus gallus	0.93	0.402
Homo sapiens	0.966	0.364
Mus musculus	0.955	0.440
Rattus norvegicus	0.98	0.363
Sus scrofa	0.946	0.467
Xenopus laevis	0.963	0.415

- **From files**: Create the model providing 2 FASTA files; one with coding sequences and another one with non-coding sequences. Please make sure to follow these guidelines: http://rna-cpat.sourceforge.net/#how-to-prepare-training-dataset
- **From NCBI sequences**: Create a new taxa-specific model from the sequences available at NCBI.

This will take into account the following for the model creation.

- The amount of ncRNA and CDS is the same.
- CDS and ncRNA datasets do not contain any duplicates.
- All CDS lengths are divisible by 3.
- To reach the minimum number of CDS and ncRNA, the tool will search in parent taxa up to the phylum rank, if necessary.

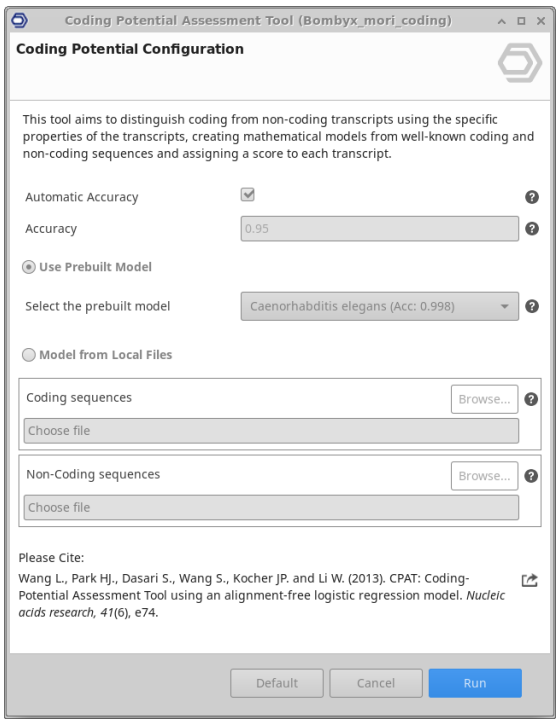


Figure 2: *Wizard Page*

Results

Once finished, three results are automatically created:

Coding Potential Table

A table containing the coding potential results for each input sequence (Figure 3).

- Tag: Marking for each sequence whether it is a coding, non-coding, or unknown coding potential transcript.
- Sequence: The name of the sequence.
- mRNA size: The length of the original transcript.
- ORF size: The size of the potential ORF within the sequence.
- Fickett score: The Fickett score is a linguistic feature that distinguishes protein-coding RNA and ncRNA according to the combinational effect of nucleotide composition and codon usage bias.
- Hexamer score: The hexamer score is calculated using a log-likelihood ratio to measure differential hexamer usage between coding and non-coding sequences.
- Coding Probability: The coding probability assigned to each transcript.

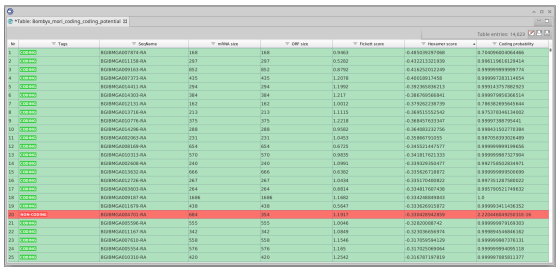


Figure 3: *CPAT Results*

Pie Chart

The coding potential distribution is shown as a pie chart of the classification results for the corresponding sequences depending on the provided cutoffs (Figure 4).

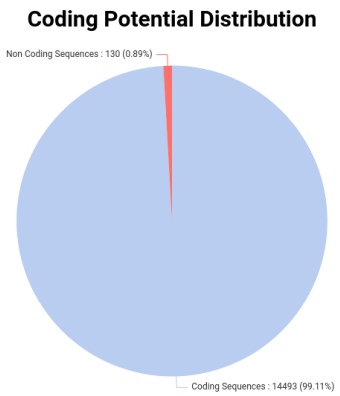


Figure 4: *Coding Potential Distribution*

Model Accuracy via a double ROC-Curve chart

This chart opens when a new model is created or when the accuracy is manually set. In this chart, we can check the quality, accuracy, and the different thresholds of a model (Figure 5).

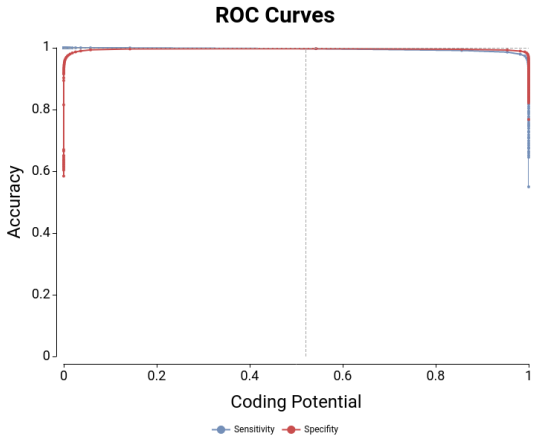


Figure 5: *Double ROC Curve*

4.4.12 Single Cell Data Analysis Tools

Single Cell Data Analysis Tools

Single Cell RNA-Seq Quantification

SINGLE CELL RNA-SEQ QUANTIFICATION

Introduction

There are two main groups of Single-cell RNA-Seq (scRNA-Seq) library construction technologies: Cell Barcoded (or 3' enriched) and Full-length technologies. The final structure of the sequencing reads will heavily depend on which technology has been used during the library preparation step.

The Cell Barcoded technologies attach two main types of tags to the sequencing reads: Cell Barcodes (CB) and Unique Molecular Identifiers (UMI). The CB is unique for each cell, and the UMI is unique for each RNA molecule. Thus, the CB helps to identify the cell and the UMI to identify the RNA molecule from which the read originated. Thanks to the CB, all the cells can be sequenced together, thus allowing a higher throughput with this type of technology. The UMI helps in reducing the amplification bias introduced during the sequencing. Since the CB and the UMI have to be sequenced, the total length of the transcript being sequenced tends to be short and biased towards one end of the molecule (3' or 5'). Examples of these technologies are 10x Chromium, Drop-seq, etc.

On the other hand, Full-length technologies don't use Cell Barcodes nor UMIs. This allows the sequencing of the RNA molecule from both ends, that's why they are called "Full-length". Since no CB is used, cells have to be sequenced separately, so the throughput is lower in comparison with Cell Barcoded technologies. Examples of these technologies are SMART-Seq, SMARTer, etc.

In Single-cell RNA-Seq analysis, gene expression level is measured by cell. Thus, it is important to take into account the structure of the reads in order to separate them and count them at the cell level. The scRNA-Seq Create Count Table tool uses STARsolo, which is the Single-cell version of the well-known aligner STAR. This tool aligns the reads against the reference genome, demultiplexes them by cells, and counts them by the genes present in the GFF or GTF annotation. In addition, in the case of analyzing Cell Barcoded technologies, it includes different options to perform the Cell Barcode calling, and CB and UMI correction and filtering. This makes the tool highly flexible and capable of analyzing reads coming from most of the scRNA-Seq library construction technologies available.

Run Single Cell RNA-Seq Quantification

Go to *transcriptomics* → *Single Cell RNA-Seq* → *Single Cell RNA-Seq Quantification*. A new wizard will be opened (Figure 1). On the first wizard page, select the type of technology used during the library preparation step. Please refer to the above introductory section for more details. Depending on the type chosen, the following pages will display different configuration parameters.

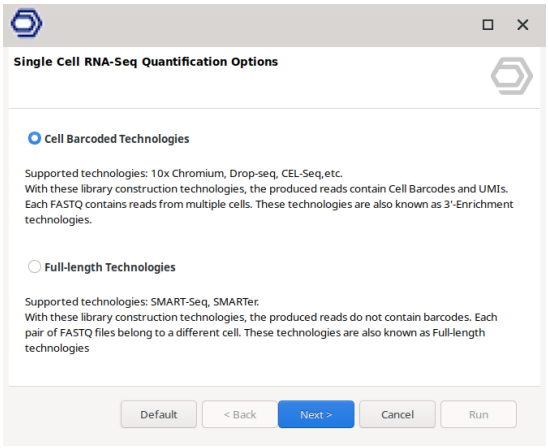


Figure 1. First wizard page.

Cell Barcoded Technologies

The pipeline to obtain counts from barcoded scRNA-Seq data consists of many steps, summarized in Figure 2. Each of the steps can be configured in the following wizard pages. This makes the analysis very customizable and adaptable to the dataset under study.

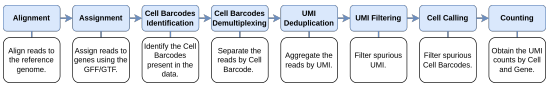


Figure 2. scRNA-Seq Quantification pipeline for barcoded library construction technologies with STARsolo.

Input

- **Reference Genome**. Reference genome in FASTA format. Reads will be mapped against this reference.
- **Annotation GFF/GTF**. Reference annotation in GTF or GFF format. Counts will be counted by the genes present in this annotation. The chromosome names present in the annotation must match the ones present in the genome sequence headers.
- **Exon Feature**. Select the feature from the GFF/GTF file containing exons.
- **Overhang**. This parameter must correspond to the length of the sequenced transcript minus 1. That is, the length of the downstream read minus 1.
- **Single-cell RNA-Seq Reads**. Select the FASTQ files containing the scRNA-seq reads. If using reads coming from 10x Chromium, avoid specifying the index FASTQ, that is, the files containing I1 and I2. In case of having multiple FASTQ for the same sample (e.g. multiple sequencing lanes), specify all the fastq files in the same run. However, in the case of analyzing multiple samples or batches, one STARsolo run must be performed for each of them.
- **Upstream Files Pattern.** Pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern**. Pattern to recognize downstream FASTQ files.
- **Strandness**. Whether the sequencing is strand-specific (Forward or Reverse) or not (Unstranded).

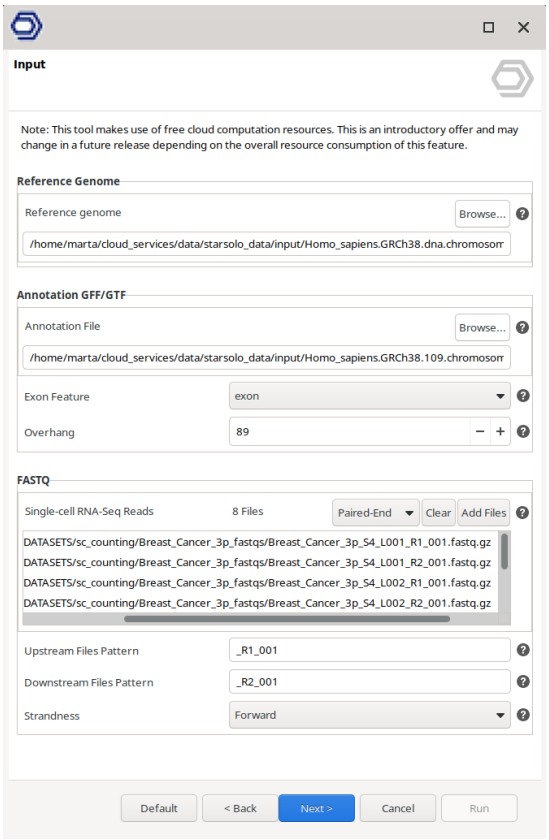


Figure 3. Cell-barcoded technologies input wizard page.

Configuration1: Reads Configuration.

- **Library Technology**. When something different than "Custom" is selected, the parameters regarding read structure are automatically updated to match the technology and version specified.
- **Barcode Mate**. In which read is the Cell Barcode and UMI located.
- Separate Read. The upstream read is composed entirely of the Cell Barcode and UMI. The downstream read contains only the transcript sequence.
- Part of Mate 1. The upstream read contains CB, UMI, and part of the transcript sequence. The downstream read is composed only of transcript sequence.
- Part of Mate 2. The downstream read contains CB, UMI, and part of the transcript sequence. The upstream read is composed only of transcript sequence.
- **Cell Barcode Start**. Starting base of the Cell Barcode in the barcoded read.
- **Cell Barcode Length**. Length in basepairs of the Cell Barcode.
- **UMI Start**. Starting base of the UMI in the barcoded read.
- **UMI Length**. Length in basepairs of the UMI.
- **Clip from 5' end**. In the case that the read(s) containing transcript sequence contain CB, UMI, or adapter sequence as well, they have to be clipped before aligning. Check this option in order to remove those sequences from the read. The bases will be removed from the read specified in the "Barcode Mate" parameter. One example is the 10x Chromium 5' protocol, in which the Mate 1 contains CB (16bp) + UMI (10bp) + adapter (13bp) which in total sum up to 39 bp. If Mate 1's sequencing extends beyond 39 base pairs, the additional bases correspond to cDNA and can be aligned with Mate 2, which exclusively contains cDNA.
- **5' Number of Bases**. Specify the number of bases to trim from the 5' end for each of the bases. The first number is for Mate 1 and the second is for Mate 2. They must be separated by a space.
- **Clip from 3' end**. Clip bases from the 3' end of the read(s). See the "Clip from 5' end" parameter above for a detailed explanation.
- **3' Number of Bases**. Specify the number of bases to trim from the 3' end for each of the bases. The first number is for Mate 1 and the second is for Mate 2. They must be separated by a space.
- **Add Cell Barcode Whitelist**. If checked, a file with the list of Cell Barcodes used by the library preparation technology must be provided. If not checked, Cell Barcodes will be automatically inferred from the data.
- **Cell Barcodes Whitelist**. Specify here the file containing the Cell Barcodes sequences.
- **Cell Barcode Match Type**. It is likely to find sequencing errors in the region of the read containing the Cell Barcode, which will produce differences between the Cell Barcodes detected in the data and the ones present in the whitelist. With this parameter, it is possible to choose how to match the sequences identified in the reads with the Cell Barcodes whitelist:
    - Exact Matches. Only Cell Barcodes with exact matches to the whitelist are kept.
    - 1MM. Cell Barcodes with only one match to the whitelist with a maximum of one mismatch are kept.
    - 1MM Multi. Cell Barcodes with multiple matches to the whitelist with one mismatch are allowed. However, allowed CBs have to have at least one read with an exact match. Then, a posterior probability calculation is used to choose one of the matches. This option matches best with CellRanger 2.2.0
    - 1MM Multi + Pseudocounts. It follows the same procedure as the 1MM Multi option, but pseudocounts of 1 are added to all whitelist barcodes.
    - 1MM Multi + Pseudocounts + N Bases. It follows the same procedure as the 1MM Multi option, but pseudocounts of 1 are added to all whitelist barcodes, and Cell Barcodes with N's are allowed. This option matches best with CellRanger >= 3.0.0.

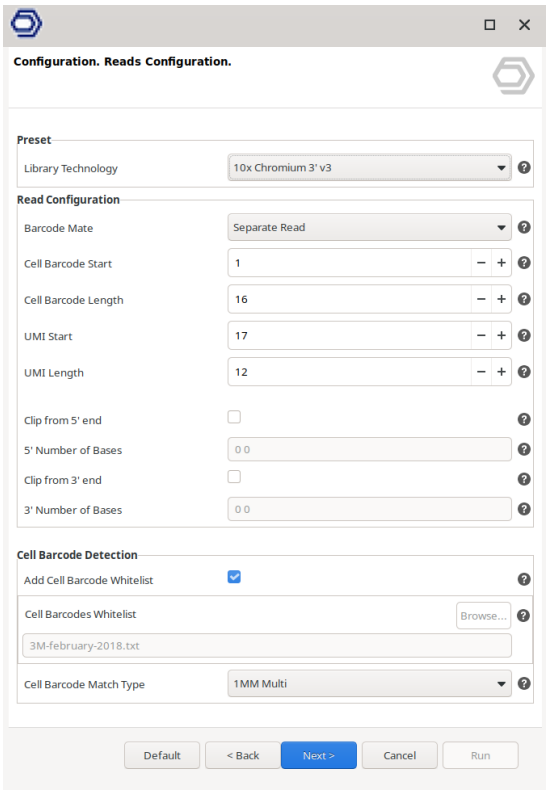


Figure 4. Cell Barcode Reads Configuration wizard page.

Configuration 2: Alignment Parameters

- **2-pass Mapping**. This option allows a most sensitive novel junction discovery. The aligner algorithm is executed first to collect the junctions. These junctions are used for second-pass mapping.
- **Min. Intron Length**. Specify the minimum intron size. A genomic gap is considered an intron if its length is equal to or greater than the given value. Otherwise, it is considered a deletion.
- **Max. Intron Length**. Specify the maximum intron size.
- **Max. # of Mismatches**. Set the maximum number of mismatches allowed per read pair.
- **Max. # of Multiple Alignments**. Establish the maximum number of multiple alignments allowed per read. If exceeded, the read is considered unmapped.
- **Include Chimeric Alignments**. This option allows including the chimeric alignments together with normal alignments in the main BAM file. The format of chimeric alignments follows the latest SAM/BAM specifications.
- **Max. Distance Between Mates**. Specify the maximum genomic distance between two mate pairs.

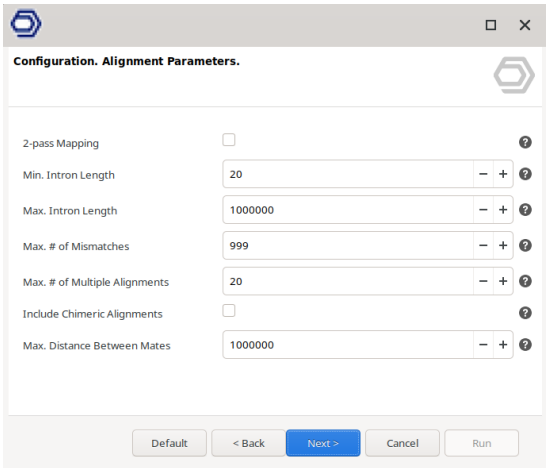


Figure 5. Alignment Parameters configuration wizard page.

Configuration 3: Counting Parameters

- **UMI Collapsing**. Algorithm for collapsing UMIs, that is, aggregate the counts of the equivalent UMIs. There are different approaches to trying to identify UMIs produced by sequencing errors:
- One Mismatch. UMIs that differ by only one mismatch are collapsed, meaning they are considered as the same one and thus counted together.
- UMI tools. Follows the "directional" method developed by Smith, Heger, and Sudbery (Genome Research 2017), first used in the UMI-tools package.
- Directional. Same as "UMI tools", but with more stringent criteria for duplicate UMIs.
- Exact. Only UMIs that are an exact match are counted together.
- No Deduplication. Do not collapse UMIs. This will produce read-level counts.
- CellRanger Algorithm. Algorithm performed by CellRanger v2.4 for UMI collapsing.
- **UMI Filtering**. How to filter poor-quality UMIs:
- Basic. Remove UMIs with N and homopolymers. This behavior is similar to the filtering performed in CellRanger 2.2.0.
- Multi Mapping UMIs. In addition to the basic filtering, this option removes lower-count UMIs that map to more than one gene.
- CellRanger > 3.0.0. The same filters as the above "Multi Mapping UMIs" option, but matching the behavior of CellRanger > 3.0.0 . It only works along with the "UMI Collapsing" parameter set to "CellRanger Algorithm".
- **Multimapping Reads**. How to handle the multi-mapping reads:
- Discard. Do not count UMIs/reads that map to more than one gene.
- Uniform. Distribute uniformly the UMI/read counts among all the mapping genes.
- Proportional. Distribute the multi-gene UMIs counts in proportion to the number of unique UMIs for each gene. UMIs mapped to genes without unique UMI support are distributed evenly.
- MLE-EM Model. Uses Maximum Likelihood Estimation (MLE) and Expectation-Maximization (EM) algorithms to distribute multi-mapping UMIs among the genes. This algorithm has previously been used in TEtranscripts, Alevin, and Kallisto-bustools tools.
- Rescue. Use the approach described by Mortazavi et. al.. It distributes the counts uniformly among the multi-mapping UMIs in a way proportionally to the sum of uniquely mapped UMIs.
- **Feature Counting**.
- Exon. For each gene, count only the reads aligning to the exonic regions.
- Exon + Intron. For each gene, count both the reads aligning to exonic and intronic regions.
- **Cell Filtering**. It is very common that the resulting count table has spurious cells, that is, Cell Barcodes that have been detected as such but they come from sequencing errors. There are different approaches to filtering detected cells:
- None. Do not filter resulting cells.
- Top Cells. Only report top cells by UMI count, followed by the exact number of cells
- CellRanger. Use the approach followed by CellRanger 2.2. It needs the expected number of cells.
- Empty Drops. Use the approach first developed by EmptyDrops filtering in CellRanger flavor. It needs the expected number of cells.

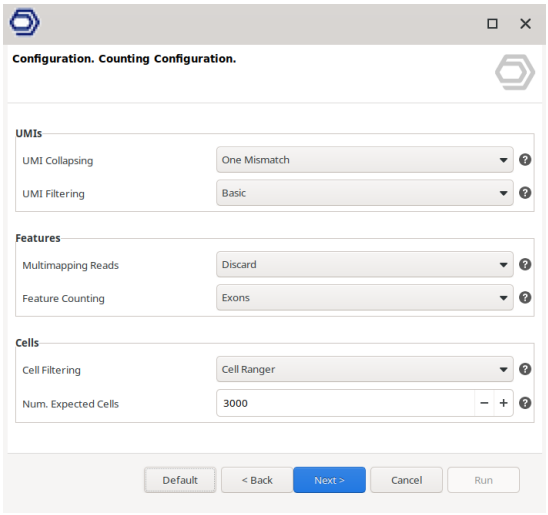


Figure 6. Cell Barcode counting configuration wizard page.

Output

- **Project Name**. Give a name to the project. The resulting count table will be named after it.
- **Save Raw Matrix**. The default output is the Coun Table obtained after the Cell Filterig. With this option checked, the prior Count Table containing all the detected cells is saved as well.
- **Matrix Folder.** Select the folder to save the unfiltered Count Table. It is saved in MTX format, thus, three files will be saved: one text file containing the counts (.mtx), one text file containing the cell barcodes (barcodes.tsv), and one text file containing the genes (features.tsv).
- **Save BAM File**. Check this option to save the BAM file generated during the read alignment step.
- **BAM File**. Specify the path to save the BAM file.
- **Sort by Coordinate**. Sort the output BAM file by coordinates. If not checked, the BAM file will be unsorted.
- **Add Read Group Information.** Include the 'Read Group' header (@RG) in output BAM files. This information may be required for downstream analysis or third-party tools. If this option is checked, the following read group tags will be included:
  - Identifier (ID), automatically generated.
  - Name of the sample (SM), inferred from file names.
  - Sequencing Platform (PL), provided in the "Sequencing Platform" parameter.
  - **Sequencing Platform**. Choose the sequencing platform that was used to obtain the input data.

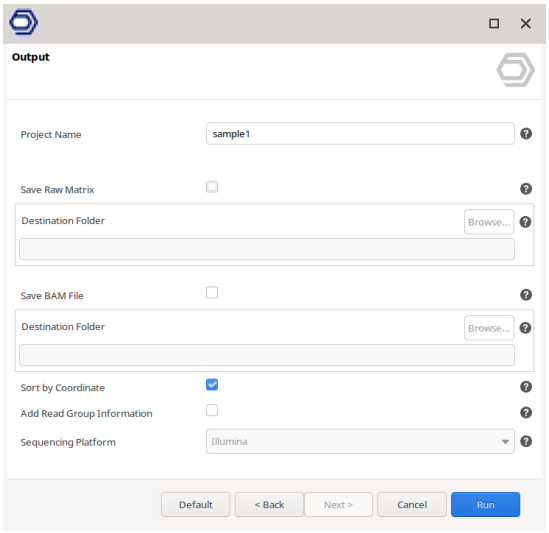


Figure 7. Cell Barcode output configuration wizard page.

Full-length Technologies Input

- **Reference Genome**. Reference genome in FASTA format. Reads will be mapped against this reference.
- **Annotation GFF/GTF**. Reference annotation in GTF or GFF format. Counts will be counted by the genes present in this annotation. The chromosome names present in the annotation must match the ones present in the genome sequence headers.
- **Exon Feature**. Select the feature from the GFF/GTF file containing exons.
- **Overhang**. This parameter must correspond to the length of the sequenced transcript minus 1.
- **Single-cell RNA-Seq Reads**. Select the FASTQ files containing the scRNA-seq reads. Each FASTQ file is supposed to belong to one individual cell. In the case of analyzing multiple samples or batches, one STARsolo run must be performed for each of them.
- **Upstream Files Pattern.** Pattern to recognize upstream FASTQ files.
- **Downstream Files Pattern**. Pattern to recognize downstream FASTQ files.
- **Strandness**. Whether the sequencing is strand-specific (Forward or Reverse) or not (Unstranded).
- **Provide Cell IDs.** If checked, it will use the cell IDs specified in the "FASTQ to Cell IDs" file to name the resulting cells. If not, the FASTQ file name is used for naming the cells.
- **FASTQ to Cell IDs**. Specify a file containing the FASTQ file names and the cell ID wanted for each of them, separated by a tab. One FASTQ file - cell ID pair per line.

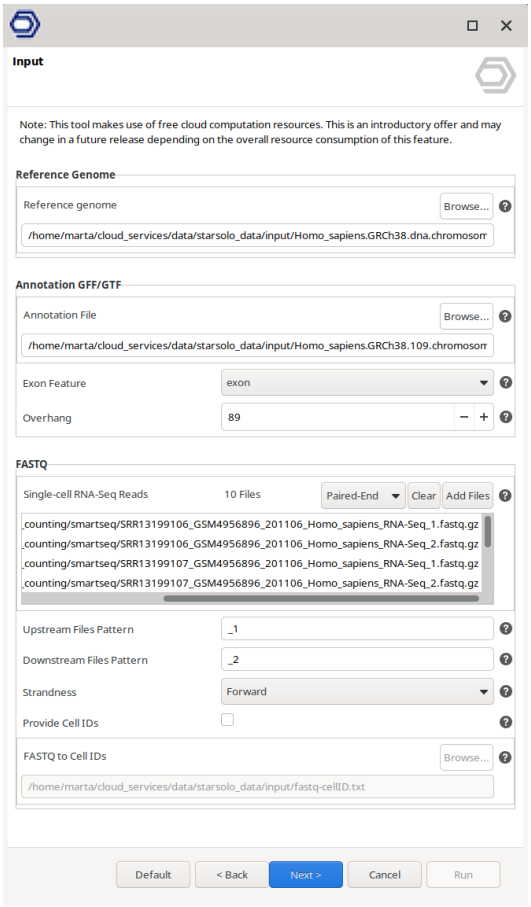


Figure 8. Full-length input wizard page.

Configuration: Alignment Parameters.

- **2-pass Mapping**. This option allows a most sensitive novel junction discovery. The aligner algorithm is executed first to collect the junctions. These junctions are used for second-pass mapping.
- **Min. Intron Length**. Specify the minimum intron size. A genomic gap is considered an intron if its length is equal to or greater than the given value. Otherwise, it is considered a deletion.
- **Max. Intron Length**. Specify the maximum intron size.
- **Max. # of Mismatches**. Set the maximum number of mismatches allowed per read pair.
- **Max. # of Multiple Alignments**. Establish the maximum number of multiple alignments allowed per read. If exceeded, the read is considered unmapped.
- **Include Chimeric Alignments**. This option allows including the chimeric alignments together with normal alignments in the main BAM file. The format of chimeric alignments follows the latest SAM/BAM specifications.
- **Max. Distance Between Mates**. Specify the maximum genomic distance between two mate pairs.

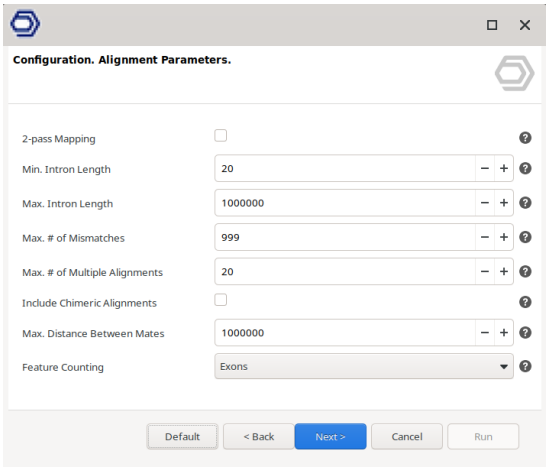


Figure 9. Alignment configuration wizard page.

Output

- **Project Name**. Give a name to the project. The resulting count table will be named after it.
- **Save BAM File**. Check this option to save the BAM file generated during the read alignment step.
- **BAM File**. Specify the path to save the BAM file.
- **Sort by Coordinate**. Sort the output BAM file by coordinates. If not checked, the BAM file will be unsorted.
- **Add Read Group Information.** Include the 'Read Group' header (@RG) in output BAM files. This information may be required for downstream analysis or third-party tools. If this option is checked, the following read group tags will be included:
    - Identifier (ID), automatically generated.
    - Name of the sample (SM), inferred from file names.
    - Sequencing Platform (PL), provided in the "Sequencing Platform" parameter.
    - **Sequencing Platform**. Choose the sequencing platform that was used to obtain the input data.

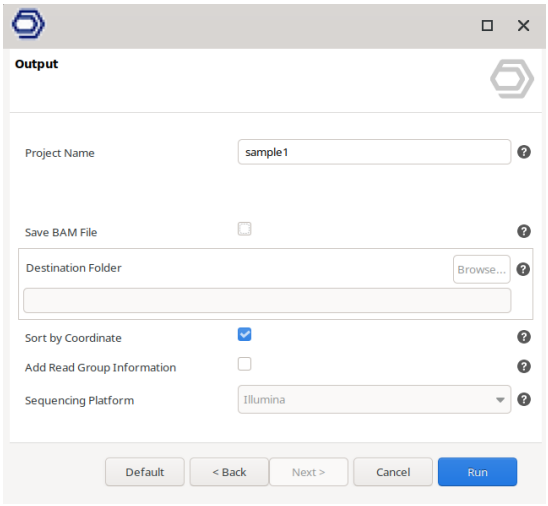


Figure 10. Full-length output configuration wizard page.

Results

The following outputs are produced.

- **Single-cell count matrix**. Two independent tables for cells on the left and features on the right side (Figure 11). Both tables can be ordered and filtered separately, while the number of visible rows is shown in the top-right corner.
- **Report** with a detailed summary of the quantification process and some common metrics.
- **UMIs per cell plot** that helps to judge the quality of the quantification process (Figure 12).
- This plot is inspired by the Barcode Rank Plot from 10x Genomics (https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/advanced/barcode-rank-plot)

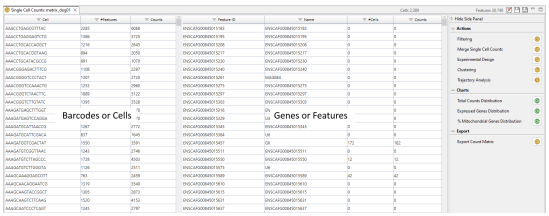


Figure 11. scRNA-Seq Count Matrix object main viewer.

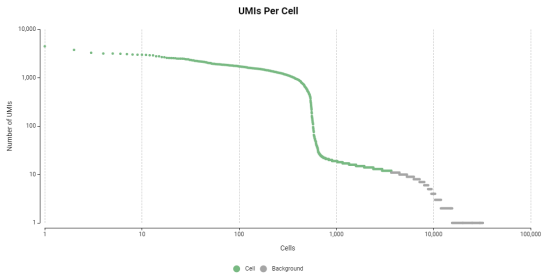


Figure 12. UMIs Per Cell plot output of scRNA-Seq Quantification tool.

SINGLE-CELL RNA-SEQ COUNT MATRIX

Main Table

The main table is divided into two independent tables for cells on the left and features on the right side (Figure 1). Both tables can be ordered and filtered separately, while the number of visible rows is shown in the top-right corner:

- Left side:
- **Cell**: The cell or barcode name.
- **#Features**: The number of features with counts (>0) for this cell.
- **Counts**: The sum of counts for this cell along all features.
- Right side:
- **Feature ID**: The unique ID of the feature or gene.
- **Name**: Names can be more descriptive, this column may contain duplicates.
- **#Cells**: The number of cells with counts (>0) for this feature.
- **Counts**: The sum of counts for this feature along all cells.

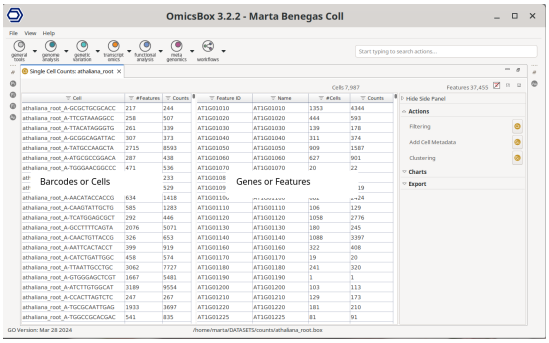


Figure 1. scRNA-Seq Count Matrix object main viewer.

Side Panel

The count matrix side panel shows all the available functionality for this type of data.

Actions

Click on the corresponding links to access the specific documentation for each analysis:

- **Filtering**. Remove low-quality cells from the count table.
- **Clustering with Seurat.** Group cells with similar gene expression patterns.
- **Trajectory Analysis with Monocle3.** Order cells in pseudotime.

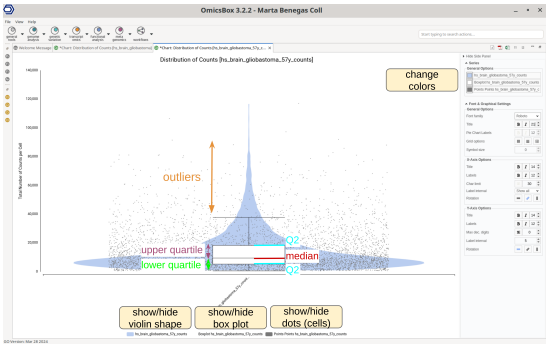


Figure 2. Parts of the violin plot.

Charts

Violin plots showing the distribution of different cells' statistics. Each dot represents a cell, the position on the Y-axis is the value of the given statistic, while the position on the X-axis is arbitrary. The width of the violin (background color) shows the density of dots on each region in the Y-axis. The box plot shows the median, the interquartile range, and the outliers (Figure 2).

You can choose a Grouping Factor (e.g., Sample), in which case a separate violin plot will be displayed for each group. This applies to all three distribution options:

- **Distribution of Counts**: Shows one violin plot per group with the total number of counts per cell (Figure 3). The total number of counts is computed by summing all the gene counts for a given cell.
- **Distribution of Expressed Genes**: Shows one violin plot per group with the total number of expressed genes per cell, that is, genes with an expression level > 0.
- **% Mitochondrial Genes Distribution**: Shows one violin plot per group with the percentage of expressed mitochondrial genes per cell. The percentage is computed taking into account the expressed genes. In order to do the calculation, a list of the mitochondrial genes has to be provided in a text file. The text file must not contain headers and one gene per line. The gene names or IDs have to be found whether in the "Feature ID" or "Name" columns (Figure 4).

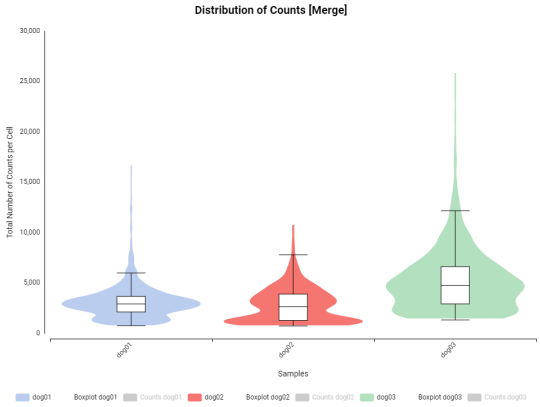


Figure 3. Violin and box plots show the distribution of the total number of counts per cell.

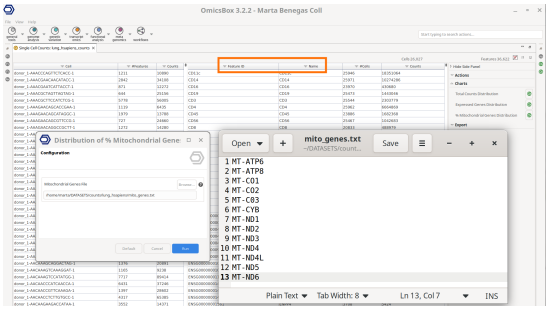


Figure 4. % Mitochondrial Genes Distribution wizard and example file.

Export

- **Export Count Matrix**:Export the count matrix in Matrix Market File format. Three locations must be specified in the wizard (Figure 5) in order to save the mtx (non-zero counts coordinates), barcodes, and features files.
- **Export Cell Metadata**. Generate a tab-delimited text file with cell metadata. The first line is the header and one line per cell. Select on the "Subgroup(s) to Extract" which metadata from the object should be included in the file (Figure 6).

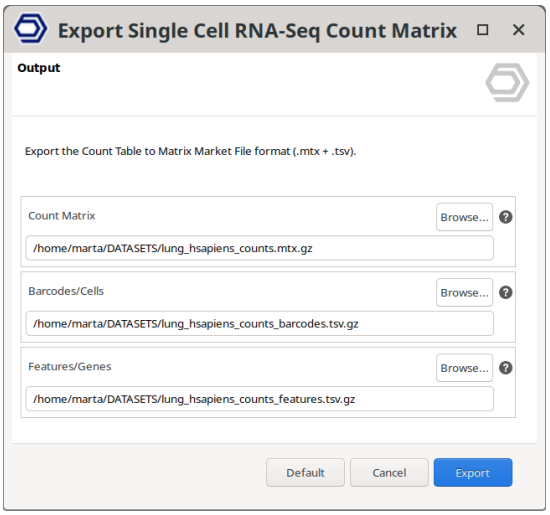


Figure 5. Export Count Matrix wizard.

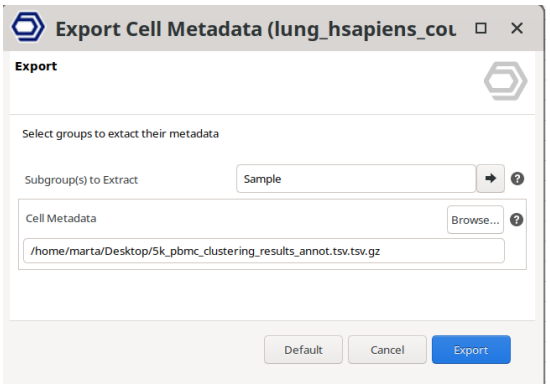


Figure 6. Export Cell Metadata wizard.

SINGLE-CELL RNA-SEQ FILTERING

Introduction

In addition to filtering low-quality reads like in bulk RNA-seq analysis, it is important to filter low-quality cells in Single-cell RNA-Seq (scRNA-Seq) datasets. This can be easily achieved by filtering cells based on their expression. This approach is based on the following assumptions:

- Cells with a comparatively **higher total number of counts** could correspond to doublets or multiplets, that is, groups of two or more cells that have been sequenced together.
- Cells with a comparatively **lower number of counts** may have resulted from sample preparation or sequencing artifacts. For example, broken cells that have lost part of their RNA content, cells for which the mRNA capture efficiency has been low, cells with a low PCR amplification during sequencing, etc. In any case, these cells in the dataset don't reflect their original mRNA content.
- Cells with a **high percentage** of total counts that correspond to **mitochondrial RNA** (mtRNA) could correspond to dying cells that began apoptotic processes. Additionally, this feature could be indicative of broken cells. This is because cells with broken cellular membranes could still conserve the mtRNA inside the mitochondrial membrane, so the percentage of mtRNA is greater compared to intact cells.
- Cells with a **low number of detected features** (genes) could have their mRNA content damaged. Sequencing reads obtained from this damaged mRNA can be different from the reference genome, which is used during quantification. This will ultimately cause a low mapping rate during quantification, causing a low number of genes to be detected.

Violin Plots are widely used by single-cell data analysts to determine cells meeting these characteristics (Figure 1). In this plot, each dot represents a cell, the position on the Y-axis is the value for the measured statistic, and the position on the X-axis is random. The shape of the violin plot represents the density of cells for a given value, thus it gives a general idea of the value distribution. Additionally, a bow plot is shown on the top of the violin. The line in the middle shows the median value, and the top and bottom lines of the box show the Q1 and Q3 of the distribution, respectively. The dots beyond the whiskers are considered outliers. Thus, in order to identify, for example, cells with a high number of counts in Figure 1, we could establish a threshold in the area where the violin narrows down. Or if we would like to be more restrictive, we could establish the threshold in the whiskers of the boxplot.

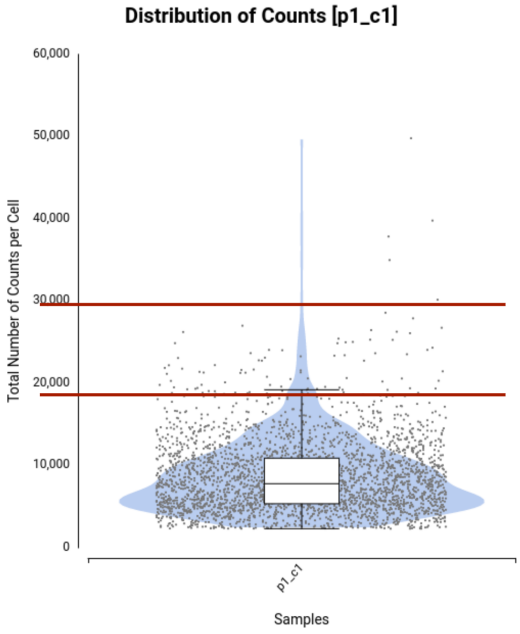


Figure 1. Violin plot showing the distribution of the total number of counts per cell in a count table. Red lines show suggested thresholds for cell filtering.

Run Single Cell RNA-Seq Quantification Input

In order to perform the Single Cell Filtering, a Count Matrix object has to be opened. It can be loaded from different formats by going to *transcriptomics → Load → Single Cell RNA-Seq Count Matrix* or generated from FASTQ sequencing files with the Single Cell RNA-Seq Quantification tool available in *transcriptomics → Single Cell RNA-Seq → Single Cell RNA-Seq Quantification.*

Once loaded, go to the *Side Panel* → *Actions* → *Filtering* (Figure 2).

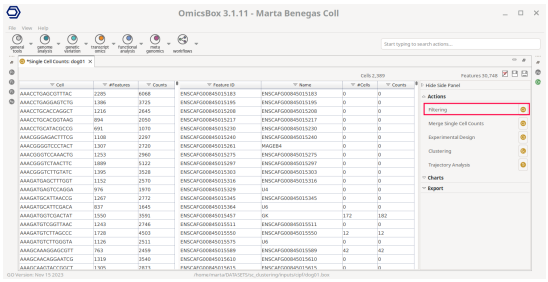


Figure 2. Launch filtering from a scRNA-Seq Count Matrix Side Panel.

Configuration

The following filters can be applied in the wizard (Figure 3). Default values are taken from the count table.

- **Minimum Cells**. Include features (genes, exons, etc.) detected in at least this number of cells. This filter is meant to exclude features that are not very informative. It removes rows from the count table.
- **Minimum Counts**. Discard cells with less than this number of reads/counts.
- **Maximum Counts**. Discard cells with more than this number of reads/counts.
- **Minimum Features**. Discard cells with less than this number of features.
- **Maximum Features**. Discard cells with more than this number of features.
- **Maximum % Mitochondrial Genes**. Discard cells with more than this percentage of mitochondrial genes.
- Mitochondrial Genes File. File with a list of mitochondrial genes, one per line. The gene IDs or names present in the mitochondrial genes file must correspond to the ones used in the count table.

Output

The output is a new scRNA-Seq object as shown in Figure 2 with cells filtered out.

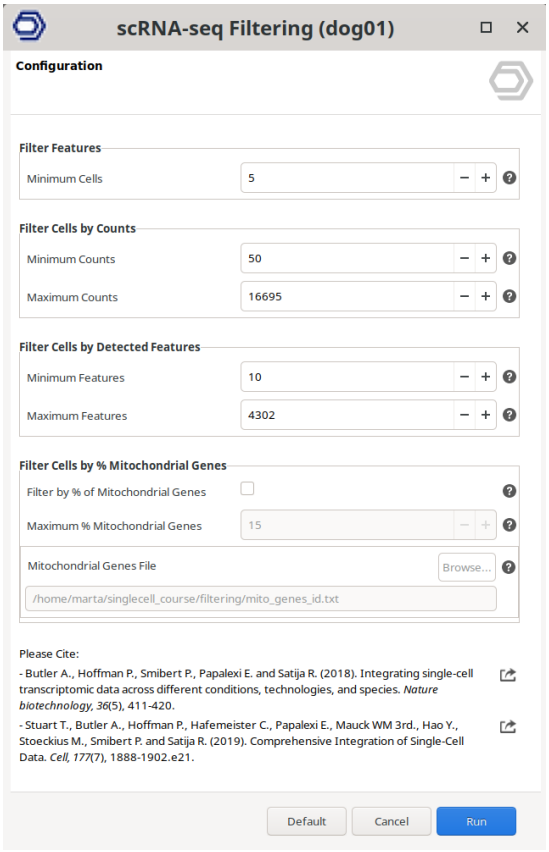


Figure 3. scRNA-Seq Filtering wizard.

Merge Single Cell Counts

This tool combines multiple matrices into a single count matrix for downstream analysis. The Merge Single Cell Counts function does **not** conduct **Data Integration**; rather, it consolidates multiple count tables into a unified dataset. This enables the specification of an experimental design, facilitating subsequent integration as an intermediate step in the clustering process. Additionally, it provides the capability to visualize violin plots of multiple samples within a single plot, as illustrated in Figure 1.

This feature is designed to enhance the ease of specifying experimental designs, and in consideration of the unique characteristics of library construction in droplet-based technologies such as 10x Chromium or Drop-seq. In these technologies, Cell Barcodes may be duplicated across samples, resulting in different cells being identified with the same Cell Barcodes. To address this, the Merge Single Cell Counts function **appends a distinct suffix to all cells** within a given count table, ensuring uniqueness across samples.

Input

This tool only accepts OmicsBox scRNA-Seq Count Matrices in .box format. They can be obtained with the scRNA-Seq Quantification tool or loaded into OmicsBox from external files.

- **Merge Single Cell Counts**: Combine count matrices into a single count matrix for downstream analysis such as clustering and trajectory analysis. On the opened wizard, select the count matrices in .box format to merge with the opened one (Figure 13). Cell barcodes will be unique for each sample (sequencing library) but features will be merged together whenever they match. See warning panel below

.

- **Experimental Design**: Assign factors (e.g. disease, age, sex, etc.) and assign a condition to each sample (count matrix). This makes sense after having merged count matrices together and can simplify the configuration of downstream analysis. On the opened wizard (Figure 14), press "Add Factor" to add a column and type the conditions on the given cells. Alternatively, press "Load Design" to specify the experimental design in a text file. Once specified, click on "Run".

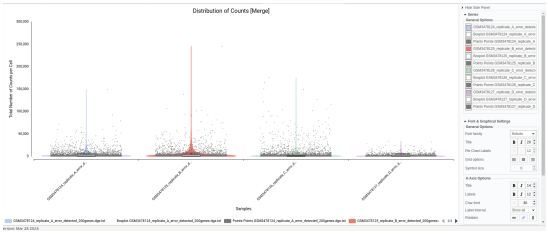


Figure 1. Violin plot showing multiple samples.

Run Merge Single-cell Counts

Go to *transcriptomics* → *Single Cell RNA-Seq* → *Merge Single Cell Counts*.

Input

Select all the count tables to merge by clicking on the "Add Files" button (Figure 2).

Configuration

Specify the experimental factors (if any) (Figure 3). In order to add a factor, click on the "Add Factor" button. Then, write the factor name on the box and the conditions in the new column, next to each sample. **Do not press enter**, click anywhere on the wizard once finished writing.

Once finished, click on "Run". The experimental factors will be stored as cell metadata.

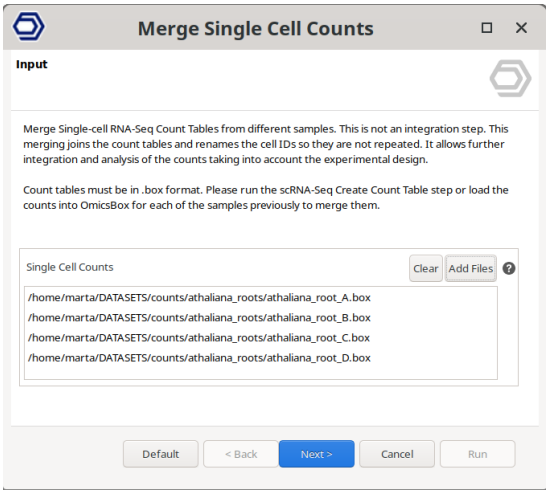


Figure 2. Input wizard page.

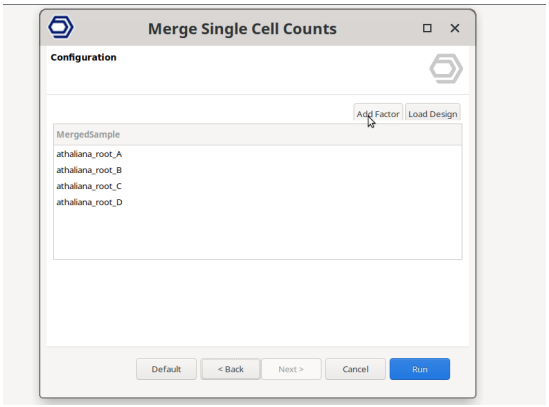


Figure 3. Specify the experimental factors for each count table.

Single Cell RNA-Seq Clustering

INTRODUCTION

This tool performs single-cell RNA-seq clustering with the widely-used Seurat package. It groups cells with similar expression profiles, which should correspond to the same cell type or state. Before clustering, the tool preprocesses the count matrix (normalization, feature selection, scaling) and reduces dimensionality (e.g., PCA) to denoise and speed up analysis (Figure 1). Afterward, cluster quality is evaluated with the bluster package, which computes statistics to assess the results.

Please cite Seurat and Bluster as:

Butler, A., Hoffman, P., Smibert, P. *et al.* Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* **36,** 411–420 (2018). https://doi.org/10.1038/nbt.4096

Stuart, T., Butler, A., Hoffman, P. *et.al.* Comprehensive Integration Of Single-Cell Data. *Cell* 177 (7): 1888-1902.e21 (2019). doi:10.1016/j.cell.2019.05.031.

Hao, Y., Stuart, T., Kowalski, M.H. *et al.* Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nat Biotechnol* **42**, 293–304 (2024). https://doi.org/10.1038/s41587-023-01767-y

Lun A (2025). bluster: Clustering Algorithms for Bioconductor. doi:10.18129/B9.bioc.bluster, R package version 1.18.0, https://bioconductor.org/packages/bluster.

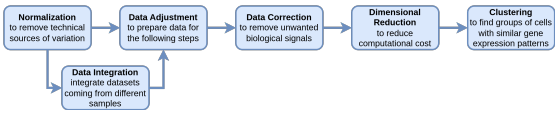


Figure 1. *General workflow for the single-cell RNA seq clustering analysis*

RUN SINGLE-CELL RNA SEQUENCING CLUSTERING

To perform the Single Cell Clustering, a Count Table object must be opened. It can be loaded from different formats by going to *transcriptomics → Load → Single Cell RNA-Seq Count Matrix*.

It can also be generated from FASTQ sequencing files with the Single Cell RNA-Seq Quantification tool available in *transcriptomics → Single Cell RNA-Seq → Single Cell RNA-Seq Quantification.*

Once the scRNA-Seq count table is loaded, go to the *Side Panel → Actions → Clustering* (Figure 2).

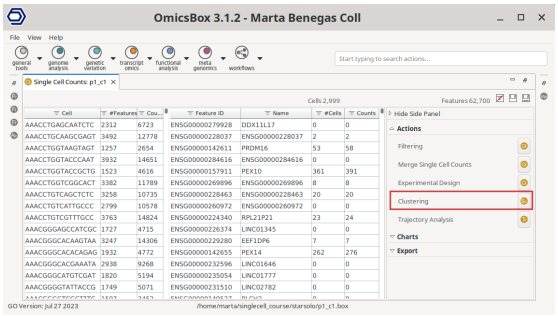


Figure 2*. Launch clustering analysis from a scRNA-Seq Count Matrix Side Panel.*

Configuration: Preprocessing.

These steps are meant to prepare data for the clustering analysis (Figure 3).

Normalization

Normalization aims to reduce differences in gene expression due to technical variation. This step tries to ensure that the observed heterogeneity within cells is due to biological reasons, rather than technical biases (Figure 3).

- **Normalize Data**. Check this option to perform the normalization step. Highly recommended unless your data is already normalized or you are certain that your data does not need it.
- **Normalization Method**. Which normalization method to use:
  - **Log Normalization**: Feature (gene) counts for each cell are divided by the total counts for that cell and multiplied by a scale factor. This is then natural-log transformed using log1p.
  - **Relative Counts**: Feature counts for each cell are divided by the total counts for that cell and multiplied by a scale factor. No log transformation is applied.
  - **Centered Log Ratio Transformation (CLR)**: For each feature, the CLR-transformation is defined as the logarithm of the feature counts for one cell, divided by the geometric mean of all counts for that cell.

Data Adjustment

These steps are necessary whether the normalization is done or not, in order to prepare data for the dimensional reduction step.

- **High Variable Genes**. Number of highly variable genes to keep for further analysis. Keeping only those genes reduces both computational cost and non-relevant signals.
- **Scale Data**. If checked, it scales features to have unit variance.
- **Center Data**. If checked, it centers features to have zero mean.

Data Correction

- **Regress Out Mitochondrial Genes**. Check this option to reduce the heterogeneity between cells associated with the expression of mitochondrial genes. This could prevent grouping cells during the clustering step that present a higher expression of those genes. In some cases, this information could not be informative since the higher expression of mitochondrial genes could be caused by technical reasons (e.g. because the mRNA has been leaked during cell manipulation) rather than by biological reasons. In order to perform this analysis, a file with a list of mitochondrial genes must be provided with one gene per line (Figure 4).
- **Regress Out Cell Cycle Genes**. Check this option to reduce the heterogeneity between cells associated with the expression of cell cycle genes. This could prevent grouping cells during the clustering step that are in the same developmental stage, independently of their type. In some cases, these differences in cell cycle may be uninformative but, in other cases, they could be treated as indicative of proliferating cell populations which can be different across treatment conditions, for example. So whether to check this option or not will depend on the dataset under study and the target of the experiment. In order to perform this analysis, a file with cell cycle genes in one column and the cell cycle phase they belong to (S or G2/M) must be provided. One gene per line and columns separated by a tab (Figure 5).

Dimensional Reduction

Common single-cell RNA-seq analysis involves hundreds to thousands of cells and tens of thousands of genes. That is, it is really high-dimensional data, so it is advisable to reduce the dimensionality of the dataset to reduce the computational cost of further analysis. The most widely used method for dimensional reduction is Principal Component Analysis (PCA). Keeping only the first Principal Components for further analysis reduces the dimensionality of the data while maintaining the

heterogeneity of the dataset, since they explain the largest amount of variance present in the sample. For more details please see "Orchestrating Single-Cell Analysis", 2020.

- **Principal Components**. Number of Principal Components to compute from the Principal Component Analysis. It must be smaller than the number of cells present in the count table.

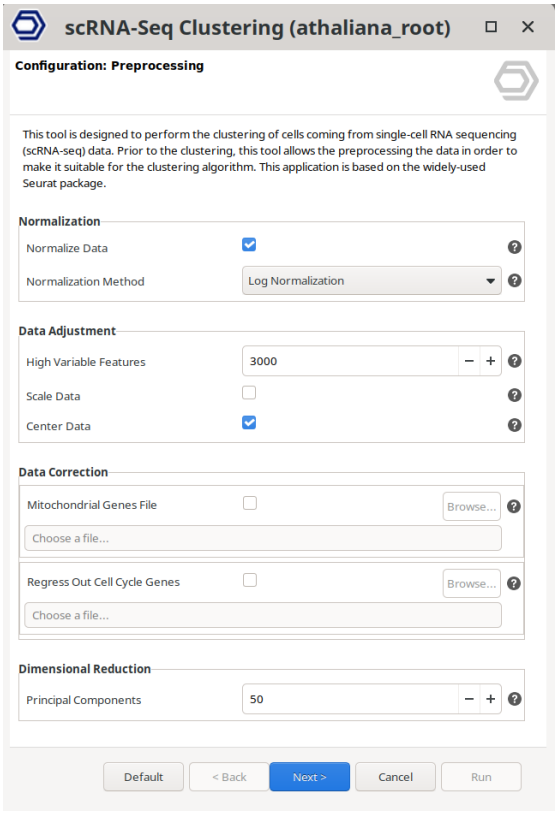


Figure 3. *Preprocessing wizard page.*

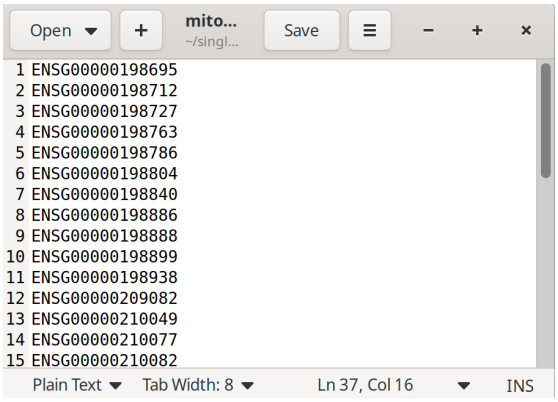


Figure 4. *Mitochondrial genes file example.*

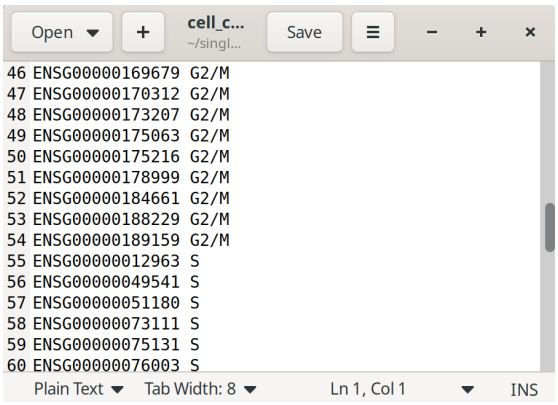


***Figure 5**. Cell cycle genes file example.*

Configuration: Multi-sample Data Integration.

This is an additional step needed when the input count table contains data from multiple samples, e.g., from wild-type and mutant organisms, from control and stimulated samples, etc. All these situations could cause changes in gene expression that could make a joint analysis of all the data difficult, with cells clustering both by condition and by cell type (Figure 6-A). In order to avoid that, the Multi-sample Data Integration step aims to integrate scRNA-seq datasets by identifying common cell types based on common sources of variation. As a result, this step enables the identification of shared populations across datasets (Figure 6-B and C) and thus further downstream comparative analyses.

- **Integration Factor**. Choose the condition to integrate datasets by. Datasets will be "integrated" by this condition so it doesn't interfere during the clustering of cells. If no integration is to be done, select "None".
- **Integration Method**. The statistical approach to use to integrate datasets.
    - **Harmony.** This method uses an iterative graph-based clustering algorithm to perform the integration. It performs a first soft clustering, favoring clusters with mixed dataset representation. Then, it gets the cluster centroids for each dataset and estimates a correction factor for each dataset and cluster. Finally, it applies the correction factor and moves the cells on each cluster accordingly. This procedure is repeated until the algorithm converges. For a detailed description of the algorithm, please see Korsunsky, I., Millard, N., Fan, J. et al. (2019).
        - **Theta**. Diversity clustering penalty parameter. Adjusting this parameter can influence the diversity of the clusters. Larger values of theta result in more diverse clusters.
        - **Lambda**. Ridge regression penalty parameter. It can affect the balance between underfitting and overfitting the model. Bigger values protect against over-correction.
        - **Tau**. Protection against over-clustering small datasets with large ones.
        - **Nº Clusters**. The number of clusters in the model. A value of 1 is equivalent to simple linear regression. This parameter can greatly influence the resolution of the data integration.
        - **Epsilon**. Convergence tolerance for Harmony. It determines when the algorithm should stop iterating, based on the improvement in the objective function.
    - **Seurat Integrations**. The Seurat algorithms are "anchor-based". These algorithms first embed all samples into a shared dimensional reduction space. Then, they find "anchors" between datasets, that is, equivalent cells across datasets. The anchors are identified with the mutual nearest neighbors (MNNs) algorithm. The different methods differ in the approach used to compute the dimensional reduction:
        - **Seurat-CCA.** The classic integration algorithm, which performs Canonical Correlation Analysis. It is recommended when the same cell types are expected in different datasets, but there are substantial gene expression differences. This integration is suitable when experimental conditions or disease states introduce very strong expression shifts, or when integrating datasets across modalities and species. For a detailed description of the method please visit Butler et al. 2018.
        - **Seurat-Joint PCA.** The dimensional reduction is one unique PCA (Principal Components Analysis) using all the datasets.
        - **Seurat-RPCA.** One PCA is computed for each dataset, and then each dataset is projected into the other's PCA. It is more conservative, faster, and doesn't tend to over-integrate the datasets as much as the CCA method. It is recommended when a high proportion of cells in one dataset have no matching type in the other datasets, when the datasets originate from the same platform (i.e. multiple lanes of 10x genomics), or when there are a large number of datasets or cells to integrate. For a more detailed description please visit Stuart et al. 2019.
        - All three methods use the same parameters for the "anchors" identification:
        - **N. Dimensions for Integration**. The number of dimensions to use from the dimensional reduction space for the integration step.
        - **K Anchor**. How many neighbors (k) to use during the MNNs when picking anchors. A higher number will result in a stronger integration.
        - **K Filter**. How many neighbors (k) to use during the MNNs when filtering anchors.
        - **K Score**. How many neighbors (k) to use during the MNNs when scoring anchors.
        - **K Weight**. How many neighbors (k) to use during the MNNs when weighing anchors.

The Cell Metadata needed to perform integration can be specified while Merging Single-cell RNA-Seq counts or by Adding Cell Metadata.

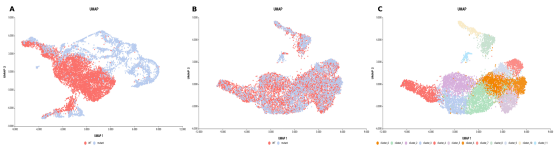


Figure 6. *UMAP plots of cells from wild type (WT) and Rb mutant Drosophila eye disc, prior to **(A)** and post **(B)** alignment. After alignment, cells across conditions are grouped together based on shared cell type, allowing for a single joint clustering **(C)** to detect 11 populations.*

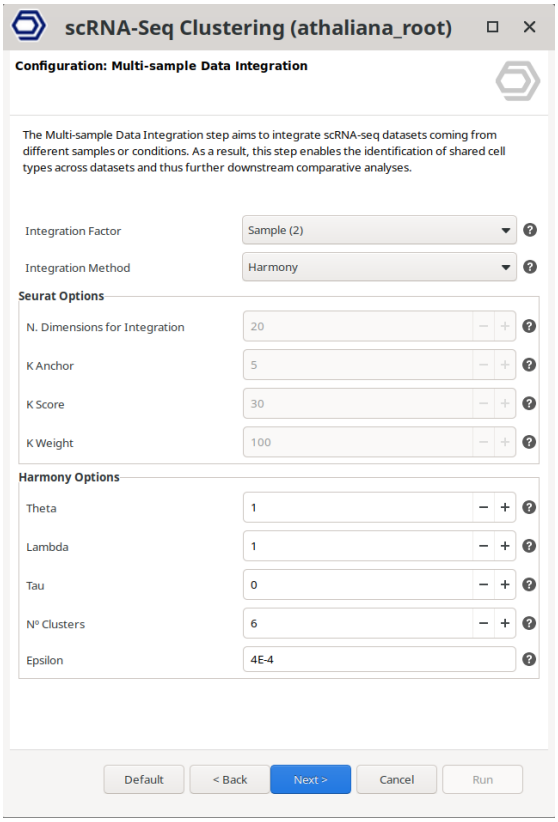


Figure 7*. Multi-sample Data Integration configuration wizard page.*

Configuration: Clustering.

This step performs the actual cell clustering. It groups cells with similar expression patterns, which should correspond to the same cell type. For this analysis, Seurat uses a graph-based clustering algorithm. For more information, please see "Orchestrating Single-Cell Analysis", 2020.

Clustering

These parameters affect the clustering algorithm (Figure 8).

- **Define Dimensions by**. How to decide the number of dimensions to use from the dimensional reduction (PCA) for the clustering step.
  - **Elbow Point**. This option automatically decides the number of dimensions to use. The assumption is that each of the top PCs capturing biological signals should explain much more variance than the remaining PCs. Thus, there should be a sharp drop in the percentage of variance explained from the last "biological" PC to the others. This is the so-called "Elbow Point".
  - **Manual**. With this option, it is necessary that the user specifies the number of PCs to use in the "Number of Dimensions" parameter.

Elbow Point Consideration

It should be taken into account that strong biological variation in the early PCs will shift the elbow point, potentially excluding weaker (but still interesting) variation in the next PCs immediately following it. So it could be interesting to repeat the analysis increasing the number of dimensions established by the Elbow Point to see if it improves the results.

- **Number of Dimensions**. The number of dimensions to use from the dimensional reduction for the clustering step. This option is only available if the "Manual" option from the "Define Dimensions by" is selected.
- **k-value**. The number of neighboring cells computed during the clustering algorithm. Normally, a greater k-value would produce a smaller number of clusters, and vice versa.
- **Resolution**. This parameter determines how "fine" the clustering is: values above 1.0 would produce a larger number of clusters, and vice versa.

UMAP Configuration

These parameters affect only the UMAP visualization. The Uniform Manifold Approximation and Projection (UMAP) method is a non-linear dimensionality reduction technique (similar to PCA), but this time it is used to plot the high-dimensional single-cell data into two dimensions. The UMAP is used for visualization purposes because it represents the variability of single-cell data more accurately than the PCA. For more information, please refer to "Orchestrating Single-Cell Analysis", 2020.

- **Point's Minimum Distance**. This controls how tightly the points (cells) are compressed together.
- **Point's Spread**. This controls how expanded the points are. In combination with the minimum distance, this determines how clustered the points are.

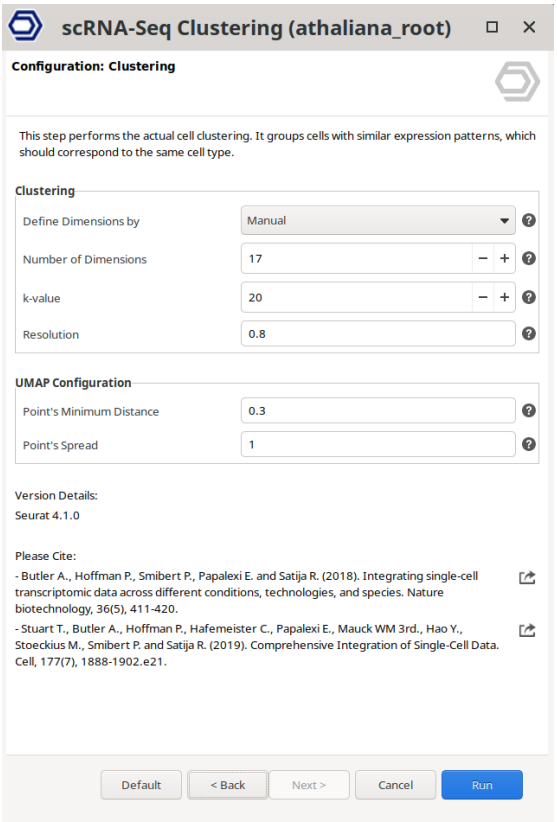


***Figure 8**. Clustering configuration wizard page.*

RESULTS

Once the input counts have been processed and analyzed via the "scRNA-seq Clustering" tool, a new tab is opened containing the Clustering Results (Figure 9). This new tab contains the same information as the count table and an additional column with the cluster assigned.

It also generates a Summary Report (Figure 9) with the following sections:

- The "**Data Overview**" table shows some general statistics about the data.
- The "**Clustering Results**" table shows, for each cluster, the total number of cells, the mean silhouette, the mean purity score and the RMSD. The button on the last column opens a list with the cell IDs of the given cluster. Before the table, there is a line specifying the number of dimensions from the PCA used for the clustering step.
- The "**Analysis Parameters**" table shows the parameters used for the clustering analysis.

Clustering Assessment Statistics

Detailed descriptions of the silhouette, purity score, and RMSD statistics are provided in their corresponding chart sections below. Silhouette and purity scores are computed per cell, while RMSD is computed per cluster. The "Data Overview" section shows overall means across all cells (silhouette/purity) or all clusters (RMSD), while the "Clustering Results" section shows means for cells within each specific cluster.

In addition, an interactive UMAP/tSNE viewer is also opened (Figure 10). This viewer allows coloring the cells by clusters, cell metadata, custom groups, or gene expression. In addition, cells can be selected with different tools and assigned to custom groups.

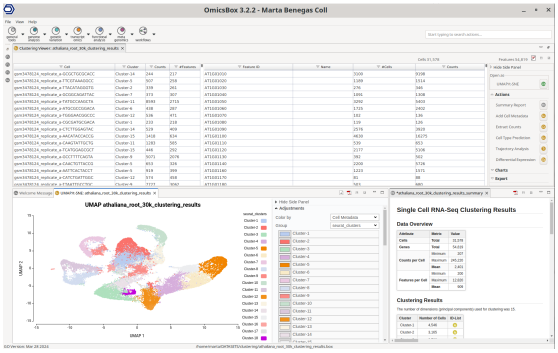


***Figure 9**. Clustering Results main viewer.*

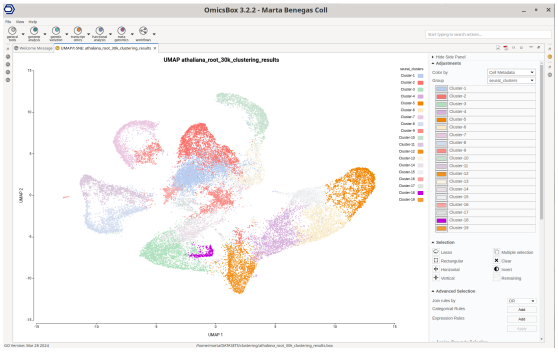


***Figure 10**. A detailed view of the interactive UMAP/tSNE viewer.*

Side Panel Features

Summary Report

It shows the Summary Report previously explained in the above "Results" section (Figure 9).

Add Cell Metadata

Add per-cell information (e.g., cell type annotations, experimental conditions, etc.) from a text file. The annotations will be stored in the object and can be used for further analysis. For example, they can be visualized in the UMAP/tSNE viewer, used for differential expression analysis or for trajectory analysis, etc. It will open a

wizard (Figure 11) to specify the text file and which group(s) to import. Columns must be tab-separated and the groups will be read from the first line in the text file (Figure 12).

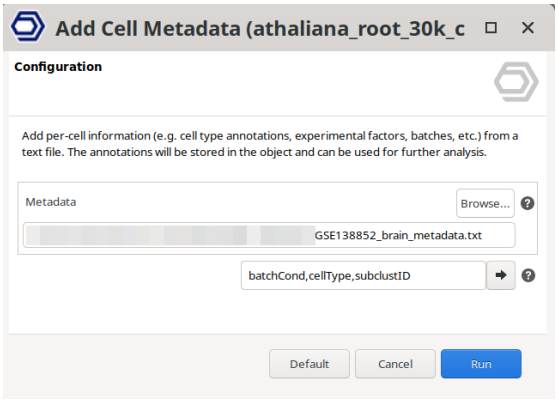


Figure 11. *Add Cell Metadata wizard.*

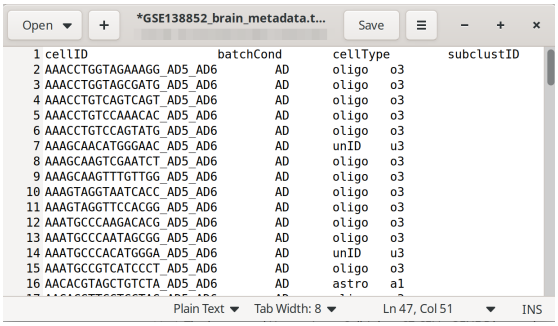


Figure 12*. Example of a text cell metadata file. Columns are separated by tab and the first line contains the column names.*

Extract Counts

Extract the counts for cells belonging to the selected subgroup(s) (Figure 13).

For example, it may be interesting in the scenario in which one or more big clusters have been obtained, so it could be desirable to extract them and re-run the clustering to obtain sub-clusters of cells within it. That could make it possible to find more specific cell types.

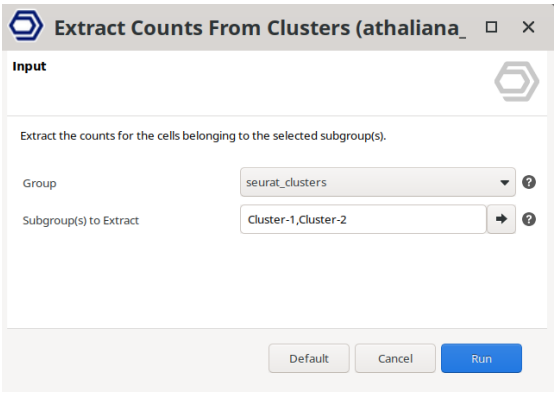


Figure 13*. Extract counts wizard.*

Cell Type Prediction

Perform Cell Type Prediction with SingleR.

Trajectory Analysis

Order cells along pseudotime using Monocle3.

Differential Expression

Perform differential expression analysis between two or more clusters, cell types, conditions, etc.

Charts Clustering Assessment

Shows quality metrics for the clustering results through different plots (Figure 14):

- **Silhouette Score**: Measures how similar a cell is to its own cluster compared to other clusters. Scores range from -1 to 1:
  - Higher values (~1) indicate cells well-matched to their cluster
  - Values near 0 suggest cells could belong to multiple clusters
  - Negative values suggest potential misclassification The distribution of scores is shown as violin plots for each cluster. In addition, a box plot is drawn on the top of the violin plot showing the median, the interquartile range, and the outliers.
- **Cluster Purity**: Purity measures how homogeneous the neighborhoods within a cluster are. For each cell, it calculates the proportion of its nearest neighbors that belong to the same cluster. Values range from 0 to 1:
  - Higher values indicate more homogeneous clusters
  - Lower values suggest more heterogeneous clusters Displayed as violin plots to show the distribution within each cluster, with a box plot drawn on the top showing the median, the interquartile range, and the outliers.
- **Root Mean Square Deviation (RMSD)**: Measures the average distance between cells in a cluster to their cluster center. Unlike silhouette and purity scores:
  - Lower values indicate more compact, well-defined clusters
  - Higher values suggest more dispersed clusters Shown as a bar chart since it's calculated per cluster rather than per cell.

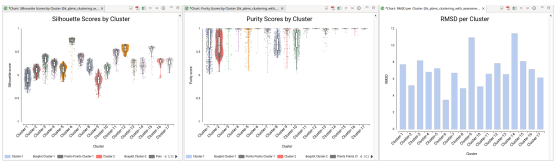


Figure 14. *Clustering assessment plots showing silhouette scores (left), cluster purity (middle) and RMSD (right).*

Expression Profile

With this feature, it is possible to see the expression levels of known gene markers across the different clusters in order to identify putative cell types. To this end, a Bubble Plot is generated with clusters in rows and the specified genes in columns (Figure 15). The size of the dot represents the percentage of cells expressing the gene, that is, the percentage of cells that have a gene expression level greater than 0. The color represents the average gene expression in that cluster. You can configure the following options in the wizard (Figure 16):

Input genes.

- **Input Genes**. You can specify here which genes to plot. The gene name or ID should correspond to the one in the input count table used during the clustering analysis. They can be specified in two ways:
- **Text**: write the genes to plot in the "Genes List" text box, one per line.
- **File**: specify a file containing the genes to plot, one gene per line.

Plot Options.

This affects the visualization.

- **Scale Gene Expression**. When checked, it applies the Z-Score transformation to scale average gene expression across clusters. It allows the visualization of both highly and lowly expressed genes on the same color scale. It should be noted that this may exaggerate the results, but it is still advisable if you are going to plot genes with different expression level ranges. If unchecked, it colors the dots by the raw average gene expression of each cluster.

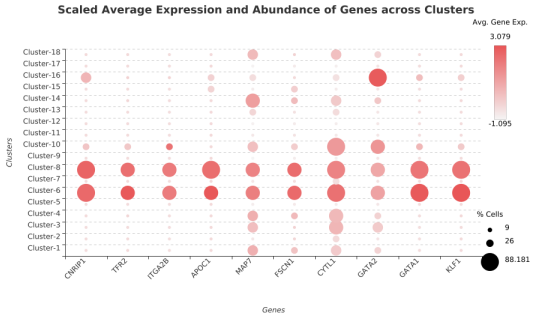


***Figure 16**. Expression Profile example.*

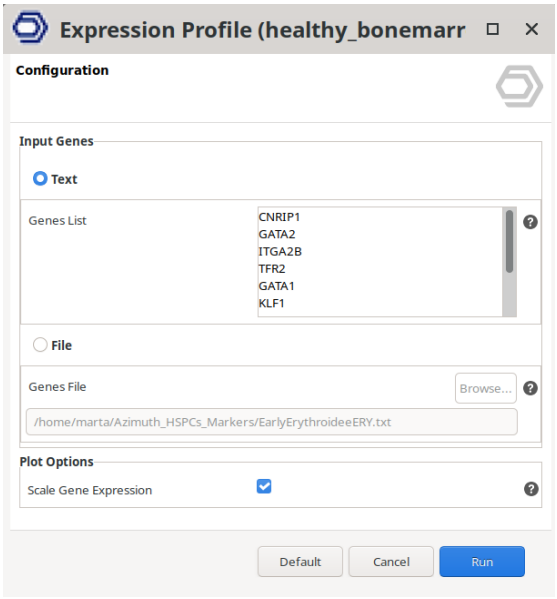


***Figure 17**. Expression Profile Wizard.*

Metadata Pie Chart

Generates a pie chart with the number of cells in each category (Figure 17).

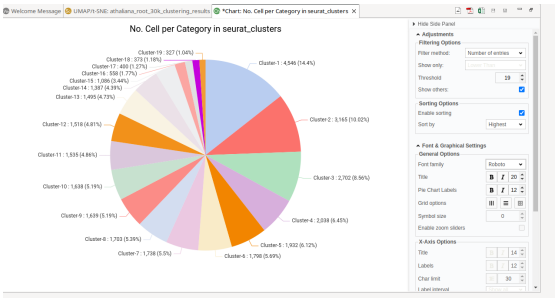


***Figure 17.** Seurat clusters pie chart.*

Elbow Plot

Elbow Plot (Figure 18) shows the amount of variance (given by the standard deviation) explained by successive PCs. This helps to decide how many principal components to use for the clustering algorithm, in case you want to re-run the clustering with different parameters.

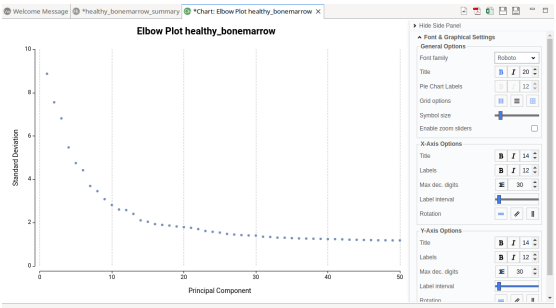


Figure 18. *Elbow Plot example.*

Export Export Cell Metadata

It generates a file containing one row per cell, containing the Cell ID, the cluster label assigned, and the sample of origin (Figure 19). Columns are separated by a tab.

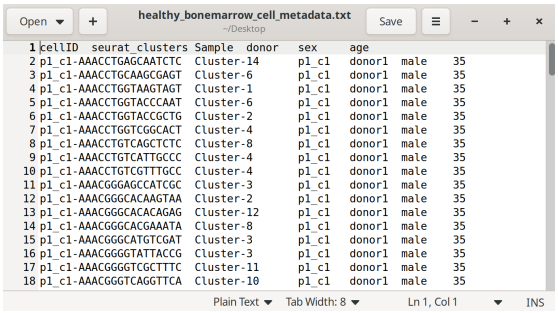


Figure 19. *Cell Metadata file.*

Export to AnnData

Export the clustering results to an h5ad file in AnnData format. This file contains the raw counts, cell metadata, and dimensional reductions (UMAP, t-SNE, and PCA).

AnnData format specifications can be found at: https://anndata.readthedocs.io/en/latest/fileformat-prose.html

Single Cell UMAP/tSNE Viewer

The UMAP/tSNE viewer displays the cells in a dimensional reduction space (UMAP or tSNE). These coordinates are obtained during Clustering or Trajectory analysis. The cells (dots) can be colored by different features. Moreover, they can be selected and added to new custom groups (Figure 1).

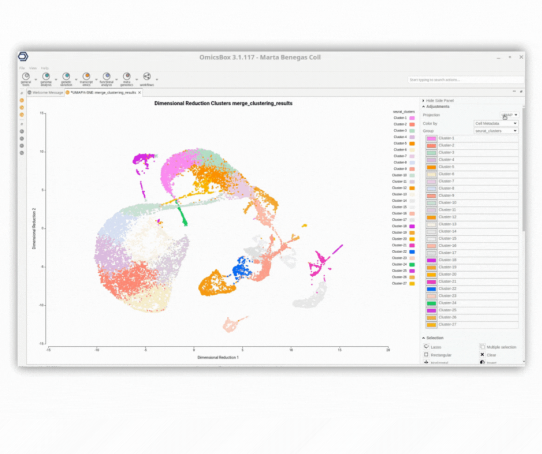


Figure 1. Interactive UMAP/tSNE viewer.

Adjustments

This section modifies the general appearance of the plot, that is, the coordinates and the color of the cells.

- **Projection**. Select which dimensional reduction space to plot: UMAP or tSNE (Figure 2). This will change the coordinates of the cells.
- **Color by**. The type of feature to color cells by (Figure 3). Depending on the category chosen, the configuration below will change.
- Cell Metadata. Color cells by the output of the Clustering or Trajectory analyses. These cell metadata groups can't be modified.
- My Classifications. Color cells by the output of Cell Type Annotation or custom groups. These cell metadata groups can be modified.
- Gene Expression. Color cells by the expression of one or multiple genes.
- Pseudotime. Color cells by pseudotime values. This is only available after Trajectory analysis.

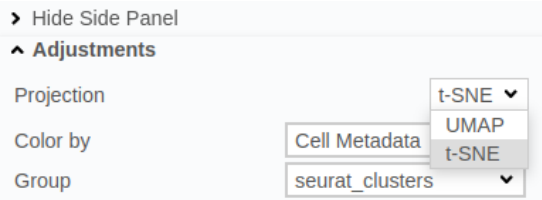


Figure 2. Select plotting the coordinates of the UMAP or tSNE in the "Projection".

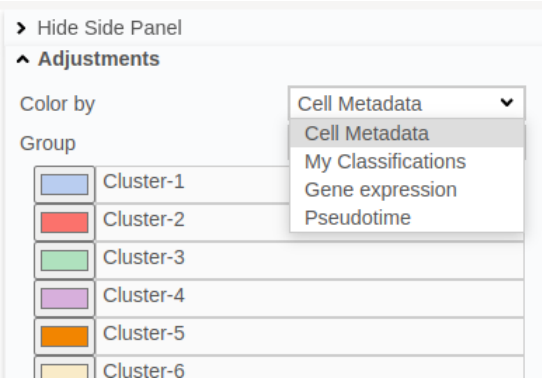


Figure 3. Different options of the "Color by" parameter.

Cell Metadata

This category includes groups obtained by Clustering or Trajectory results (Figure 4-A). Clustering results include the "seurat_clusters" groups. Trajectory results include "monocle_clusters", "monocle_partitions", and "ptime_ranges" groups. The levels appearing on the Side Panel and the plot's legend will change according to the selected group (Figure 5).

The color of the cells belonging to those groups can be modified by clicking on the corresponding color square and selecting a new color (Figure 4-B).

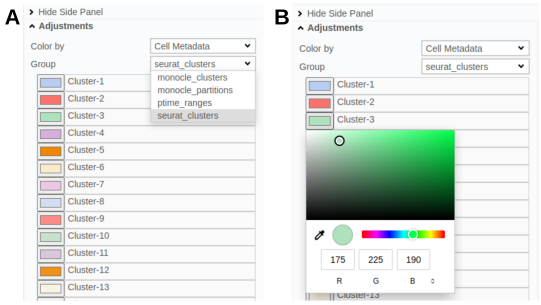


Figure 4. Cell Metadata groups (A) and how to modify the color of a group (B).

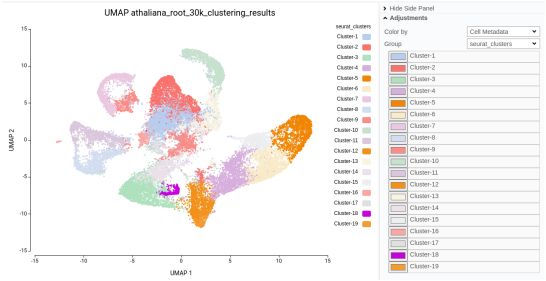


Figure 5. UMAP with cells (dots) colored by the "seurat_clusters" group in the Cell Metadata category.

My Classifications

This category includes groups generated after Cell Type Prediction or by the "Assign Group to Selection" tool (explained below) (Figure 6). The characteristic of this category is that both groups and levels can be renamed or deleted (Figure 7).

Cells will be colored attending to the label assigned in the selected Group. The color of a given label can be modified by clicking on the corresponding color square and selecting a new color (Figure 4-B).

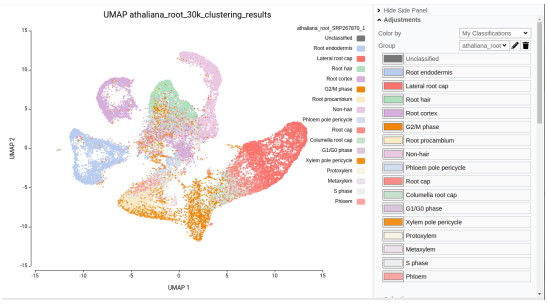


Figure 6. UMAP with cells (dots) colored by results of the Cell Type Prediction tool.

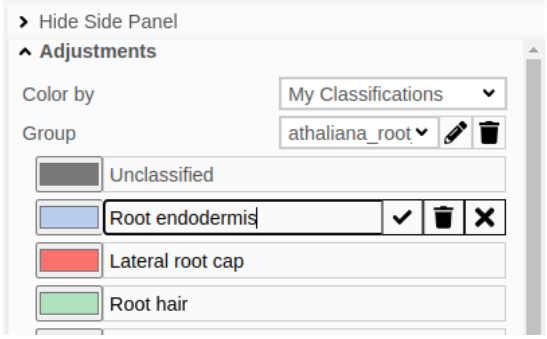


Figure 7. Modification buttons for the group and level names for the My Classifications category.

Gene Expression

This category will change the display of the section below. Cells will be colored according to the expression of the specified gene(s) (Figure 8).

Genes to plot can be specified on the "Gene" box (Figure 9-A). Gene name suggestions will appear while typing. Alternatively, a gene list can be pasted on the box.

More than one gene can be specified while plotting. In this case, it is possible to select how to summarize the expression values for each cell: by computing the average, the sum of all expressions, taking the minimum value, or taking the maximum value (Figure 9-B).

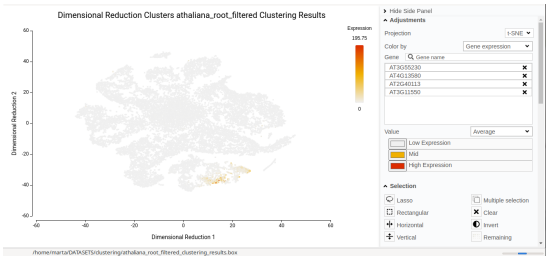


Figure 8. tSNE visualization with cells colored by the average expression of the specified genes.

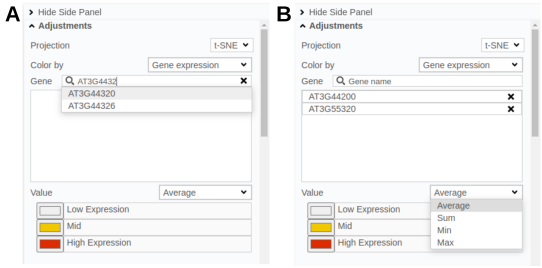
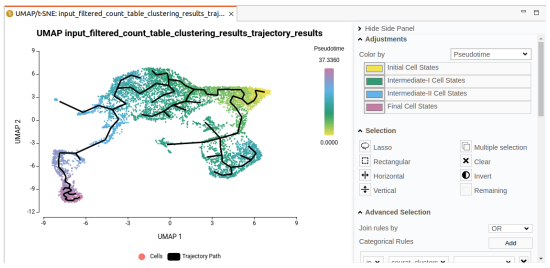


Figure 9. Color cells by Gene Expression category. A - Feature name suggestions appear while typing. B - Statistics to group gene expression levels for an individual cell.

Pseudotime

This category will change the display of the section below. Cells will be colored according to their pseudotime value (Figure 10). This option will be available only after Trajectory Analysis.

The color of each state (Initial, Intermediate I, Intermediate II, and Final) can be modified by clicking on the corresponding color square and selecting a new color.



Selection

This section allows selecting cells directly on the UMAP/tSNE. The selection tools are available independently of which category is selected in the "Adjustments" section.

Once the selection is done, the cells can be assigned to a new group. There are different options for performing the selection:

- **Lasso**. It allows drawing free-form shapes to select and highlight specific points or areas of interest (Figure 11).

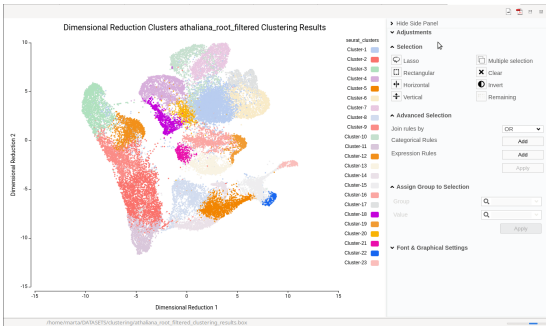


Figure 11. Example of the "lasso" tool for selecting areas of the UMAP/tSNE viewer.

- **Rectangular**. Draw a rectangular shape to select an area of interest (Figure 12).

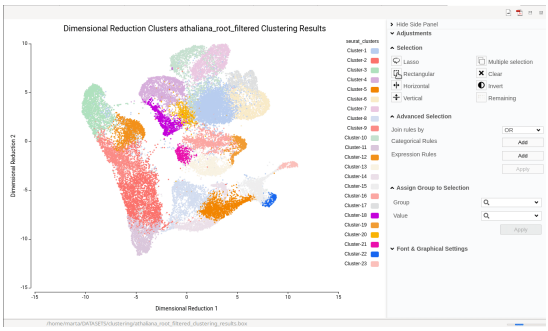


Figure 12. Example of the "rectangular" tool for selecting areas of the UMAP/tSNE viewer.

- **Horizontal**. Select cells based on the horizontal axis, that is, by the values on the X-axis (Figure 13).

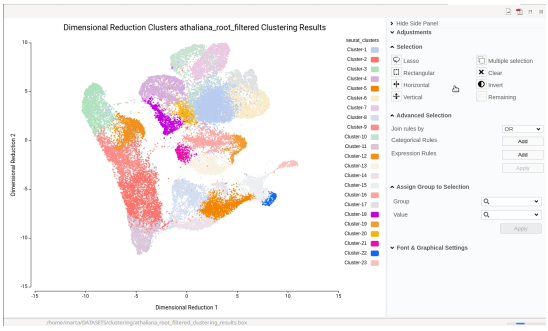


Figure 13. Example of the "horizontal" tool for selecting areas of the UMAP/tSNE viewer.

- **Vertical**. Select cells based on the horizontal axis, that is, by the values on the X-axis (Figure 14).

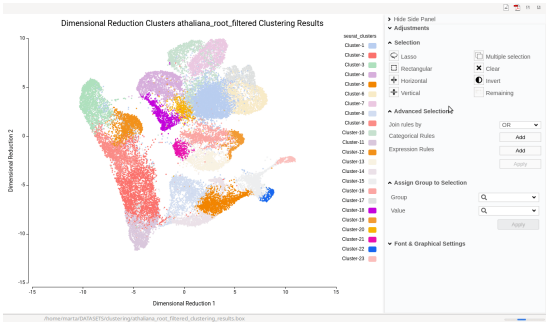


Figure 14. Example of the "vertical" tool for selecting areas of the UMAP/tSNE viewer.

- **Multiple selection.** Clicking this button allows selecting multiple areas of the plot. If this option is not activated, a new selection will overwrite the previous one (Figure 15).

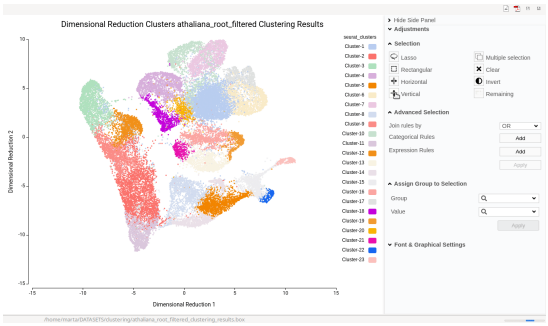


Figure 15. Example behavior of the "multiple" tool in the UMAP/tSNE viewer.

- **Clear**. This button removes the actual selection.
- **Invert**. This button reverses the selection process, deselecting the points or areas previously selected and selecting all other points or areas that were not initially included (Figure 16).

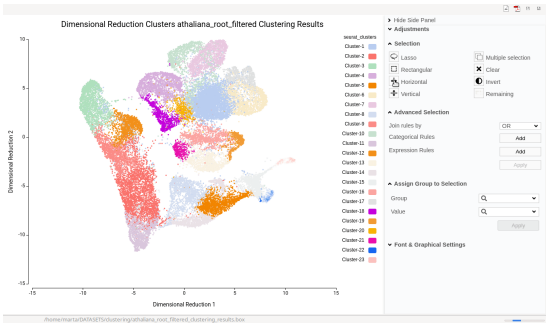


Figure 16. Example behavior of the "invert" tool in the UMAP/tSNE viewer.

- **Remaining**. If a group with unclassified cells is displayed in the "Adjustment" section, clicking this button selects all cells that have not been classified (Figure 17).

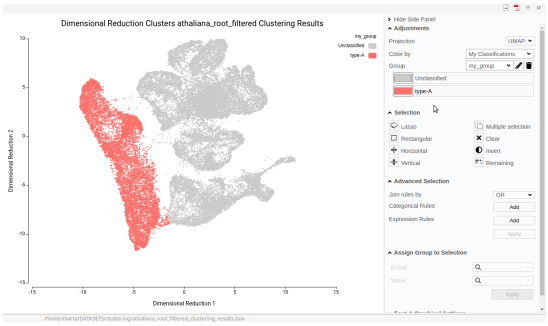


Figure 17. Example behavior of the "remaining" tool in the UMAP/tSNE viewer.

Advanced Selection

This section allows specifying rules for selecting the cells, instead of selecting them directly on the plot. It is possible to apply both categorical (based on group identity) and/or based on gene expression levels (Figure 18).

One or several categorical and/or expression rules can be applied at the same time. In this case, the "Join rules by" parameter controls if the rules will be joined with AND or OR.

Click on the buttons "Add" next to the "Categorical Rules" and "Expression Rules" parameters to specify a new rule.

For adding a categorical rule, select if the cells must be IN or NOT IN the group and label selected in the following dropdown lists.

For adding an expression rule, type a gene name, select a criterion (greater than >, equal to =, not equal to !=, etc.), and type the threshold value.

After all the rules have been specified, click on **Apply** to make the selection. If a new rule is added or removed, the Apply button must be clicked again.

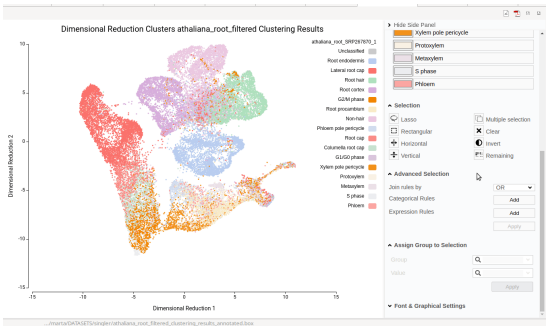


Figure 18. Example behavior of the "Advance Selection" section. Cells belonging to the "Root endodermis" group and with an AT3G55230 expression level greater than 1 are selected.

Assign Group to Selection

Once cells have been selected with the tools explained above, they can be assigned to custom groups in the "Assign Group to Selection" section (Figure 19).

The selection can be added to a preexisting group/value or a new group and/or value can be created by typing on the corresponding box.

The new group and value will be available in the "My Classifications" category in the "Adjustments" section.

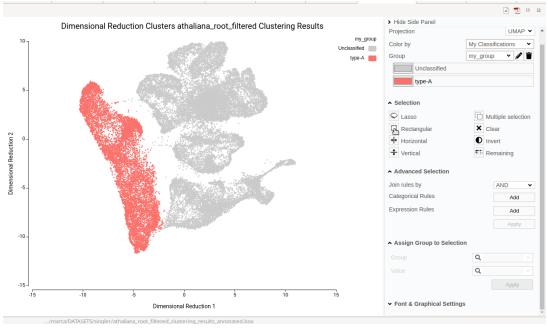


Figure 19. Example of how to add selected cells to custom groups.

Font & Graphical Settings

This section controls the general settings of the plot appearance. E.g: font family and size for each axis and title, the size of the dots, wether to include grids or not, etc.

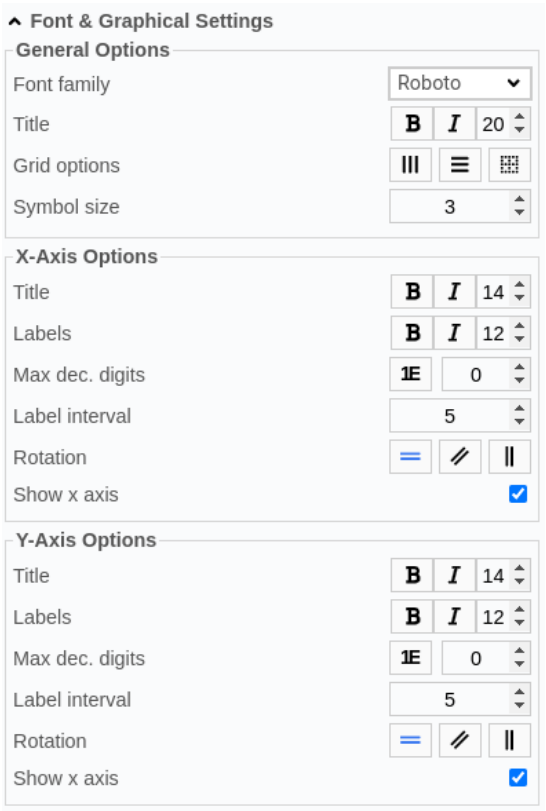


Figure 20. Font and graphical general settings.

Single Cell RNA-Seq Cell Type Prediction

SINGLE CELL RNA-SEQ CELL TYPE PREDICTION

Identifying cell types is a crucial yet challenging step in single-cell RNA sequencing analysis. The complexity arises from the vast heterogeneity of cell populations and the subtle differences in gene expression profiles, even within similar cell types. Cell type prediction algorithms play a pivotal role in addressing this challenge by comparing the gene expression profiles of the cells under study to annotated reference datasets.

While these tools provide valuable insights, it is important to recognize that they do not offer definitive predictions. Instead, they serve as an additional layer of evidence, guiding researchers toward a more informed interpretation of the likely cell types present in their data. Further exploration of the data with other references and by looking at group-specific gene expression is still needed to achieve a robust prediction.

Nevertheless, OmicsBox offers two powerful algorithms to perform cell type prediction:

- **SingleR**. This tool labels **individual cells** by comparing their gene expression profiles against a reference dataset, which can be downloaded from public databases. This approach assigns each cell a type based on gene expression profile similarities.
- **CellKb**. This tool labels **groups of cells** by comparing their differentially expressed genes against a curated knowledge base. With this tool, no external reference annotation is needed. A differential expression analysis with Scanpy between groups is performed previous to cell type annotation.

SINGLE CELL RNA-SEQ CELL TYPE PREDICTION WITH SINGLER

Introduction

Identifying the cell types present in your dataset is one of the key steps in Single-cell RNA-Seq analysis (scRNA-Seq). On this matter, the reference-based methods demonstrated to be very powerful and sensitive, being SingleR one of the most widely used.

This method compares the gene expression patterns of the cells in a query scRNA-Seq dataset with the expression of a reference, annotated, single-cell dataset. The method is divided into the following steps:

1. **Training**. Gene markers are identified for each cell type on the reference dataset. Those gene markers will be used to identify each cell type in the classification step.
2. **Prediction**. Spearman correlation scores are computed for each query cell and reference label. The label with the highest score is assigned to each cell.
3. **Fine-tuning**. For each cell, a second round of prediction is performed only with the highest scoring labels.
4. **Pruning**. Low-confident label assignments are pruned.

Please cite SingleR as:

Aran D, Looney AP, Liu L, Wu E, Fong V, Hsu A, Chak S, Naikawadi RP, Wolters PJ, Abate AR, Butte AJ, Bhattacharya M (2019). "Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage." Nat. Immunol., 20, 163-172. doi:10.1038/s41590-018-0276-y.

Run Single-cell RNA-Seq Cell Type Prediction

This option is available on the Side Panel of a Seurat Clustering object (*Side Panel > Actions > Cell Type Prediction*) (Figure 1). This will open a wizard to specify the reference annotation and the execution parameters.

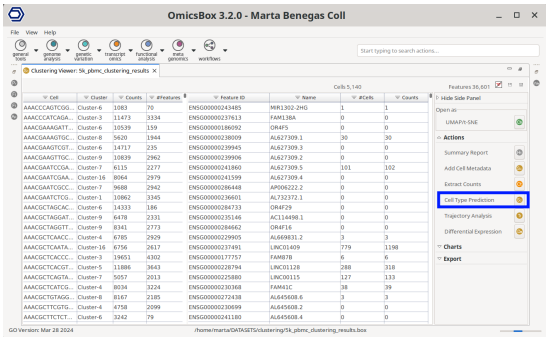


Figure 1. Cell Type Prediction tool in the Side Panel of a scRNA-Seq Clustering results object.

Input

Specify the input reference annotation on this page and which metadata to use during the analysis (Figure 2).

Reference Annotation

- **Reference Format.** The format of the reference annotation.
  - H5 Annotated Data. Files with the .h5ad extension, called AnnData. This is a compressed format used by many single-cell data scientists. It can be visualized with the software HDFView or loaded in OmicsBox. Inside the h5ad file, the count matrix must be stored in a group named "X", the cell metadata must be in a group named "obs" and the feature metadata in a group named "var". For more details about the format, please visit the AnnData Documentation.
  - Text File. Plain text file containing the count table. Cells must be in columns and genes in rows. In order to provide the cell annotations, an additional file must be specified in the "Annotation" parameter.
  - OmicsBox File. A .box file containing a Single-cell Annotated object, which can be the result of a Clustering, Cell Type Prediction, or Trajectory analysis.
  - **Reference.** Select here the single-cell annotated file.
  - **Annotation**. Only available if the "Reference Format" parameter is set to "Text File". It must be a text file containing the labels for each cell in the "Reference" file. It must include a header, one row per cell, and columns must be separated by a tab.
  - **Cell Types.** Factor in the cell metadata to predict the cell types. The values present in this factor will be assigned to the cells in the query dataset.
  - **Aggregate Refrence**. If checked, the reference counts will be aggregated by cell type. It is recommended for highly-sparse reference datasets. In addition, it increases the computational speed.

Single-cell RNA-Seq reference annotation can be downloaded from databases or generated with OmicsBox. Recommended databases are Tabula Sapiens, Tabula Muris, CellXGene, scPlantDB, etc.

Feature Matching

- **Select Query Feature**. The type of feature to use from the query single-cell dataset: "ID" or "Name". They correspond to the "Feature ID" and "Name" columns in the opened clustering results, respectively.
- **Select Matching Reference Feature**. The type of feature name or ID to use from the reference annotation. It is mandatory that the selected feature type matches the feature names in the query dataset.

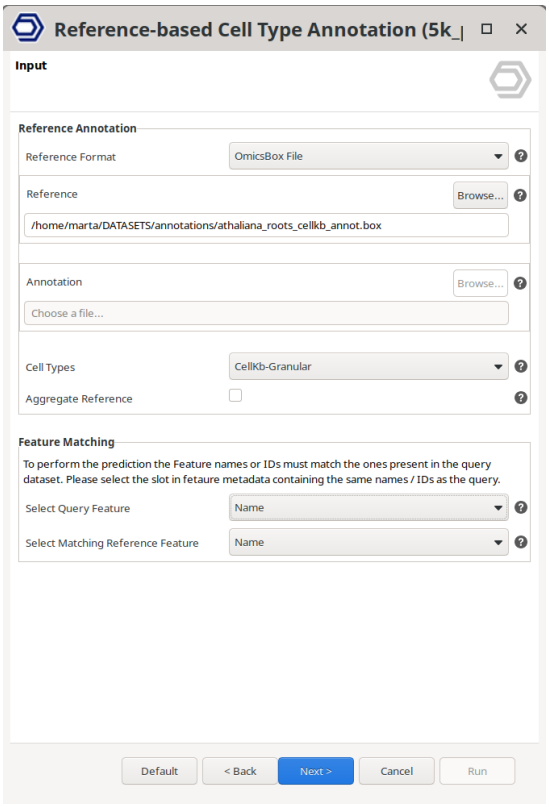


Figure 2. Input wizard page.

Configuration

This page allows configuring the parameters for the cell type prediction, refining, and pruning steps (Figure 3).

Classification Parameters

- **Annotate by Group**. When checked, the annotation is performed per group selected in the Group parameter, instead of per cell. This means that all cells within each selected group will be treated as a single unit for cell type prediction.
  - **Group**. Select the cell metadata group to use for group-based annotation. This parameter is only available when "Annotate by Group" is checked.
- **Gene Marker Selection Method**. First, SingleR is trained with the reference single-cell annotation by computing marker genes for each of the cell types present. Select here the statistical method to find differentially expressed (DE) genes between pairs of cell type labels in the reference. The identified DE genes obtained for each cell type will be used as marker genes to perform the prediction. Available options:
  - Classic. For each gene, this method computes the log-fold change between the medians of labels. Then, it sorts genes by the log-fold changes and takes the top DE genes. It is more suited if the reference comes from a bulk RNA-Seq analysis.
  - Wilcoxon Ranked Sum-Test. Instead of comparing means, this test evaluates differences in ranks of observations between two groups. Then, it takes the top 10 upregulated genes per comparison. More suitable in scenarios where data does not adhere to normal distribution assumptions, like Single-cell RNA-Seq references.
  - Welch T-test. This statistical test compares the means of each pair of labels, even when their variances are unequal. Then, it takes the top 10 upregulated genes per comparison. It is especially useful when assumptions of equal variances are not met.
  - **Tune Threshold.** SingleR performs a first round of cell type predictions and, for each cell in the query, an annotation score is calculated for all the labels in the reference. The label with the highest score is assigned to each cell. During fine-tuning, a second round of prediction is performed only with the highest-scoring labels. The 'tune threshold' sets a range below the highest score to decide which labels to keep for refining the prediction. Only labels within this range are considered in the next iteration during the classification process. For example, consider cell X with 3 cell type candidates: A = 1, B = 0.9, C = 0.85. With a 'tune threshold' of 0.1, only cell types A and B will be included in the next classification cycle of SingleR.
  - **Quantile**. A numeric value (between 0 and 1) that specifies which quantile of the correlation distribution to use when computing the score for each label. This parameter controls the balance between specificity and robustness in cell type assignment. Higher values (e.g., 0.8-0.9): Focus on the most similar reference cells. This approach is more specific and precise, but may leave some cells unassigned if they don't have strong matches. Lower values (e.g., 0.5-0.7): Focus on broader similarity patterns. This approach is more robust and assigns more cells, but may increase noise and less confident assignments.

Pruning Parameters

After predicting and fine-tuning the prediction, a pruning step is performed. It removes low-quality assignments based on the cell-label score.

The SingleR algorithm inherently labels every cell, even when the cell's true label is not present in the reference set, leading to potentially incorrect assignments. To identify and prune low-quality prediction, SingleR calculates a "delta" value for each cell, representing the difference between the score for the assigned label and the median score across all labels. A small delta suggests that the cell matches all labels with similar confidence levels, meaning that the assigned label is less significant. There are two methods to prune labels:

- **Prune Outliers**. For every label before fine-tuning, delta distribution across all assigned cells is generated (Figure 4). Cells falling more than a given number of Median Absolute Deviations (MADs) below the median score are pruned. This approach assumes that the majority of cells are accurately assigned to their true labels and that cells sharing the same label exhibit an unimodal distribution of delta values.
- **Nº MADs.** Numeric scalar specifying the number of Median Absolute Deviations (MADs) to use for defining low outliers in the per-label distribution of delta values. The default is 3, which is motivated by the fact that, for a normal distribution, 99% of observations lie within 3 standard deviations from the mean. Smaller values for Nº MADs will increase the stringency of the pruning.
- **Prune by Threshold**. Cell labels with deltas under a given threshold are pruned. This serves as an alternative filtration method if the assumptions underlying outlier detection are not met. For instance, if a label consistently experiences misassignment, the erroneous assignments will not be pruned. In such scenarios, setting a threshold will help with the removal of low-scoring cells.
- **Min. Delta.** The minimum acceptable delta for each cell.

Output

- **Save PNG Scores Heatmap.** Where to store scores heatmap in png format.

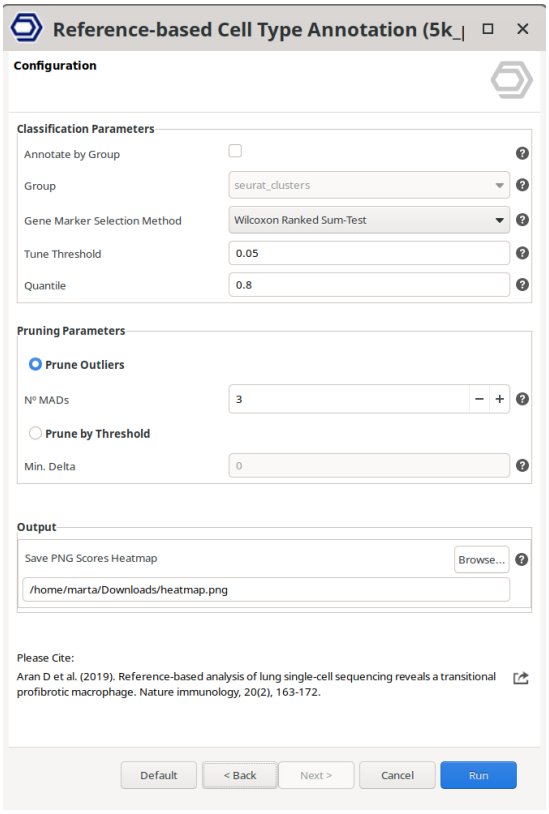


Figure 3. Configuration wizard page.

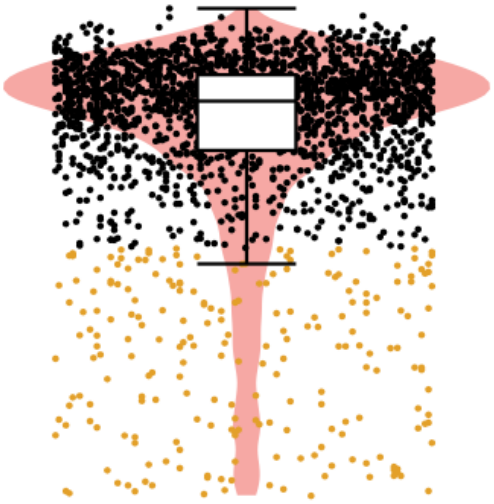


Figure 4. Example of a Delta Distribution score for a cell type. Each dot represents a cell, colored in yellow if it has been pruned.

Results Annotated Clustering Object

The main result is an updated clustering object with the cell-type predictions stored in the cell metadata. The new annotations can be seen in the UMAP/tSNE viewer (Figure 5). The new annotation will be named as the input reference file. In addition, a second annotation with the suffix "**_pruned**" will be generated as well. In this annotation, the cells that have not passed the pruning thresholds will be labeled as "Pruned". This visualization is useful to see if the pruned labels are more present in a

particular cell type or if they are distributed along all the cell types. The former case may indicate that the real cell type of the cells labeled as "Pruned" is not present on the reference dataset. This visualization also aids in identifying the amount of pruned labels visually.

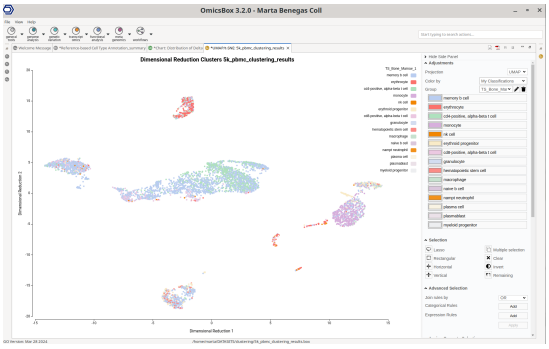


Figure 5. UMAP colored by the predicted cell types.

Delta Distribution Chart

For each predicted cell type, a violin plot showing the delta score distribution is shown (Figure 6). Each violin plot contains only the cells assigned to that label. The dots (cells) colored in yellow represent cells that have been pruned. Please see the above "Pruning Parameters" section for a more detailed description of the delta score and the pruning procedure.

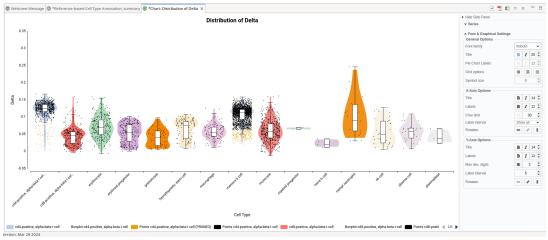


Figure 6. Delta Score distribution for each predicted cell type. Cells colored in yellow have been pruned.

Scores Heatmap

This heatmap shows the scores for all cells across the reference labels, allowing an easy assessment of the confidence of predicted labels (Figure 7). Ideally, each cell (represented by a column in the heatmap) should exhibit one score significantly higher than the others, indicating a clear assignment to a single label. However, if scores for a cell are similar, it suggests uncertainty in the assignment. Nevertheless, this might be acceptable if the uncertainty spans similar cell types that are difficult to distinguish.

The labels displayed on the top legend are the final assignment. It must be noted that the final label assignment may not correspond with the highest-scoring label (more yellow). This is because the scores displayed are the ones obtained before the fine-tuning step, since the scores after it are not comparable between labels.

This heatmap is generated by SingleR and is stored in the location indicated by the "Save PNG Scores Heatmap" parameter.

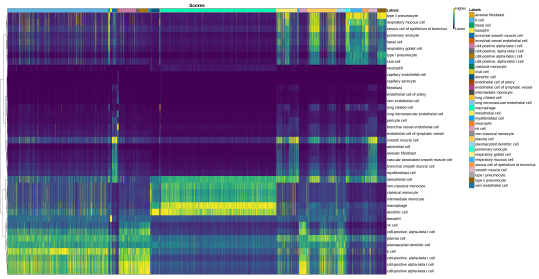
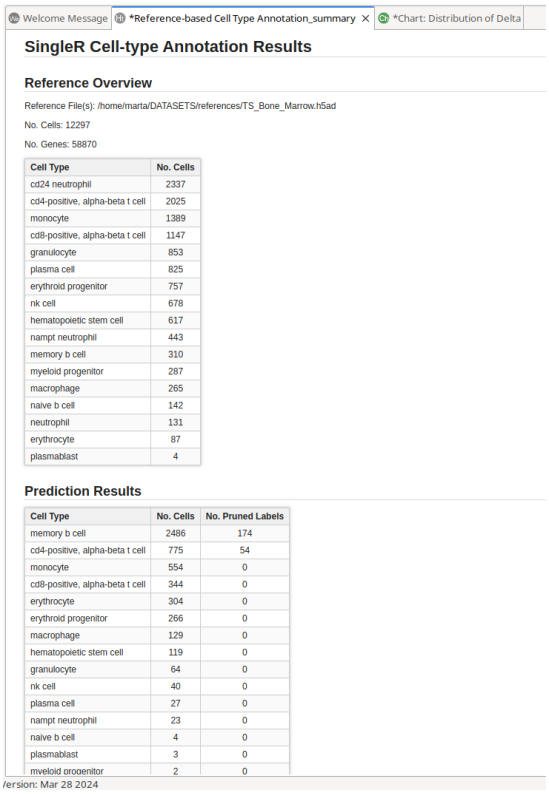


Figure 7. Score heatmap with cells in columns and labels in rows. The value is the score given for a particular cell-label pair. The labels displayed on the top legend are the final label assignment.

Summary Report

This summary (Figure 8) provides a basic overview of the reference scRNA-Seq annotation used for prediction. It shows the number of cells, genes, and the number of cells for each cell type. In addition, the number of cells in the query assigned to each cell type is also displayed, along with the number of pruned cells.

The parameters used for the analysis and the citation are displayed at the bottom of the summary report.


Welcome Message | *Reference-based Cell Type Annotation_summary | *Chart: Distribution of Delta

SingleR Cell-type Annotation Results

Reference Overview

Reference File(s): /home/marta/DATASETS/references/TS_Bone_Marrow.h5ad

No. Cells: 12297

No. Genes: 58870

Cell Type	No. Cells
cd24 neutrophil	2337
cd4-positive, alpha-beta t cell	2025
monocyte	1389
cd8-positive, alpha-beta t cell	1147
granulocyte	853
plasma cell	825
erythroid progenitor	757
nk cell	678
hematopoietic stem cell	617
nampt neutrophil	443
memory b cell	310
myeloid progenitor	287
macrophage	265
naive b cell	142
neutrophil	131
erythrocyte	87
plasmablast	4

Prediction Results

Cell Type	No. Cells	No. Pruned Labels
memory b cell	2486	174
cd4-positive, alpha-beta t cell	775	54
monocyte	554	0
cd8-positive, alpha-beta t cell	344	0
erythrocyte	304	0
erythroid progenitor	266	0
macrophage	129	0
hematopoietic stem cell	119	0
granulocyte	64	0
nk cell	40	0
plasma cell	27	0
nampt neutrophil	23	0
naive b cell	4	0
plasmablast	3	0
myeloid progenitor	2	0

/ersion: Mar 28 2024

SINGLE CELL RNA-SEQ CELL TYPE PREDICTION WITH CELLKB

Introduction

CellKb is an advanced tool that combines a robust **cell-type prediction algorithm** with an extensive **knowledge base**.

The knowledge base contains thousands of manually curated references obtained from research papers. For each cell type identified on a paper, the ranked list of up-regulated genes is kept and enriched with sample metadata. The ranked gene list is called a gene signature. Thus, the same cell type in the knowledgebase has multiple signatures (or ranked gene lists) associated, coming from different experiments.

The prediction algorithm operates by comparing a provided list of up-regulated genes against the signatures in the knowledge base. By evaluating the similarity between the query and the curated signatures, this approach delivers highly precise cell-type predictions. This approach enables CellKb to harness the collective power of diverse reference datasets, ensuring reliable and context-aware predictions.

Thus, previously to run CellKb, the list of up-regulated genes for each group of cells is needed. To this end, the widely known Scanpy package is used.

For more information about the knowledgebase curation and the cell type prediction approach please visit:

Ajay Patil, Ashwini Patil. CellKb Immune: a manually curated database of mammalian hematopoietic marker gene sets for rapid cell type identification. bioRxiv 2020.12.01.389890; doi: 10.1101/2020.12.01.389890.

For more information about CellKb please visit:

CellKb. Combinatics Inc. https://www.cellkb.com/.

For Scanpy please cite:

Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018 Feb 6;19(1):15. doi: 10.1186/s13059-017-1382-0.

Run scRNA-Seq Cell Type Prediction with CellKb

This option is available on the Side Panel of a Seurat Clustering object (Side Panel > Actions > Cell Type Prediction) (Figure 1-A). To run CellKb, select the option on the opening wizard (Figure 1-B).

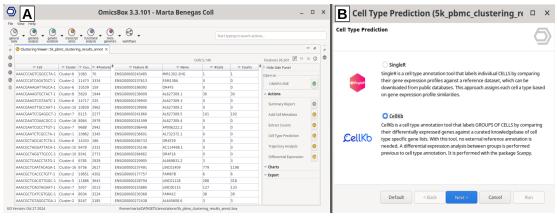


Figure 1. *(A) Cell Type Prediction tool in the Side Panel of a scRNA-Seq Clustering results object, and (B) opening wizard to select CellKb algorithm.*

Configuration 1: Differential Expression Between Groups.

The first step of the analysis is to perform a differential expression analysis between the selected groups of cells (Figure 2). This will generate a ranked list of up-regulated genes per subgroup, that later will be used to query against the CellKb knowledge base and obtain a cell type prediction.

- **Group**: Select the group to perform the analysis.
- **Diff. Expression Method**: Select the statistical test to perform differential expression between the subgroups.
  - T-Test: This method performs a standard statistical test to compare the means of two groups, assuming normally distributed data and equal variance between groups.
  - T-Test with Variance Overstimation: A variation of the T-Test that overestimates variance to account for noise or outliers.
  - Logistic Regression: The logistic regression models the probability of group membership based on gene expression, enabling the detection of subtle patterns at the cost of higher computational demand.
  - Wilcoxon rank-sum: This is a non-parametric approach that compares ranks instead of raw values, making it robust to non-normal data and suitable for sparse single-cell datasets.

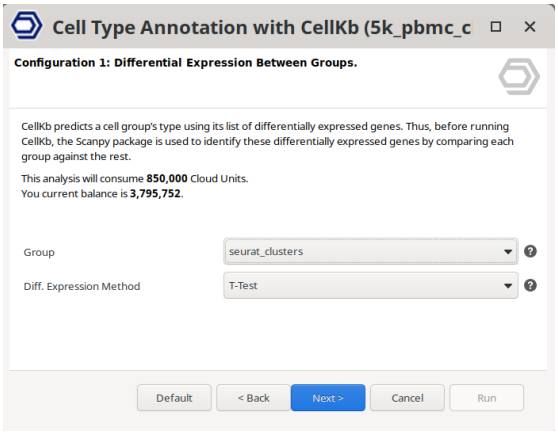


Figure 2. *Differential Expression configuration wizard page.*

Configuration 2: Cell Annotation.

This configuration wizard page allows specifying filters for the knowledge base, so our data is only compared against the gene lists meeting the applied criteria (Figure 3). This allows for a more tailored and precise annotation.

- **Input Species**: Select the species of your data. It is only possible to analyze species on this list. Please see the info panel above for more information.
- **Query Species**: Query your data against gene lists from this species. This filter is mandatory. The query species can be different from the input species for cross-species annotation.
- **Tissues, Conditions, Cell Types**: Compare your data against gene lists from the selected tissue(s), conditions(s), and/or cell type(s). Those filters are optional and multiple options can be selected on each of them. Each time an option is selected on one of the filters, the options available in the rest are updated. Start typing or press the space key to see available options and click to add one.

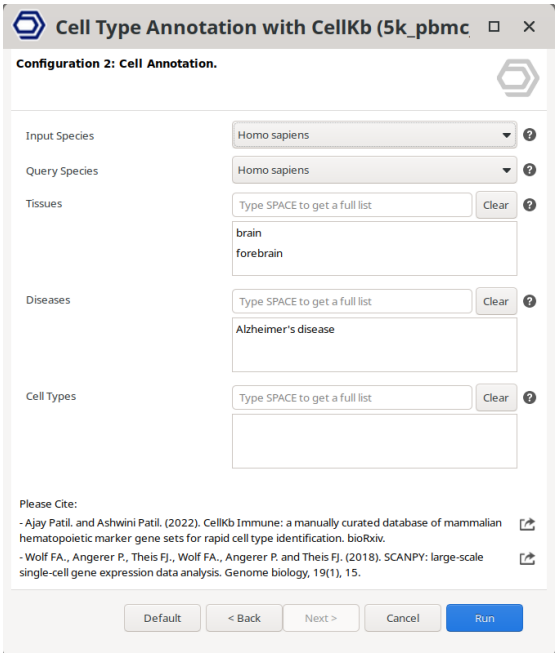


Figure 3. *Cell Annotation configuration wizard page.*

Adding more species

Only the species listed in the “Input Species” menu are available for cell type prediction. If you are missing a species, please feel free to contact support with your request (support@biobam.com).

Additionally, if you would like to add a reference to the knowledge base, please contact support with the reference you would like to add. The suggestions must be accompanied by the paper so it can be evaluated prior to adding it to the CellKb knowledge base.

Results

Annotated Clustering Object

The main result is an updated clustering object with the cell-type predictions stored in the cell metadata. The new annotations can be seen in the UMAP/tSNE viewer (Figure 4) under the “My Classifications” category. The new annotations will be named “CellKb-Broad” and “CellKb-Granular”. The first annotation contains more general cell-type terms, whereas the latter contains more specific terms. The names can be changed on the UMAP/tSNE viewer.

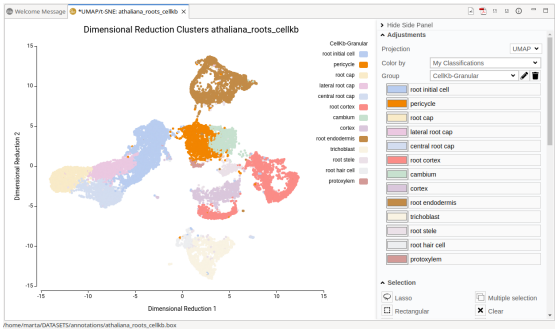


Figure 4. *UMAP colored by the predicted cell types.*

Summary Report

The summary (Figure 5) provides an overview of the cell type prediction. For each group, the final predictions including both broad (general) and granular (specific) cell types, along with a detailed list of matches to the reference are shown.

It also presents the candidate cell type predictions with their scores, as well as a list of the cell type marker genes that were also found in the query list, if any. These candidate cell types are chosen from the top gene lists matching the query. The score assigned to each candidate is a **relative** score, calculated by comparing the rank-based scores of the top matching references with each other. Thus, a high score means that the rank-based score of that candidate cell type is significantly higher than the rank-based score of all other cell types.

The detailed results (Figure 6) are shown after clicking on the button in the “Top Hits” column. The table shows the top 10 most similar reference gene lists with their associated statistics. The ranked-based score measures the degree of similarity between the query and the reference gene lists. It takes into consideration the number of genes in the query and the reference gene list, the number of overlapping genes between the two, and their positions on the lists.

Info

CellKb will always return an annotation, whenever at least one gene is found in the reference gene lists that match the filter criteria. It returns the cell type whose gene list is the closest match to the query, but it doesn't imply that it's a strong match. Thus, it is highly recommended to look into the detailed results to verify the robustness of the prediction.

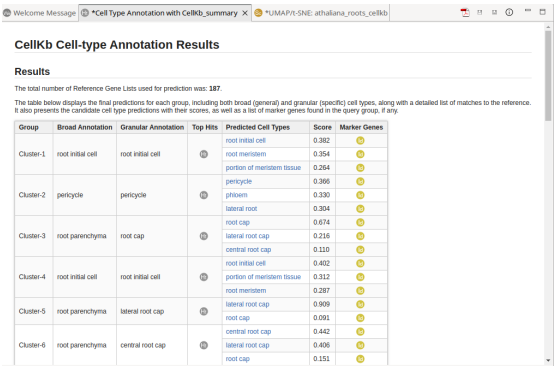


Figure 5. *UMAP colored by the predicted cell types.*

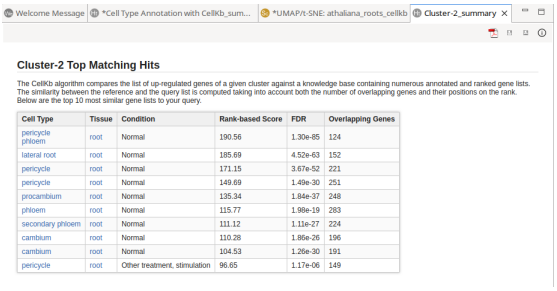


Figure 6. *UMAP colored by the predicted cell types.*

Single Cell RNA-Seq Trajectory Inference

SINGLE CELL RNA-SEQ TRAJECTORY INFERENCE

Introduction

Monocle3 is a scRNA-Seq data analysis toolkit developed by Trapnell lab, mainly used for Trajectory Inference analysis. Trajectory inference analysis aims to reconstruct the developmental trajectory of single cells, mapping out their developmental paths or states. The cells are stratified with the "Pseudotime", which measures the progression of individual cells along some biological processes. The essential input required for trajectory inference analysis is the knowledge of starting points or root cells. Please cite Monocle3 as:

Qiu, Xiaojie, et al. "Reversed Graph Embedding Resolves Complex Single-Cell Trajectories." *Nature Methods*, vol. 14, no. 10, 21 Aug. 2017, pp. 979–982, 10.1038/nmeth.4402.

Qiu et al."Single-Cell mRNA Quantification and Differential Analysis with Census." *Nature Methods*, vol. 14, no. 3, 23 Jan. 2017, pp. 309–315, 10.1038/nmeth.4150

Trapnell, Cole, et al. "Pseudo-Temporal Ordering of Individual Cells Reveals Dynamics and Regulators of Cell Fate Decisions." *Nature Biotechnology*, vol. 32, no. 4, 1 Apr. 2014, pp. 381–386, 0.1038/nbt.2859.

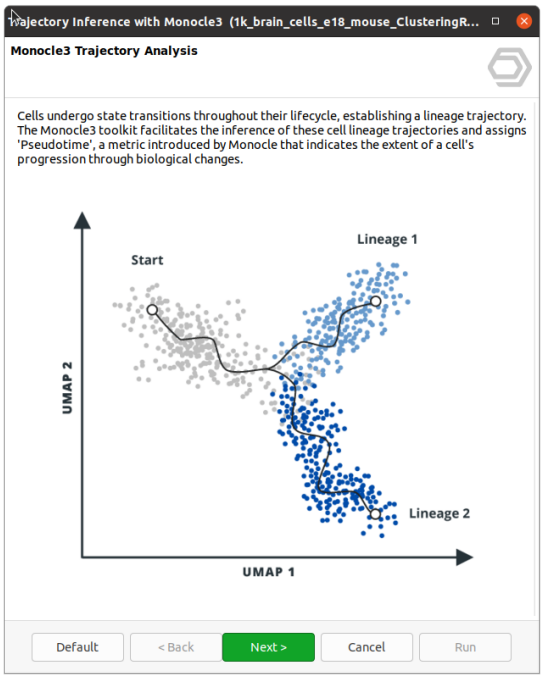


Figure 1. *Monocle3 Wizard in OmicsBox*

Accessing Monocle3 in Omics Box

One of the essentials to perform trajectory analysis is knowing the starting or root cells. Therefore, the Monocle3 Trajectory Inference wizard is available right after the clustering analysis. After completing the scRNA-Seq Clustering, click "Trajectory Analysis" on the side panel to initiate Monocle3 (refer to Figure 2).

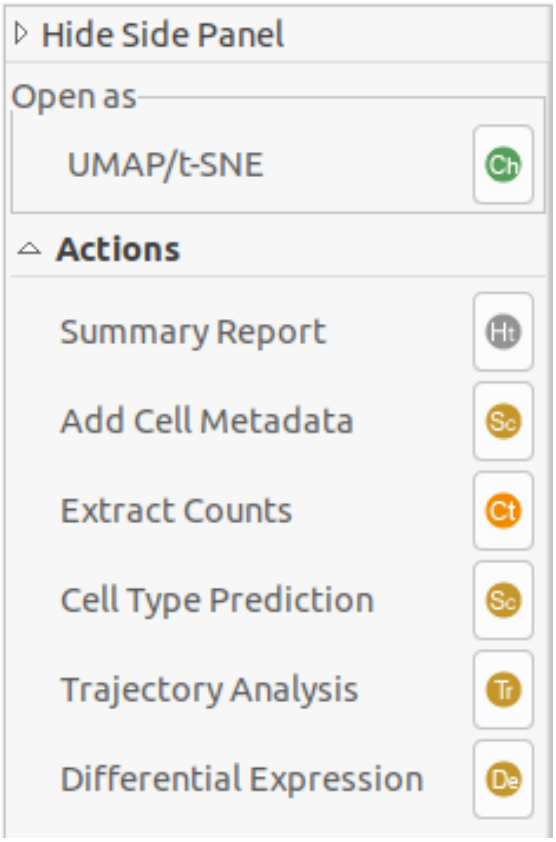


***Figure 2**. Trajectory Inference with Monocle3, available as the side panel option of scRNA-Seq Clustering output (After Seurat Clustering Analysis)*

Select Starting Points (Root Cells) of the Trajectory

Select the root node (a collection of root cells) for Monocle3, as it serves as the reference point for trajectory construction. OmicsBox offers two methods to provide this information:

1. *Cell Metadata*: The Monocle3 wizard automatically reads the cell metadata information from the clustering results.
2. *Metadata Group:* The options available will display all potential columns with root information based on the information in the cell metadata.
3. *Root Cells:* After selecting the column with potential starting point information, decide on the actual starting point. If you're considering experimental time, the starting point might be the initial capture time (0h) or something similar. Alternatively, it could be a specific cell type, like a hematopoietic stem cell or Seurat's cluster-x.
4. *List of Root Cells:* Supply a text file with the list of cell IDs as root cells (with one name per line). These cells act as the starting points for the trajectory.

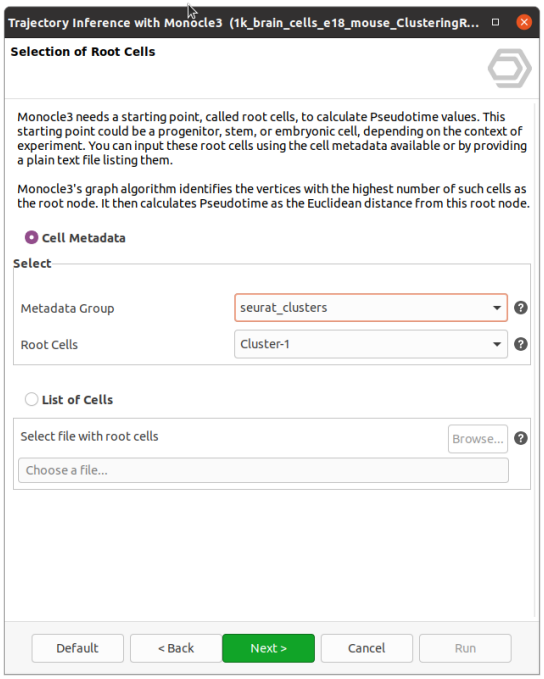


Figure 3. *Selection of starting points for the trajectory analysis*

Configuration 1. Data Pre-processing

1. *Transformations:* Monocle3 can perform transformations independently or use the transformation applied during Seurat's clustering analysis. When "Raw counts" is selected, the following options are available to fine-tune normalization and feature selection.
2. *Normalization Method:* Normalization aims to minimize non-biological variation. Two options are available: log-normalization and size-factor normalization. Log normalization standardizes data, which is especially useful for columns with high variance. Size-factor normalization removes bias from each cell by dividing its counts by size factor. The user can also skip the normalization by selecting "none".
3. *Principal component analysis (PCA)*: This classic dimension reduction method creates linear combinations of gene expressions termed as principal components (PCs). These PCs, orthogonal to each other, effectively capture the gene expression variation and often have a lower dimensionality.
    a. *Dimensions:* This refers to the number of dimensions post-PCA. Selecting the top 50 principal components for datasets exceeding 5,000 cells is advisable.
    b. *Scaling:* Scaled data facilitates model learning. Scaling before PCA computation is beneficial when dealing with variables in different units.
    c. *Embeddings*: Monocle3 can recompute the UMAP embeddings from scratch or use Seurat's UMAP/t-SNE embeddings for trajectory analysis. When "Re-Compute UMAP" is selected, the following options are enabled to compute the UMAP embeddings.
4. *UMAP Minimum Distance:* This parameter dictates UMAP's cell clustering tightness. Low values result in dense cell clusters, while higher values emphasize preserving broad topological structure.
5. *UMAP Neighbours:* This balances local versus global structures. Lower values direct UMAP to concentrate on local structures, while higher values emphasize a broader view, potentially sacrificing fine details.

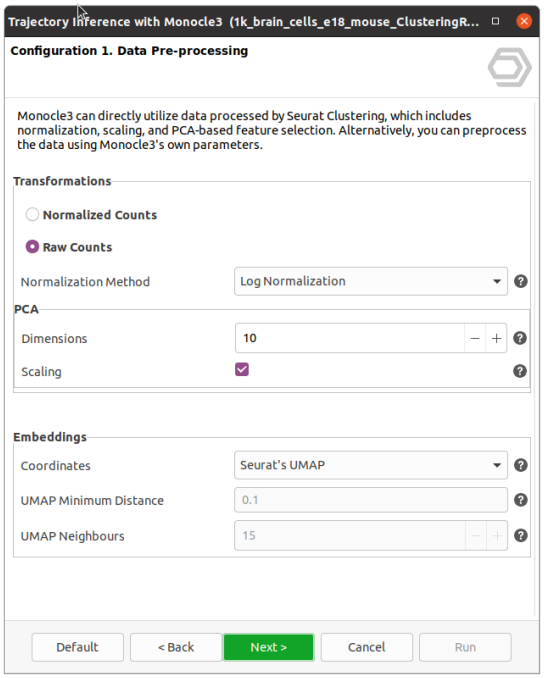


Figure 4. *Parameter tuning for Monocle3-based trajectory analysis.*

Configuration 2. Clustering and Trajectory Control *Clustering:*

Clustering of cells during the trajectory analysis significantly reduces the computational complexity of learning the trajectory graph. In a trajectory, clustering represents the stable checkpoints (cell-states) in a biological process. Monocle3 in the OmicsBox can perform its own clustering or use the clusters inferred during Seurat's clustering analysis.

1. *Cluster Method:* Select the algorithm for clustering. If "Seurat-Clusters" is selected, the cluster labels from Seurat clustering results are transferred to the Monocle3. The other options are "Louvain Clustering" and "Leiden Clustering", for which the following parameters are available:
2. *Nearest Neighbours:* Set the number of expected clusters (k-clusters).
3. *Resolution:* Set the resolution clustering. A higher value generates a larger number of smaller clusters, and a lower value generates a smaller number of larger clusters.

Fine Tune Trajectory:

1. *Allow Disjoint Graph:* Activating allows merging different partitions into a single trajectory. Otherwise, distant partitions are allotted "Infinite" pseudotime.
2. *Allow loops:* Whether to look for potential cyclic trajectories within the data.
3. *Number of Centers:* This will determine the expected centres in a trajectory.
4. *Prune Branches:* Whether to remove branches that do not meet the specific length criteria.
5. *Minimum Branching Length:* Set the minimum length of the branch (number of centres in a branch).

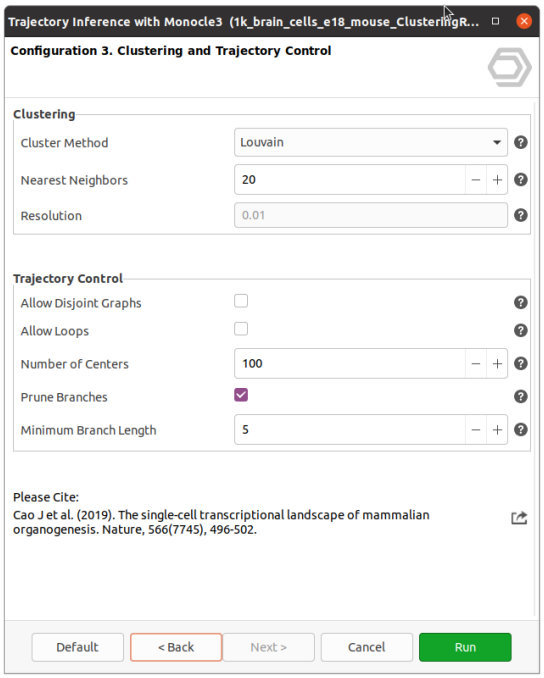


Figure 4. *Parameter tuning for Monocle3-based trajectory analysis and clustering.*

Side Panel Actions

You can see the side panel actions after completing the analysis and obtaining the results. The currently available side panel options include:

1. UMAP/t-SNE: This will open up an interactive wizard.
2. Add Cell Metadata: Add information per cell.
3. Summary Report: Produces a summary of the analysis.
4. Extract Count: Retrieves the count of different Pseudotime Ranges.
5. Autocorrelation: Analyze differences in the gene expression along the trajectory.
6. Differential Expression: Analyze the differences in gene expression.

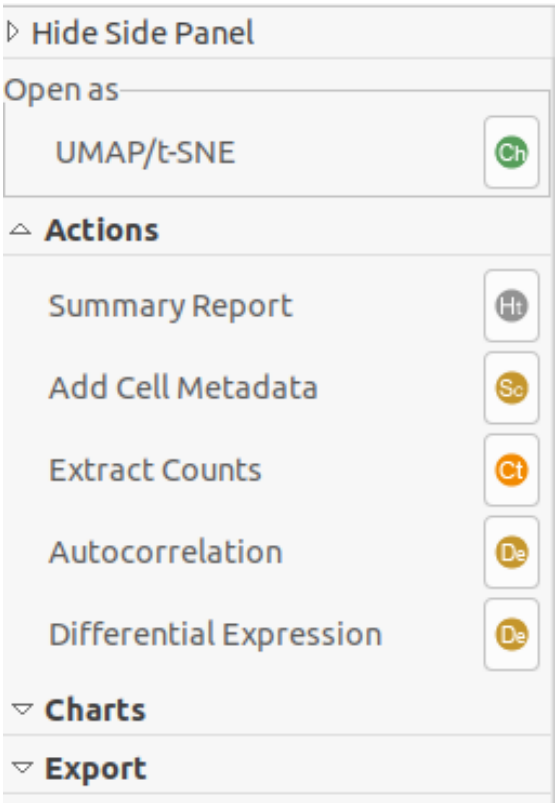


Figure 5. *Side Panel Action after the Monocle3 trajectory analysis in OmicsBox*

Actions: UMAP/t-SNE

Opens UMAP/t-SNE wizard for intuitive analysis of the results.

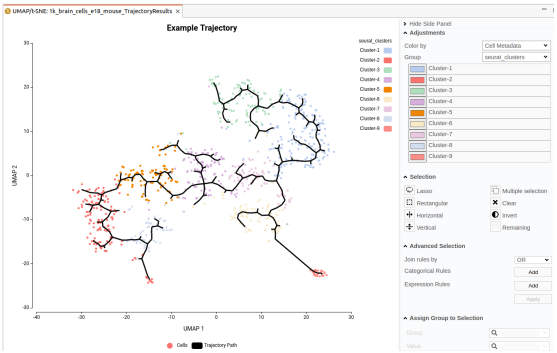


Figure 6. *Interactive UMAP/t-SNE Plots for intuitive interpretation of the results.*

Actions: Summary Report

Generates a summary report of the analysis.

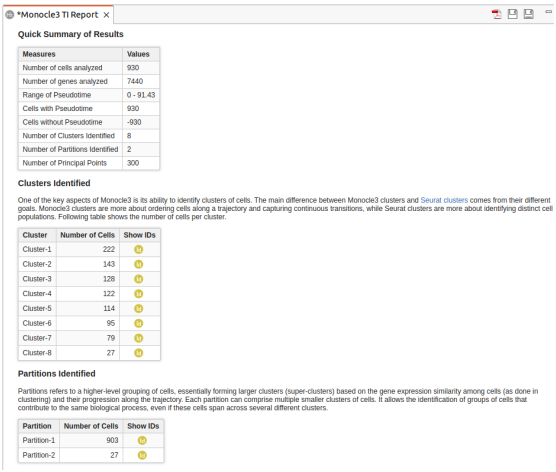


Figure 7. *Detailed summary of the results and the parameters used during the analysis.*

Actions: Extract Cluster Counts

1. *Group:* Select the group for which you want to extract the counts. It includes all the columns of the cell-level metadata describing different attributes of the cells.
2. *Subgroup(s) to Extract:* Select the subgroups (e.g., a specific cell type) from which to extract the counts.

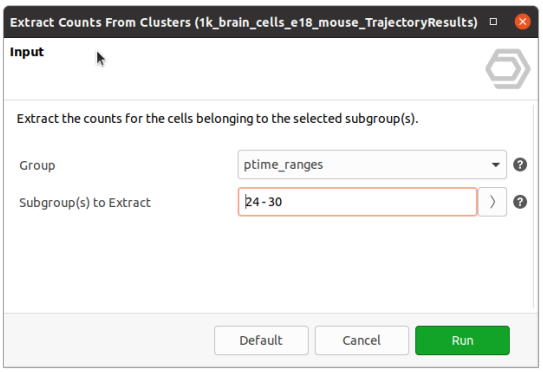


Figure 8. *OmicsBox wizard for the extraction of raw counts from Monocle3 results.*

Actions: Autocorrelation Analysis

Monocle3 provides a way of finding genes that vary between groups of cells in UMAP space. It uses a statistic from spatial autocorrelation analysis called Moran's I, which Cao & Spielmann et al. showed effective in finding genes that vary in single-cell RNA-seq datasets. Visit Monocle3 Autocorrelation Analysis using OmicsBox to learn more.

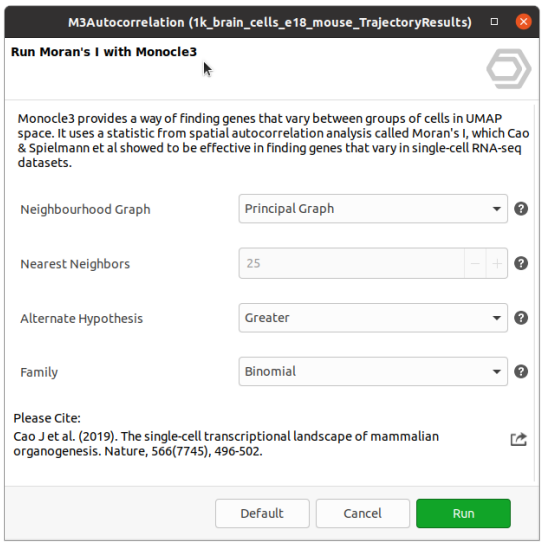


Figure 9. *OmicsBox wizard for Monocle3 Autocorrelation Analysis.*

Actions: Differential Expression

For differential expression analysis, refer to the single-cell differential expression tutorial. In this context, the pseudotime range labels are used instead of Cluster labels.

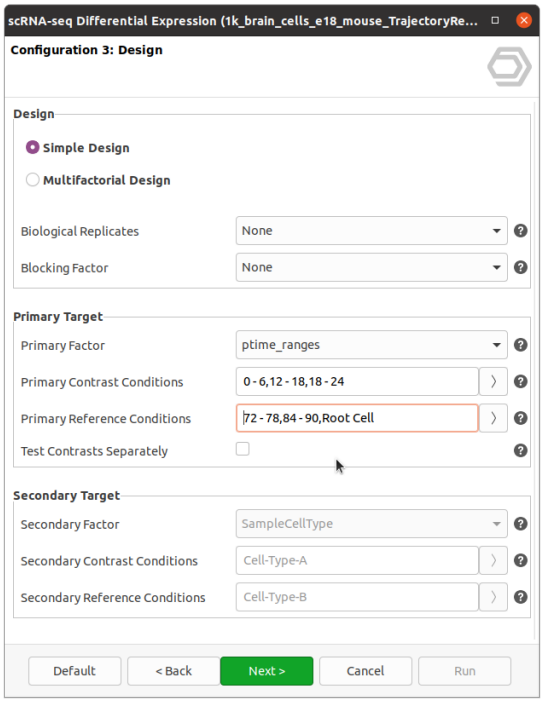


Figure 10. *OmicsBox wizard for Monocle3 Differential Expression Analysis.*

Side Panel Charts

1. **Expression Trends:** Charts the trends in gene expression.
2. **Distribution of Cell in Pseudotime:** Displays how cells are distributed across different pseudotime values.

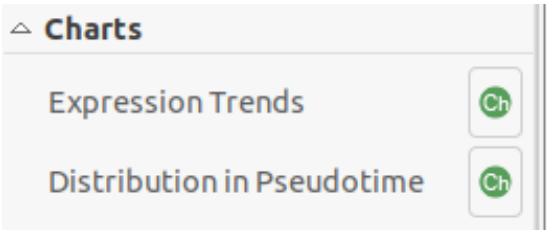


Figure 11. *Options for data visualization after trajectory analysis in OmicsBox*

Expression Trend (Side-Panel Charts)

1. *Gene ID/ Name:* Choose the feature or gene for which you want to plot the trend.
2. *Scaling:* Large variations in counts can sometimes obscure finer details. Scaling adjusts the data range to highlight these subtleties.
3. *Log Transform:* If a dataset contains minimal differences among large values, applying a log transformation can magnify these variations, making them more explicit. The process adds a pseudo-count of 0.5 to raw counts before log transformation.
4. *Smoothness:* Modulate the trend line's smoothness to fit your preference.
5. *Colour Cells By:* colour the cells based on inferred clusters or partitions from Monocle3.

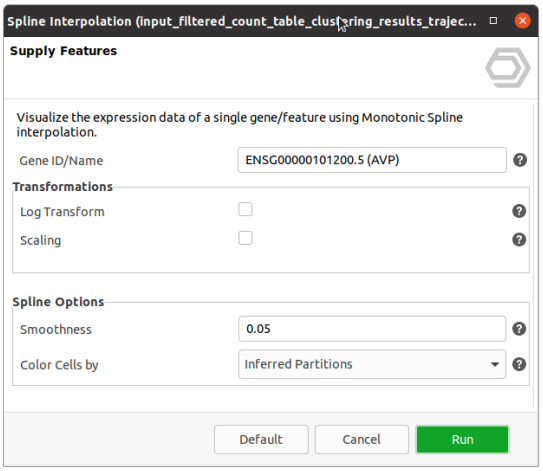


Figure 12. *Wizard for plotting gene/feature expression trends along pseudotime using monotonic spline in OmicsBox*

Output

Monocle3 in OmicsBox delivers several outputs aligned with a conventional trajectory analysis. These outputs comprise a primary output table, three key plots, and a succinct report detailing the parameters used and the results obtained.

1. **Main Output Table with Pseudotime Information**: This table provides detailed pseudotime data for the analyzed cells.
2. **Trajectory UMAP**: A visualization showing the trajectory of cells using the UMAP (Uniform Manifold Approximation and Projection) technique.
3. **Expression UMAP**: A plot that illustrates how the expression of a particular gene varies across cells in UMAP embeddings.
4. **Expression Trends**: Displays the trends in gene expression across pseudotime.
5. **Distribution of Cells Over Pseudotime Ranges**: This visualization depicts how cells are spread across various pseudotime values.

Output Table Fields:

1. **Cell:** The names of the cells are provided in the count table and experimental design file.
2. **Pseudotime:** The pseudotime assigned to each cell by Monocle3. Cells not allocated a pseudotime will not have a value in this column.
3. **Pseudotime Range:** This field represents the clusters of pseudotime. For cells without an assigned pseudotime, it will explicitly state so. The intervals for these ranges are left-closed (right-open).
4. **Cluster:** The clusters to which the cells have been assigned.
5. **Partition:** This refers to the assigned super-cluster or partition.

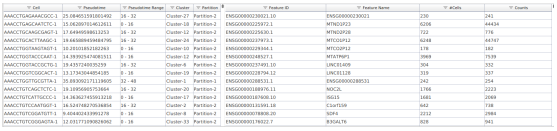


Figure 10. *Main output table*

Results (Trajectory UMAP)

The Trajectory UMAP is a visualization that combines a UMAP coloured by the continuum of pseudotime with a superimposed line graph. This line graph represents the overall progression pattern among the cells. Using the pseudotime slider, users can focus on cells within a particular pseudotime range. If a cell hasn't been assigned a pseudotime, the visualization will display the progression line without any coloured cells associated with that specific cell. Additionally, it offers an interactive selection of cells, allowing users to select a starting cell and run trajectory analysis interactively.

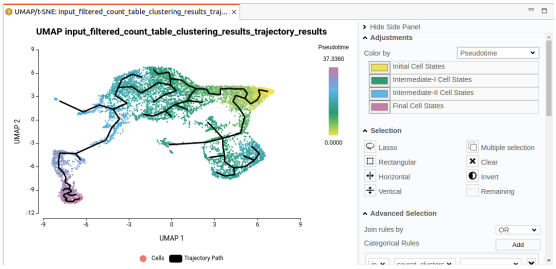


Figure 11. *Interactive Trajectory UMAP of Monocle3 in OmicsBox*

Expression Trend (Side-Panel Charts)

The expression trend plots the expression of a chosen feature gene per cell against its pseudotime using the monotonic spline interpolation method. This visualization offers insights into the expression trends of a specific gene feature along the pseudotime in different cell clusters or partitions.

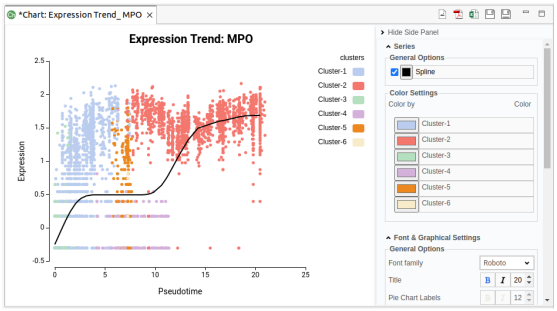


Figure 12. *Expression trend of the selected feature in OmicsBox*

Results (Distribution of Cells Across Pseudotime Range) (Side-Panel Charts)

This visualization displays the distribution of cells based on their pseudotime range, showcasing the number of cells within each specific range. By correlating this distribution with cell type annotations, one can identify progenitor cell types or intermediate cell states, which often possess lower pseudotime values. The intervals for these ranges are defined as left-closed (right-open).

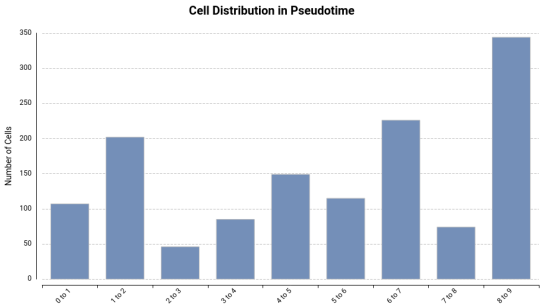


Figure 14. *Number of cells distributed across pseudotime*

SINGLE CELL RNA-SEQ MONOCLE3 AUTOCORRELATION

Introduction

Monocle3 provides an alternative method for identifying genes that vary between groups of cells in lower dimension space. In the context of trajectories, it focuses on genes that change along the trajectory graph within UMAP space.

Monocle3 employs Moran's I statistic to identify such genes, a measure of spatial autocorrelation. Spatial autocorrelation refers to a correlation in a signal among neighbouring locations in space. For scRNA-Seq trajectories, the space is the UMAP, and the locations are along the trajectory graph. By conducting this Monocel3 aims to discover genes associated with specific locations on the trajectory, such as branching points.

The concept of autocorrelation differs from that of differentially expressed (DE) genes. In scRNA-Seq trajectories, DE genes usually change their mean expression as a function of pseudotime. This does not necessarily consider the UMAP subspace or the branching patterns.

This tool is based on the R package Monocle3. Please cite Monocle3 as:

Qiu, Xiaojie, et al. "Reversed Graph Embedding Resolves Complex Single-Cell Trajectories." *Nature Methods*, vol. 14, no. 10, 21 Aug. 2017, pp. 979–982, 10.1038/nmeth.4402.

Qiu et al. "Single-Cell mRNA Quantification and Differential Analysis with Census." *Nature Methods*, vol. 14, no. 3, 23 Jan. 2017, pp. 309–315, 10.1038/nmeth.4150

Trapnell, Cole, et al. "Pseudo-Temporal Ordering of Individual Cells Reveals Dynamics and Regulators of Cell Fate Decisions." *Nature Biotechnology*, vol. 32, no. 4, 1 Apr. 2014, pp. 381–386, 0.1038/nbt.2859.

Accessing Monocle3 Autocorrelation in OmicsBox

Perform trajectory analysis with Monocle3 in OmicsBox. After the trajectory analysis, the results table will appear in the Main Table Output. On the side panel of this output, click "Autocorrelation" to initiate Monocle3 Autocorrelation (refer to Figure 1). This action uses the current trajectory analysis results as input for the Monocle3 Autocorrelation Wizard.

By default, the autocorrelation analysis will consider all the results after the trajectory analysis. If you wish to run the analysis for a subset of the trajectory, please subset it beforehand.

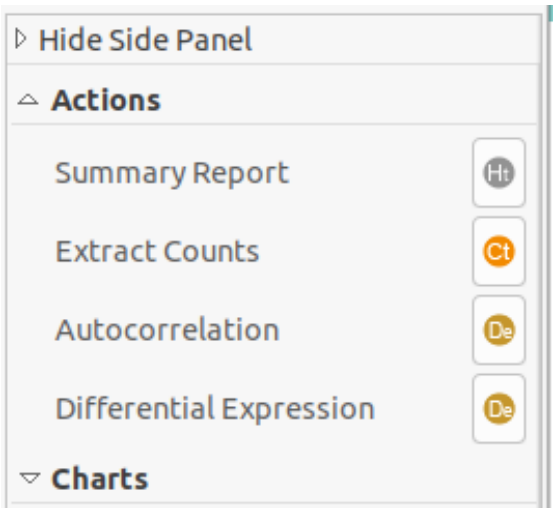


***Figure 1**. Autocorrelation with Monocle3 is available as the side panel option for Monocle3 Trajectory Inference Results in OmicsBox.*

Configure Monocle3 Autocorrelation

1. *Neighbourhood Graph:* Select whether you want to run the auto-correlation for the Principal graph (PQ-graph) or the KNN graph. The Principal graph is the default setting, which will select the trajectory line as the input, basically a PQ-Tree (a particular case of Minimum Spanning Tree).
2. *Nearest Neighbours*: This option is only enabled when the requested *Neighbourhood Graph* is set to "KNN ".
3. *Alternative Hypothesis:* Hypothesis to test against the NULL hypothesis, which states that none of the genes are significantly expressed.
4. ***Two-sided*:** This alternative hypothesis tests for any form of spatial autocorrelation without specifying the direction.
5. ***Greater:*** This tests for positive spatial autocorrelation specifically. It suggests that similar genes are clustered together in space more than expected under a random spatial distribution. This type of hypothesis is chosen when there is a reason to test for the presence of clustering.
6. ***Less****:* This test is for negative spatial autocorrelation. This implies that dissimilar genes are located near each other more frequently than would be expected by chance, indicating a spatial pattern of dispersion. This hypothesis is selected when the interest lies in detecting spatial segregation or dispersion of genes.
7. *Family:* Expected distribution of the residuals for modelling the expression. Refer to the table below,

Expression Family	Accuracy	Speed	Recommendation
Negative Binomial (Default)	+++	+	It is recommended for most users and is highly accurate.
Quasi-Poisson	++	++	It is recommended for users with massive datasets.
Poisson	-	+++	For debugging and testing only.
Binomial	++	++	High zero inflation with extremely low counts.

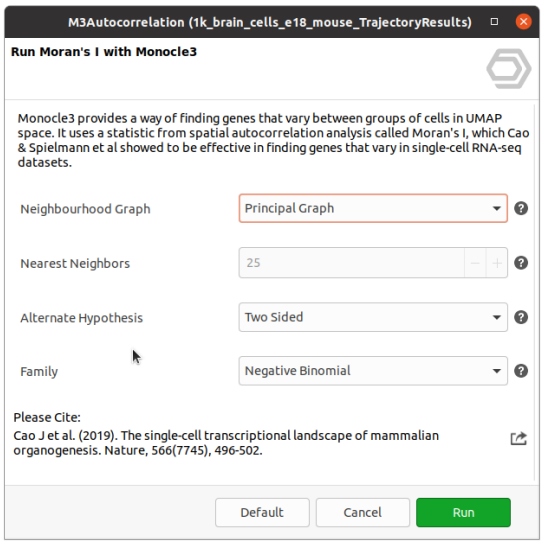


Figure 2. *Configuration of Autocorrelation Analysis.*

Side Panel Actions

You can see the side panel actions after completing the analysis and obtaining the results. The currently available side panel options include:

1. **Summary Report:** Produces a summary of the analysis.
2. **Update Tags:** Update the significant tags based on the test results.
3. **Fisher's Exact Test**: Perform overrepresentation analysis of the significantly tagged genes.

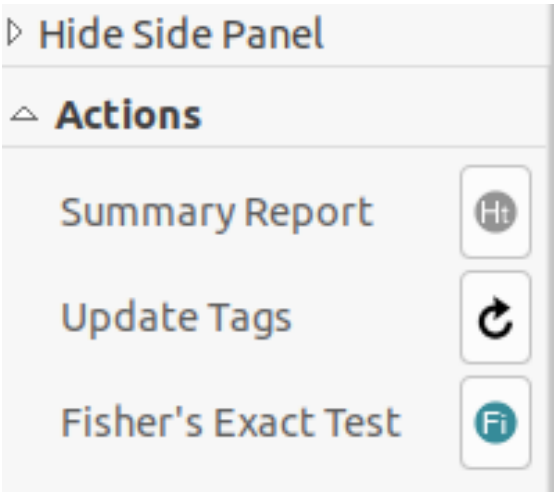


Figure 3. *Side Panel Action after the Monocle3 Autocorrelation analysis in OmicsBox.*

Output

Monocle3 Autocorrelation in OmicsBox generates a main table and a summary report. This table has the following columns.

1. **Tags:** Whether the gene is significant or not. Tags can be updated using the side panel actions.
2. **Gene ID:** IDs of the genes.
3. **Gene Name:** The gene names supplied; if the gene names are not supplied initially, this column will reflect the IDs.
4. **Moran's-I:** This is a measure of spatial autocorrelation. Moran's I is a statistic that can range from -1 to +1, where a value close to +1 indicates strong positive spatial autocorrelation (meaning similar values cluster together in space), a value close to -1 indicates strong negative spatial autocorrelation (meaning dissimilar values are adjacent), and a value around 0 indicates a random spatial pattern (no autocorrelation).
5. **Moran's Test Statistic:** Once Moran's I is calculated, it is typically transformed into a test statistic that follows a known probability distribution (under the null hypothesis of no spatial autocorrelation). The Moran's Test Statistic enables evaluating the observed Moran's I's significance, determining whether the observed spatial pattern (as measured by Moran's I) could reasonably occur by chance.
6. **P-Value:** The p-value associated with Moran's I indicate the probability of observing a Moran's I as extreme as the one calculated from your data, assuming there is no spatial autocorrelation (the null hypothesis).
7. **Q-Value:** Its relevance emerges in the context of multiple testing or multiple comparisons, such as when conducting Moran's I test across multiple spatial datasets or variables. The q-value could control the false discovery rate across all these tests.

Tags	Gene ID	Gene Name	Moran's I	Moran's Test St...	P-Value	Q-Value
	ENSMUSG00000033845	Mrpl15	0.01855	1.88691	5.917212E-2	8.392623E-2
Significant	ENSMUSG00000025903	Lypla1	0.02893	2.88305	3.938476E-3	6.850665E-3
Significant	ENSMUSG00000033813	Tcea1	0.04383	4.30331	1.682645E-5	3.994286E-5
Significant	ENSMUSG00000002459	Rgs20	0.26735	25.81273	6.381256E-147	1.705742E-145
Significant	ENSMUSG00000033793	Atp6v1h	0.08945	8.66519	4.507684E-18	2.4447E-17
Significant	ENSMUSG00000025907	Rb1cc1	0.03207	3.18	1.472766E-3	2.736244E-3
	ENSMUSG00000090031	4732440D04Rik	0.01868	1.94077	5.228645E-2	7.481425E-2
	ENSMUSG00000033740	St18	0.00079	0.19436	8.458957E-1	8.756633E-1
	ENSMUSG00000051285	Pcmtd1	0.00886	0.96008	3.370163E-1	4.027151E-1

Figure 4. *Output table of the main table after the Monocle3 Autocorrelation analysis in OmicsBox.*

Actions: Summary Report

Generates a summary report of the analysis.

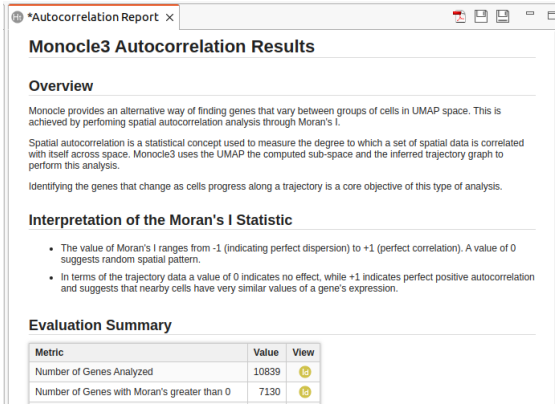


Figure 5. *Detailed summary of the results and the parameters used during the analysis.*

Actions: Update Tags

Update tags based on the results columns.

1. Select Statistic: Choose between the Moran's I and the Moran's I Test Statistic.
2. Select direction: Choose the direction for the selected statistic (Moran's I or the Moran's I Test Statistic).
3. Significance Criteria: Choose between P-Value or Q-Value (Adjusted P-Value).
4. Threshold: Select the threshold for the selected significance criteria.

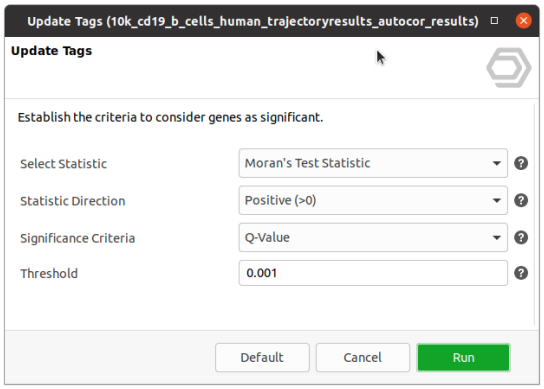


Figure 6. *Multiple options to update tags.*

Actions: Fisher's Exact Test

Please visit the section for Fisher's Exact Test of the OmicsBox Manual for more details.

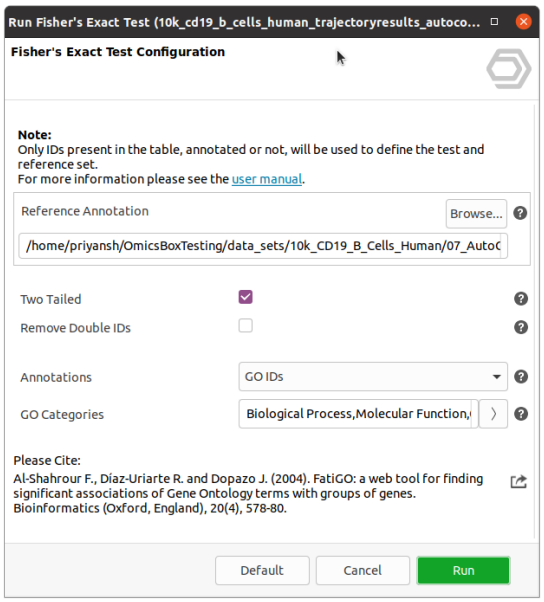


Figure 7. *Fisher's Exact Test Wizard to perform overrepresentation analysis of significantly tagged genes.*

Single Cell RNA-Seq Differential Expression Analysis

Introduction

This tool is designed to perform a Differential Expression Analysis from Single-cell RNA-seq data. It can be performed both after the Single-cell RNAseq Clustering or the Trajectory Analysis, since these tools assign each cell to a group (a cluster in the former and a pseudotime range in the latter).

This tool is based on the R package EdgeR. Please cite EdgeR as: Robinson MD, McCarthy DJ and Smyth GK (2010). "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data." Bioinformatics, 26, pp. -1.

Run scRNA-seq Differential Expression Analysis

From the **Side Panel** of a scRNA-seq Clustering object or Trajectory Analysis object, go to **Actions → Differential Expression**.

Input

The inputs necessary to perform a Differential Expression analysis are a count table and an experimental design. A count table is a table with genes in rows, and cells in columns with each value corresponding to the gene expression level. The experimental design specifies which group belongs to each of the cells. The two inputs are automatically retrieved from the scRNA-seq Clustering or Trajectory Analysis objects.

Configuration 1. Filtering and Normalization Genes Filtering

This step is thought to remove genes with low counts from the analysis. These genes could interfere with some of the statistical approximations of the analysis and don't apport meaningful information. The filtering can be applied in two different ways (Figure 1), depending on the value given:

- **Counts Per Million.** Filtering is performed on a count-per-million (CPM) basis to account for differences in library size between cells. For example, a CPM of 1 corresponds to a count of 6 in a cell with 6 million reads.
- **CPM Filter**. Establish the minimum CPM. Set this parameter to 0 to not filter.
- **Cells Reaching CPM Filter**. Set a minimum number of cells in which the gene's CPM is above the filter level. If this value is set to e.g. five, at least 5 of the cells have to be above the given CPM. Set it to 0 to not filter.
- **Raw Counts.** The values introduced in the parameters below refer to raw expression values. In summary, the strategy keeps genes with at least a minimum reads count in a worthwhile number of cells. More details can be found in EdgeR's Documentation.
- **Minimum Sample Count**. Keep genes that have at least this minimum of counts in at least n cells, where n is the smallest group size. For example, in the case of testing clusters, the n would be the size of the smallest cluster.
- **Minimum Total Count**. Keep genes that have at least this total number of counts across all the cells.

Normalization

Here the normalization takes the form of scaling factors for library sizes that enter into the statistical model. These correctional factors are used to compute effective library sizes. For further details please refer to the EdgeR User's Guide. You can select the normalization method to be used:

- **TMM (Trimmed Mean of M values)**: In this method, weights are obtained from the delta method on Binomial Data.
- **TMM with Zero Pairing**: This is a variant of TMM that should perform better for data with a high proportion of zeros.
- **RLE (Relative Log Expression)**: Scale factors are the median ratio of each sample to the median library (geometric mean of all samples).
- **Upper-quartile**: 75% quantile for the counts for each library is used to calculate the scale factors.
- **None:** All normalization factors are set to 1, so no normalization is performed.

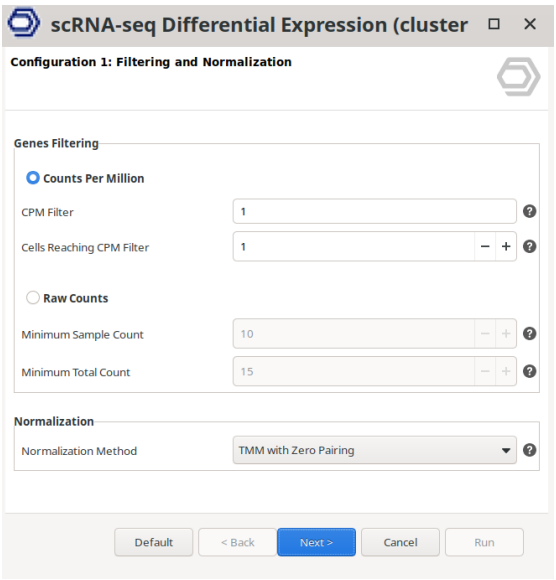


Figure 1. *Filtering and Normalization Page.*

Configuration 2. Metadata

This page shows the experimental design stored in the input object (Figure 2). It contains each of the samples or count matrices present in the object with the factors and conditions specified by the user.

In addition, it is possible to specify the column containing **Biological Replicates**. In this case, a **pseudobulk** analysis will be performed. That means aggregating the cell counts belonging to the same biological replicate and cluster (Figure 3). If not, each cell would be treated as a replicate. It is highly recommended to perform a pseudobulk approach in the presence of biological replicates (Squair et al., 2021).

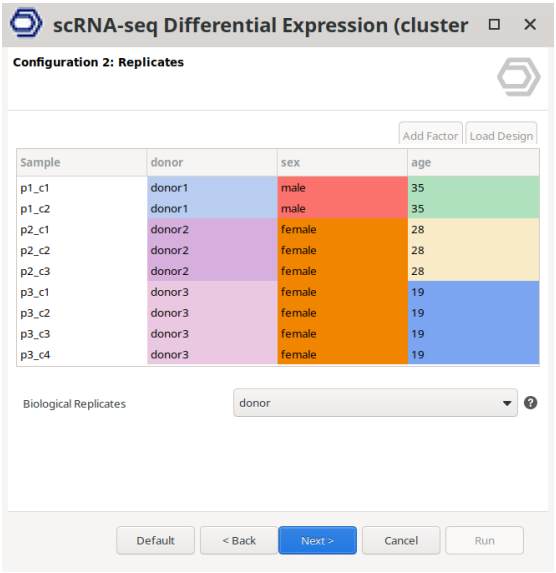


Figure 2. *Metadata Page*

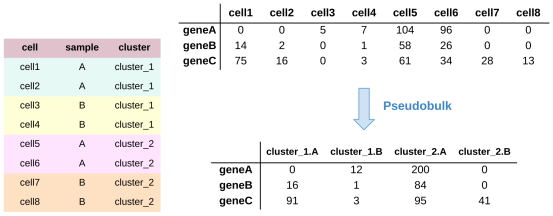


Figure 3*. Pseudobulk approach.*

Configuration 3. Design.

This page allows configuring the design for the differential expression test (Figure 4). Please refer to the blog "Tutorial: Single-cell RNA-Seq Differential Expression Analysis " for a detailed explanation of how to effectively configure the analysis.

Simple Design

This will test for differential expression taking into account only one factor.

- **Test Contrasts Separately**. If checked, one DE test will be performed for each of the conditions specified in "Primary Contrast Conditions". Otherwise, only one DE test will be performed taking as contrast all the specified conditions together.
- **Primary Factor**. Select which factor (or column) from the metadata to test for differential expression.
- **Primary Contrast Conditions**: select which condition(s) to use as contrast.
- **Primary Reference Conditions**: select which condition(s) to use as reference.
- **Blocking Factor**. Adjust for baseline differences of the selected experimental factor. Please refer to the "3.4.2 Blocking" section of edgeR's User Manual for a detailed description.

Notice that if the option "Test Contrasts Separately" option is checked, it is possible to select the same condition in the primary contrast and reference. However, during each of the test the condition selected as contrast won't be used in the reference group.

Multifactorial Design

This will test for differential expression between cells belonging to the Primary Contrast Condition + Secondary Contrast Condition against cells belonging to Primary Contrast Condition + Secondary Reference Condition. Only available if the metadata contained in the object has more than one factor. For example, if an experimental design has been provided with the Merge Single Cell Counts.

- **Secondary Factor**. Select which factor (or column) from the metadata to use as the secondary factor.
- **Secondary Contrast Conditions**: select which condition(s) to use as contrast.
- **Secondary Reference Conditions**: select which condition(s) to use as reference.

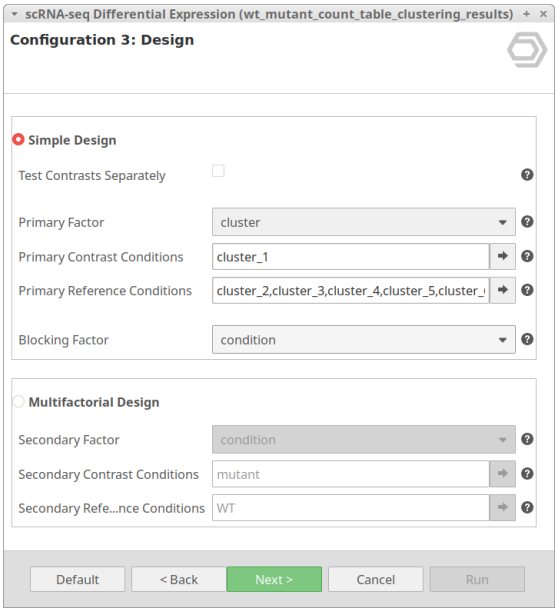


Figure 4. *Design Page.*

Results

When the analysis finishes, the Single-Cell Differential Expression (SCDE) results are opened in a new tab (Figure 5). The results table shows the differential expression statistics, where each row corresponds to a contrast and a feature:

- **Tags**: Indicate whether a gene is considered upregulated (FDR ≤ 0.05, logFC ≥ 0) or downregulated (FDR ≤ 0.05, logFC ≥ 0).
- **Contrast**: which conditions from the primary factors have been used as contrast.
- **Reference**: which conditions from the primary factors have been used as reference.
- **Feature**: feature used for counting in the input Count Table and for DE testing (eg. gene, exon, transcript, etc.).
- **FDR**: False Discovery Rate calculated by the Benjamini-Hochberg method (multiple hypothesis testing corrections).
- **logCPM**: The average log2-counts-per-millions.
- **logFC**: A measure that describes how much the expression changes between conditions (log2-fold-changes are shown).
- **LR**: Likelihood ratio statistic for the GLM (Likelihood Ratio Test).
- **F**: Quasi-likelihood F-statistic for the GLM (Quasi Likelihood F-test).
- **PValue**: raw p-value.

Genes that have not passed the filtering step are not shown in the new tab.

In addition, a Summary report and a chart showing an overview of the results are generated as well.

The **Summary report** (Figure 6) contains general information about the analysis, divided into these sections:

- Dataset Overview: shows the total present in the starting count table, the filtered, and the total in the final count table number of features.
- Results: shows the number of features considered UP and DOWN regulated in the entire project and for each of the primary contrast conditions. In addition, it is possible to obtain the list of UP or DOWN features of each contrast by clicking on the Id list button.
- Analysis Parameters: shows the parameters used for the analysis.

The **Results Overview** chart (Figure 7) shows the total number of features present in the analysis as well as the number of UP and DOWN features for each of the primary contrast conditions.

Tip: it is possible to remove the "Total" column from the chart by applying the rigth filters in the Side Panel. On the "Filtering Options" section, select the following configuration:

- Filter method: Absolute Value.
- Show only: Lower Than.
- Threshold: specify a value lower than the "Total" number of features, but greater than the rest.
- Show others: disabled.

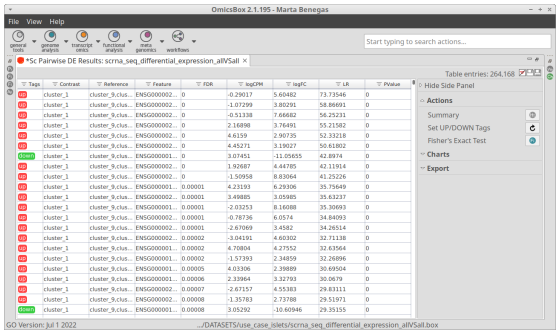


***Figure 5**. Single-cell Differential Expression results.*

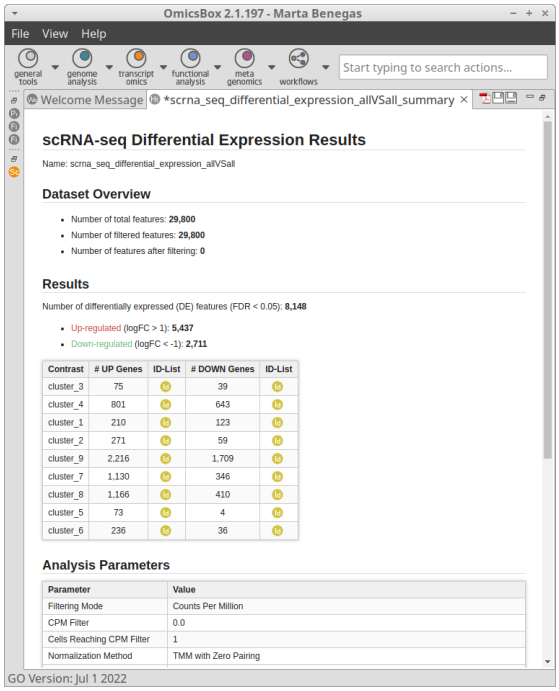


***Figure 6**. Single-cell Differential Expression results summary.*

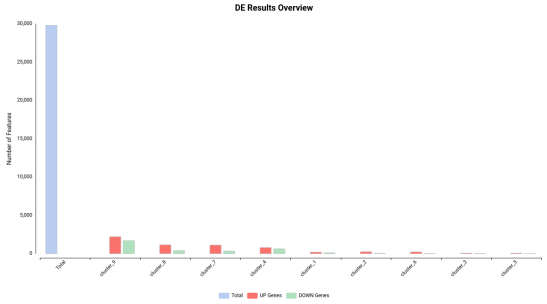


Figure 7. *DE Results Overview chart.*

Side Panel

Actions Summary Report

It shows the Summary report previously explained in the above "Results" section (Figure 6).

Rename Features

This option allows modifying the sequence IDs in the Feature column using different methods:

- **Add**: Add a prefix or sufix to all IDs in the table.
- **Replace**: Replace specific text within the IDs. The text to be replaced must be defined in the Find parameter using a regular expression (regex).
- **Mapping**: Use a mapping file to rename features. The mapping file must be a tab-separated text file with two columns: the first column contains the original feature IDs from the dataset, and the second column contains the new feature names. If duplicate IDs occur during renaming, you can define how they are handled:
  - Sum Rows: Combine counts for all matching features.
  - First Row: Retain only the counts of the first occurrence.

Set UP/DOWN Tags

It re-assigns the UP and DOWN labels based on different filtering cutoffs (Figure 8). Tags will be updated, and the result section of the Result Summary and statistical charts will change according to the new cutoffs.

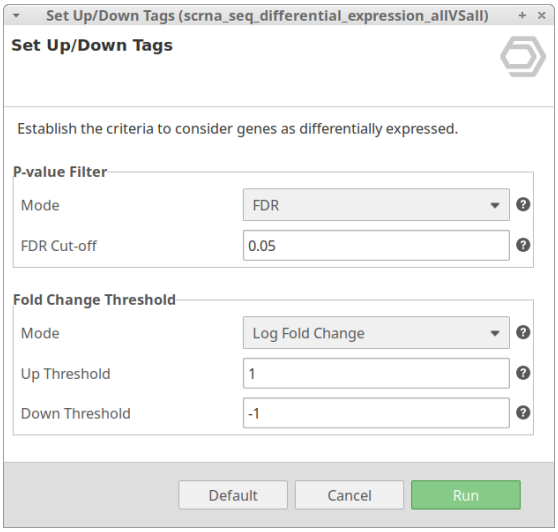


Figure 8*. Set Up/Down Tags wizard.*

Fisher's Exact Test

Fisher's Exact Test can be used to find GO terms that are over and under-represented in a set of genes (test set) with respect to a reference group (reference set). In this case, the test set is composed of all the features tagged as UP or DOWN and belonging to the primary contrast condition specified in "Contrast to Test" (Figure 9). Once

finished, it will open the Fisher's Exact Test results in a new tab (Figure 10). Please refer to the Fisher's Exact Test section of the manual for further details about the analysis and the parameters.

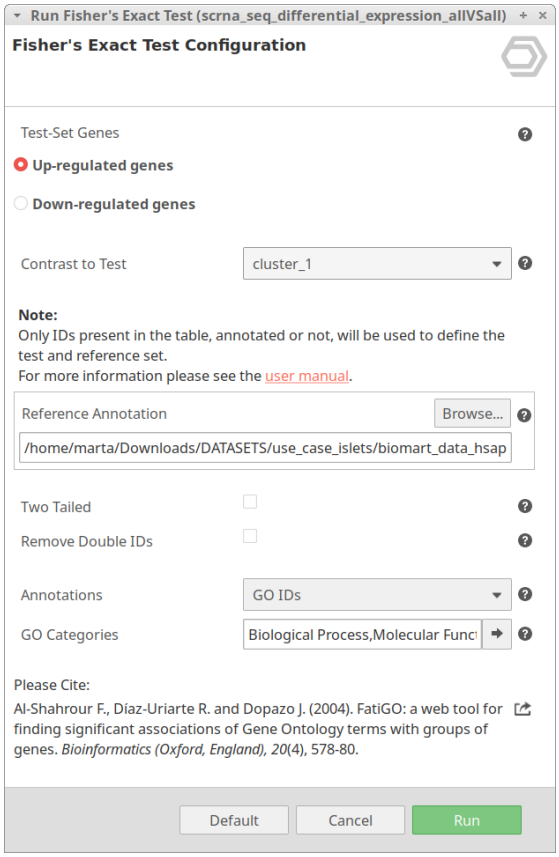


***Figure 9**. Fisher's Exact Test from scRNA-Seq Differential Expression results.*

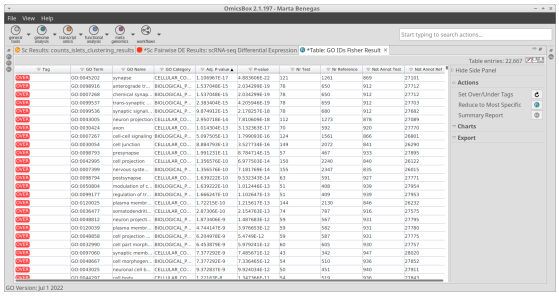


***Figure 10**. Fisher's Exact Test results from Single-cell Differential Expression.*

Charts Heatmap

A heatmap is a two-dimensional visual representation of data in which numerical values of points are represented by a range of colors. In this heatmap, rows correspond to the top differentially expressed features, columns to contrast conditions, and values to mean feature expression level.

It is possible to configure the visualization in the wizard (Figure 11). Firstly, it allows deciding which genes to plot

- Top N Differentially Expressed Genes. Features are ranked according to the FDR and then the top N is selected, where N can be set in the "Nº of DE Genes" parameter.
- ID List. Features specified in the list will be plotted. A text file or an ID-Lis Object must be specified in the "ID List" parameter.

In addition, how to plot the data can be configured as well:

- Expression Data. Which type of data use for plotting: raw counts or CPM (Count Per Million).
- Logarithm: if checked, it applies the log2 for each expression value.
- Z-Score: if checked, it applies the z-score.

Results Overview

This tool generates the chart explained in the above "Results" section (Figure 7).

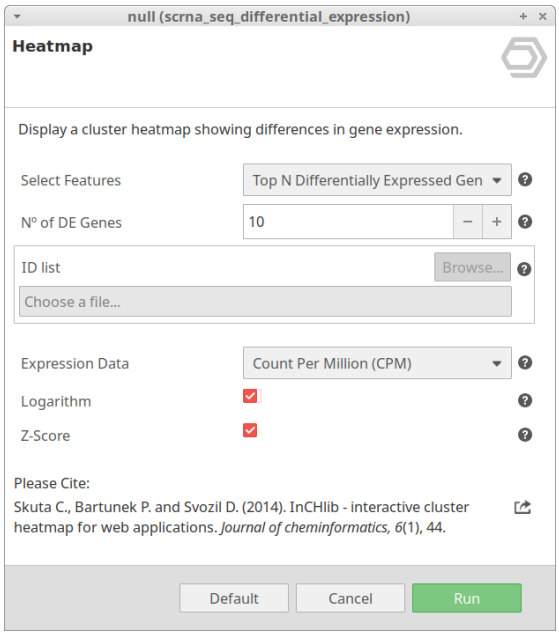


Figure 11. *Heatmap configuration wizard.*

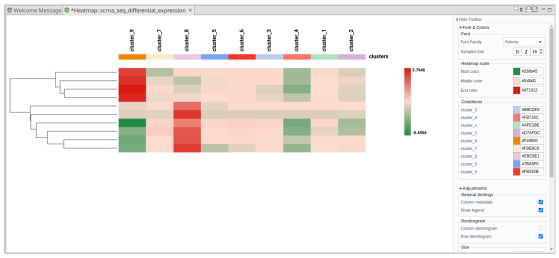


Figure 12. *Heatmap plot per condition.*

Export

Export data with the generic Export Table.

4.4.13 Long Read Data Analysis Tools

Long Read Data Analysis Tools

The Transcriptomics module offers a variety of tools specifically designed for the analysis of long-read RNA-sequencing data from PacBio or ONT platforms.

- **Long-Read Alignment with minimap2:** Minimap2 is a sequence alignment tool designed to efficiently and accurately map long and noisy DNA or RNA sequences against a reference genome or sequence collection. It uses a 'minimizer' indexing approach to rapidly identify potential alignment anchors, which are then refined through dynamic programming to generate accurate and sensitive alignments.
- **Transcript Identification and Quantification:** OmicsBox offers a variety of tools for the identification and quantification of transcripts from long-read RNA-sequencing data, suitable for different use-cases.
    - **PacBio-based Identification with IsoSeq:** PacBio's IsoSeq pipeline preprocesses PacBio single-molecule sequencing data and defines transcript models. This composable workflow combines existing tools and algorithms with a novel clustering technique to handle the increasing data output from PacBio sequencing platforms.
    - **Identification and Quantification with FLAIR:** FLAIR enables transcriptome reconstruction and quantification from long-read RNA sequencing data. During reconstruction, it uses reference annotations and/or short-read data to correct splice junctions observed in long reads, then identifies both known and novel transcript isoforms. For quantification, FLAIR can map long reads to either a newly reconstructed transcriptome or a provided reference transcriptome.
    - **Identification and Quantification with IsoQuant:** IsoQuant performs genome-based analysis of long RNA reads, enabling reconstruction and quantification of transcript models with high precision and good recall. When a reference annotation is provided, IsoQuant assigns reads to annotated isoforms based on intron-exon structure and performs quantification at both gene and isoform levels.
    - **Reference-free Isoform Reconstruction with the isON-pipeline:** The isON-pipeline reconstructs transcriptomes from long-read sequencing data (PacBio or ONT) without requiring reference annotations or genomes. This three-component pipeline (isONclust3, isONcorrect, and isONform) is particularly well-suited for non-model organisms.
- **Curation of Long-Read Transcriptomes with SQANTI3:** SQANTI3 enables quality control and filtering of custom transcriptomes generated from long-read RNA sequencing data. It compares these transcriptomes against reference transcriptomes and incorporates orthogonal data including short reads, CAGE peaks, polyA peaks, and polyA motifs.
- **Combining Transcriptomes with TAMA Merge:** TAMA Merge combines multiple transcriptome annotations into a single, unified transcriptome. This tool is particularly useful when transcriptome reconstruction has been performed separately on individual samples that need to be integrated.

Long-Read Alignment

INTRODUCTION

Minimap2 is a versatile sequence alignment tool widely used in genomics and transcriptomics research. It efficiently aligns DNA or RNA long reads to a reference genome or among them, allowing for accurate identification of sequence similarities and aligning reads to a reference. Minimap2's speed, sensitivity, and ability to handle long reads make it an essential tool for tasks like genome assembly or transcriptome reconstruction. With its diverse range of applications, Minimap2 provides researchers with a powerful tool to explore genomic data and gain deeper insights into the complex biological processes underlying diverse organisms.

Minimap2 follows a typical seed-chain-align procedure as is used by most full-genome aligners. It begins by identifying short seed matches between the query sequence and the reference genome, using a hash minimizer index for rapid retrieval. Then, it constructs longer alignments by chaining and extending the seed matches, considering their quality and proximity. Minimap2 employs bidirectional extension and dynamic programming to accurately align sequences, considering parameters such as gaps and mismatches.

Please cite Minimap2 as:

Li, Heng. "Minimap2: pairwise alignment for nucleotide sequences." *Bioinformatics* 34.18 (2018): 3094-3100.

RUN MINIMAP2 TO ALIGN SEQUENCES

Minimap2 can be found under **Transcriptomics → Long-Read Analysis → Long-Read Alignment with Minimap2.**The wizard consists of 5 pages and allows to define the input and output options as well as the analysis parameters (Figure 1, Figure 2, Figure 3, Figure 4 and Figure 5).

Input

- First of all, FLAIR requires some necessary files:
- **Long-Reads Files:** FASTA/Q files containing long reads proceeding from PacBio or ONT technologies.
- **Reference Genome:** FASTA file with the reference genome.

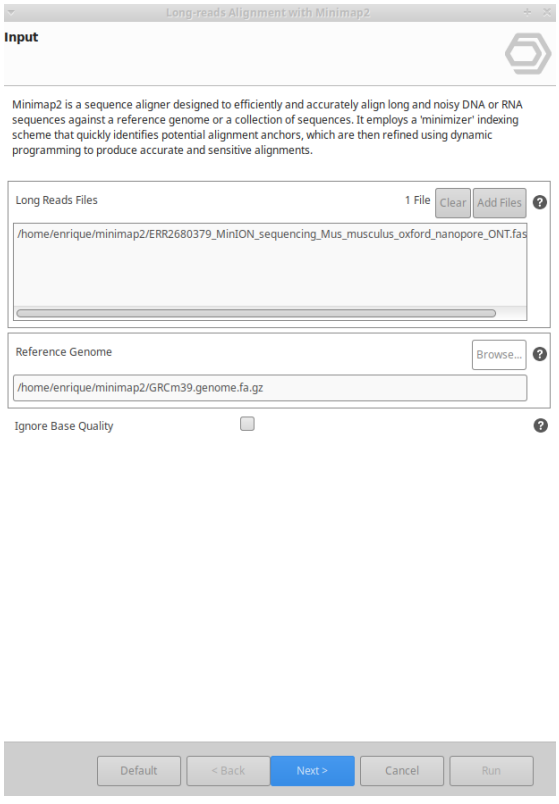


Figure 1. *Input Page*

Configuration 1

- **Ignore Base Quality:** check it to ignore base quality in the input file.
- **Preset Options:**

- **Preset Options:**
  - Map ONT Reads (Default): align noisy long reads of ~10% error rate to a reference genome. This option can be used with ONT long reads.
  - Map PacBio HiFi Reads: align PacBio high-fidelity (HiFi) reads to a reference genome.
  - Map PacBio CLR Reads: align older PacBio continuous long (CLR) reads to a reference genome.
  - Long-Read Spliced Mapping: long-read spliced alignment. In the splice mode:
  - Long deletions are taken as introns and represented as the '**N**' CIGAR operator.
  - Long insertions are disabled.
  - Deletion and insertion gap costs are different during chaining.
  - The computation of the '**ms**' tag ignores introns to demote hits to pseudogenes.
  - Long-Read Splice Alignment for PacBio CCS Reads: This preset is similar to the last one but it's designed for high-quality data or small genomes.
  - **Indexing Options:** Indexing referes to the first step of minimap2 and the majority of full-genome aligners. Minimap2 obatins minimizers and index them in a hash table. When chosen a preset index, recommended index parameters are already chosen. Nevertheless, you are able to modify these parameters:
- **Minimizer k-mer Length:** Number of nucleotides for each minimizer. A minimizer is the smallest k-mer in a window of consecutive k-mers.
- **Minimizer Window Size:** Number of consecutive k-mers. It is recommended that this value represents 2/3 of the k-mer length.
- **Use HPC Minimizers:** This option is recommended if you are going to map PacBio Long Reads. A Homopolymer-Compressed Minimizer (HPC) sequence is constructed by contracting homopolymer runs to a single base. An HPC minimizer is a minimizer on the HPC sequence.
- **Seeding Options:** Seeding refers to the second step of minimap2. In this step, minimizers are taken as seeds to initiate alignments. This is an efficient method to sample k-mers from genomic sequences that unconditionally preserve sufficiently long matches between sequences.
- **Ignore Most Frequent Minimizers:** If a fraction, ignore the top fraction of most frequent minimizers. If an integer, ignore minimizers occurring more than that number of times.
- **Lower bound of K-Mer occurrences:** Prevents excessively small numbers of ignored minimizers.
- **Upper bound of K-Mer occurrences.** Prevents excessively big numbers of ignored minimizers.
- **Ignore Most Frequent Minimizers:** Discard a query minimizer if its occurrence is higher than that fraction of query minimizers and the reference occurrence threshold. Set 0 to disable.
- **Basepairs to Sample Minimizer:** Number of basepairs before sampling a high-frequency minimizer.

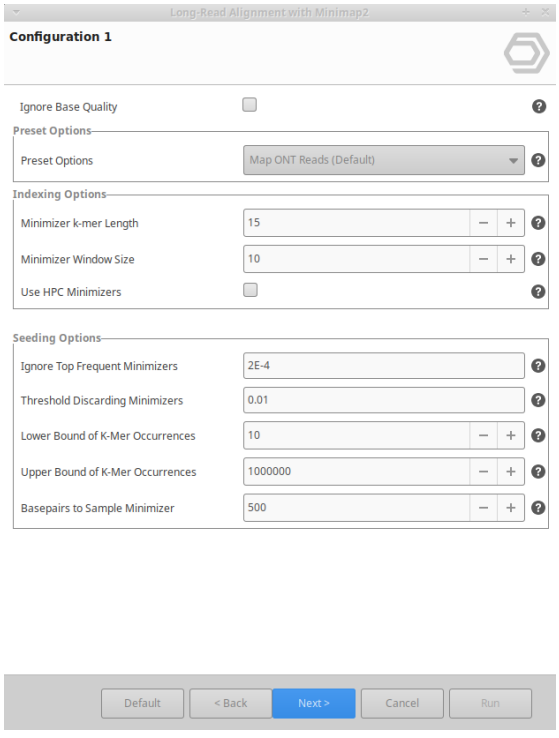


Figure 2. *Configuration 1 Page*

Configuration2

- **Chaining Options:** Chaining is the next step after seeding. This step consists of elongating the seed anchored before.
- **Basepairs to Stop Chain Elongation:** Number of basepairs to look for minimizers after stopping chain elongation.
- Chain Gap Scale: Scale of gap cost during chaining.
- **Bandwidth for chaining:** Bandwidth used for initial chaining and alignment extension.

- **Minimum Number of Minimizers in a Chain:** Discard chains consisting of less than this number of minimizers.
- **Minimum Chaining Score:** Minimum chaining score not to discard a chain. Chaining score equals the approximate number of matching bases minus a concave gap penalty. It is computed with dynamic programming.
- **Chain Gap Scale:** Scale of gap cost during chaining.
- **All-vs-All Overlapping:** Primarily used for all-vs-all read overlapping.
- **Secondary Alignment Options:**
- **Retain Secondary Alignments:** Check this option to get secondary alignments. Secondary chains are segments mapped in the same place as another chain with better quality**.**
- **Retain All Chains:** Retain all chains and don't attempt to set primary chains.
- **Fraction to Mark a Chain as Secondary:** Mark as secondary a chain that overlaps with a better chain by this fraction or more of the shorter chain.
- **Minimal Secondary-to-Primary Score Ratio:** Between two chains overlapping the fraction set in the previous parameter, the chain with a lower score is secondary to the chain with a higher score. If the ratio of the scores is below this fraction, the secondary chain will not be outputted.
- **Number of Secondary Alignments:** Maximum number of secondary alignments.
- **Splice Options:**
- **Splice Mode:** Enable the splice alignment mode. In this mode, the chaining gap cost distinguishes insertions to and deletions from the reference.
- **Maximum Gap on the Reference:** Maximum number of nucleotides in a insertion or deletion.
- **Where to Find Splice Sites:**
    - Both Strands: find splice sites in both strands, not only in the strand that has been transcribed.
    - Transcript Strands: find splice sites only in the transcript strand.

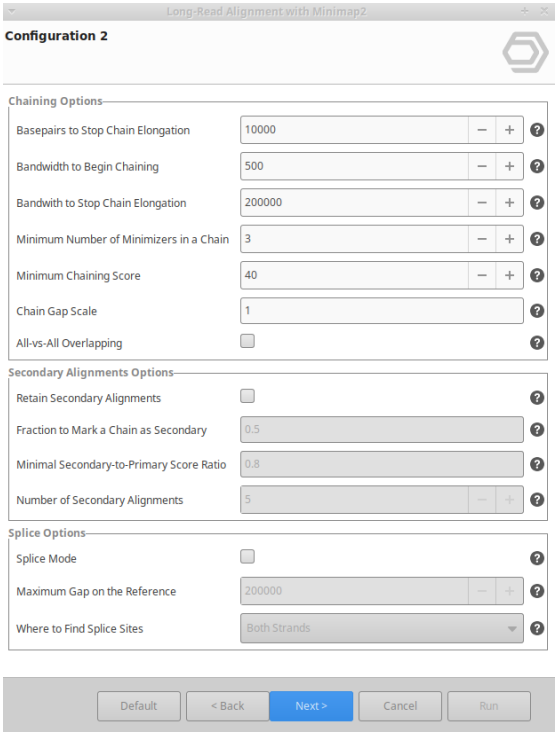


Figure 3. *Configuration 2 Page*

Configuration 3

- **Alignment Options:**
- **Matching Score:** Score of a nucleotide in the query matching a nucleotide in the reference.
- **Mismatch Penalty:** Penalty when a nucleotide in the query and a nucleotide in the reference do not match.
- **First Gap Open Penalty:** Penalty when a first gap is opened in the alignment.
- **Subsequent Gap Open Penalty:** Penalty for the next gaps after the first one is opened in the alignment.
- **First Gap Extension Penalty:** Penalty for each nucleotide that the first gap is extended.
- **Subsequent Gap Extension Penalty:** Penalty for each nucleotide that the next gaps are extended.
- **Non-Canonical Splice Site Penalty:** Cost for a non-canonical GT-AG splicing.
- **End Bonus Score:** Score bonus when alignment extends to the end of the query sequence.

- **Ambiguous Base Penalty:** Penalty of a mismatch involving ambiguous bases.
- **Use Splice Flank:** Assume the next base to a GT donor site tends to be A/G (91% in human and 92% in mouse) and the preceding base to a AG acceptor tends to be C/T. This trend is evolutionarily conservative, all the way to S. cerevisiae. Specifying this option generally leads to higher junction accuracy by several percents, so it is applied by default with the splice mode. However, the SIRV control does not honor this trend (only ~60%). This option reduces accuracy. If you are benchmarking minimap2 on SIRV data, please do not use this option.
- **Splice Junctions Information:**
- **Use Junction File:** Check this option if you want to use a Junction File.
- **Junction Bonus:** Bonus Score when a Junction is present in the Junction File.
- **Junction Bed File:** You can introduce a BED, BAM or GTF file as Junction File. The easiest option, which is also efficient, is to map short reads using STAR and then introduce that BAM as input here.

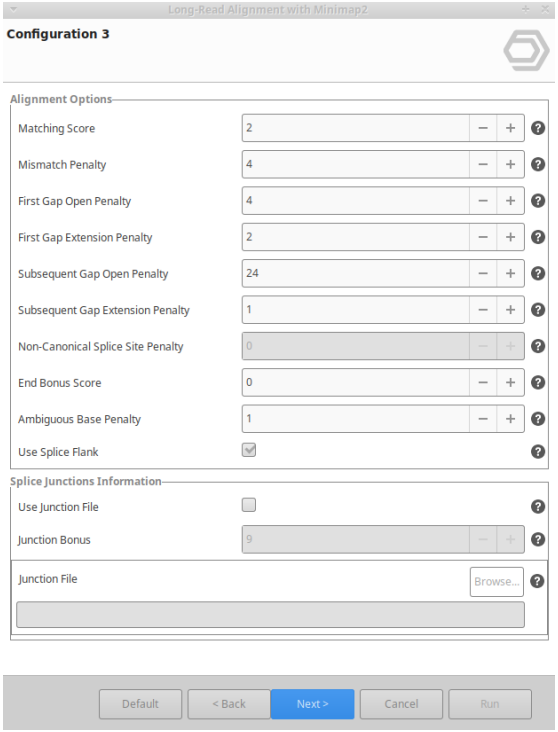


Figure 4. *Configuration 3 Page*

Output

- **BAM file:** path to save BAM files with aligned long reads. Each fastq file with generate one BAM file.
- **Output (BAM) Options:**
- Make CIGAR: Generate CIGAR in BAM file.
- Make Long CIGAR: Write CIGAR with >65535 operators at the CG tag. Older tools are unable to convert alignments with >65535 CIGAR ops to BAM. This option makes minimap2 SAM compatible with older tools. Newer tools recognizes this tag and reconstruct the real CIGAR in memory.
- Add MD Tag to BAM: Check this option to add the MD tag, important to do Variant Calling without the reference genome.

• Add CS Tag to BAM: Check this option to add the CS tag. This tag is similart to the MD tag as it shows nucleotidic differences between the reference and the read.

Figure 5. *OutPut Page*

RESULTS

The main outputs are the BAM files. A BAM file (*.bam) is a compressed binary version (BGZF format) of a SAM file that is used to represent aligned sequences. SAM is a TAB-delimited text format consisting of a header section and an alignment section. Header lines start with '@', while alignment lines do not.

In addition, a report and two charts are generated with complementary information. The report (Figure 6) shows a summary of the DNA-Seq Alignment results. This page contains information about the reference genome sequences, the input FASTQ files, and a results overview. The last section is divided into several subsections: globals, paired information, ACTG content, coverage, mapping quality, insert size, mismatches, and indels.

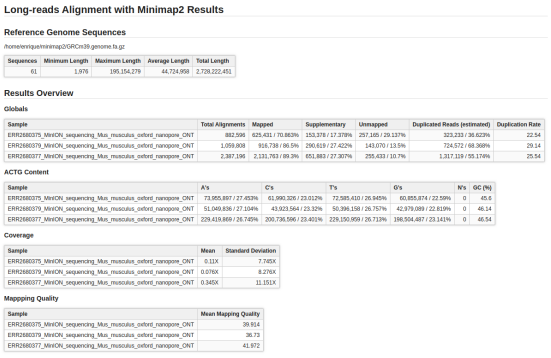


Figure 6. *Minimap2 Summary Report*

The bar charts (Figures 7 and 8) show the number of mapped and unmapped reads of each input file.

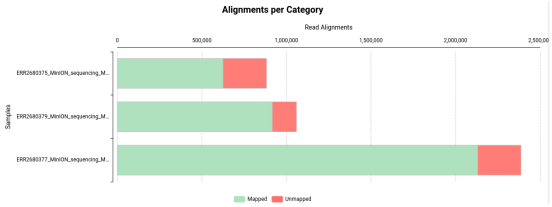


Figure 7. *Absolute Alignments Per Category Per Sample*

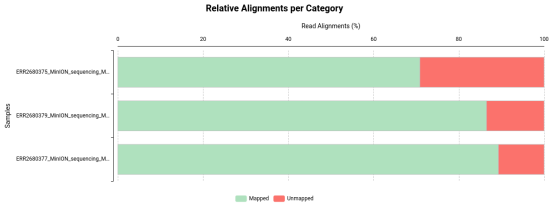


Figure 8. *Relative Alignments Per Category Per Sample*

Transcript Identification and Quantification

LONG READ TRANSCRIPT IDENTIFICATION AND QUANTIFICATION

OmicsBox offers a variety of tools for the identification and quantification of transcripts from long-read RNA-sequencing data, suitable for different use-cases. We strongly recommend that any such workflow should be followed by careful curation of discovered transcripts through **SQANTI3**. After curation, FLAIR or IsoQuant can re-quantify the reads on the curated transcriptome.

- **PacBio-based Identification with IsoSeq:** PacBio's IsoSeq pipeline preprocesses PacBio single-molecule sequencing data and defines transcript models. This composable workflow combines existing tools and algorithms with a novel clustering technique to handle the increasing data output from PacBio sequencing platforms. This tool is recommended to be used primarily for pre-processing of PacBio data, supporting the following steps:
    - **CCS Calling** from subreads
    - **Kinnex-Demultiplexing** with skera
    - **Primer removal and demultiplexing** with lima
    - **Refine** to trim poly(A) tails and remove artificial concatemers (chimeric reads)

  Further, it can also perform transcript identification. Compared to other tools such as FLAIR and IsoQuant, IsoSeq is notably more permissive in defining a wide variety of transcript models, leading to high levels of redundancy and potential artifacts. While we recommend that any transcriptome reconstruction workflow should be followed by careful curation with SQANTI3 regardless of the tool used, this becomes especially important when identifying isoforms with IsoSeq.
- **Identification and Quantification with FLAIR:** FLAIR enables transcriptome reconstruction and quantification from long-read RNA sequencing data. During reconstruction, it uses reference annotations and short-read data to correct splice junctions observed in long reads, then identifies both known and novel transcript isoforms. For quantification, FLAIR can map long reads to either a newly reconstructed transcriptome or a provided reference transcriptome.

  FLAIR is a top-performing transcriptome reconstruction tool in benchmarks such as the LRGASP challenges. When both reference transcriptome annotations and short-read RNA-seq data are available, it excels in the discovery of novel isoforms. However, without short-read data, it is unable to discover novel splice junctions, limiting the discovery of potential novelty.
- **Identification and Quantification with IsoQuant:** IsoQuant performs genome-based analysis of long RNA reads, enabling reconstruction and quantification of transcript models with high precision and good recall. When a reference annotation is provided, IsoQuant assigns reads to annotated isoforms based on intron-exon structure and performs quantification at both gene and isoform levels.

  IsoQuant has also displayed strong performance in benchmarks such as the LRGASP challenges. While it is generally more conservative in reporting novel isoforms than FLAIR, it can discover novel splice junctions even without access to reference transcriptome annotations or short-read data. However, when available, their use is still recommended.
- **Reference-free Isoform Reconstruction with the isON-pipeline:** The isON-pipeline reconstructs transcriptomes from long-read sequencing data (PacBio or ONT) without requiring reference annotations or genomes. This three-component pipeline (isONclust3, isONcorrect, and isONform) is particularly well-suited for non-model organisms.

  As the isON-pipeline should only be used where a reference genome is not available, it cannot be followed by curation with SQANTI3.

PACBIO-BASED IDENTIFICATION WITH ISOSEQ

Introduction

IsoSeq is a composable workflow of existing tools and algorithms, combined with a new clustering technique, which allows processing the ever-increasing yield of PacBio machines. Starting from subreads or CCS reads, this tool allows identifying transcripts in PacBio single-molecule sequencing data. The IsoSeq pipeline is made up of up to eight steps:

1. **CCS Calling:** Each sequencing run is processed by the **ccs** software to generate one representative circular consensus sequence (CCS) for each ZMW (Zero-mode Waveguide).
2. **Kinnex-Demultiplexing**: Kinnex reads are demultiplexed into individual transcript reads using **skera**.
3. **Primer removal and demultiplexing:** Removal of primers and identification of barcodes is performed using **lima**.
4. **Refine:** This step consists of trimming of poly(A) tails and identification and removal of artificial concatemers (chimeric reads).
5. **Clustering:** Clustering using hierarchical n*log(n) alignment and iterative cluster merging.
6. **Polishing** (optional)**:** Generate per base QVs for transcript consensus sequences and improve results.
7. **Mapping:** clustered reads are mapped to a reference genome.
8. **Collapsing:** the mapped reads are finally collapsed into transcripts in order to define isoforms and obtain a transcriptome.

Please, cite IsoSeq as:

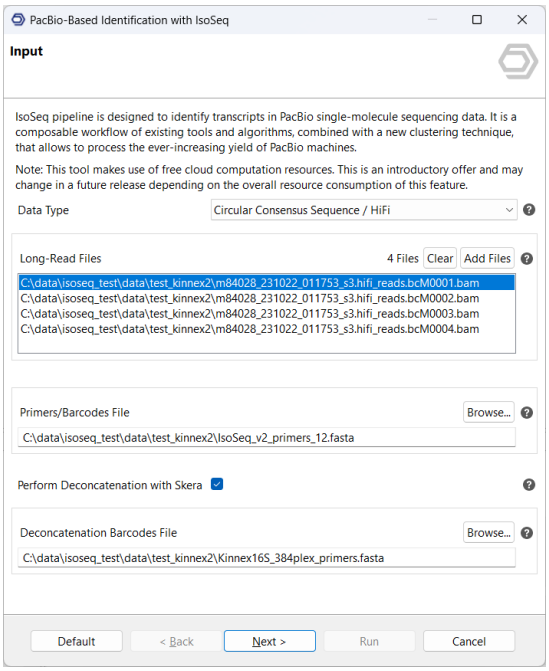
IsoSeq v4. Scalable De Novo Isoform Discovery. Töpfer, A. and Tseng, E. 2018. Retrieved 2024 from https://github.com/PacificBiosciences/IsoSeq

Run IsoSeq

This functionality can be found under **Transcriptomics → Long-Reads Analysis → Transcript Identification → PacBio-Based Identification with IsoSeq3.**The wizard allows adjusting analysis parameters (Figure 1, Figure 2, Figure 3, and Figure 4).

Input

- **Data Type:**
- Subreads: Subreads are the continuous raw sequences produced from a single pass of the polymerase around the circular DNA template.
- Circular Consensus Sequence / HiFi: CCS refers to the high-accuracy consensus sequence derived from multiple subreads of the same circularized DNA molecule. HiFi (High-Fidelity) reads are high-accuracy consensus sequences generated by this CCS technology.
- **Long-Read Files**: Select the files containing PacBio sequencing reads in the selected data type.
- **Primers/Barcodes File**: Specify a FASTA file with primer or barcoded primer sequences. This file will be used in lima in order to remove primers and demultiplex input reads.
- **Perform Deconcatenation with Skera:** Check this option if you want to perform deconcatenation. Skera is used to deconcatenate or split the HiFi reads generated using Kinnex (formerly MAS-Seq) methodology at adapter positions, generating segmented reads (S-reads). For each input BAM file (e.g., HiFi), skera will create a BAM file with deconcatenated reads. A parent HiFi read can contain many S-reads.
- **Deconcatenation Barcodes File:** Specify a FASTA file with Kinnex barcodes to be used by skera.

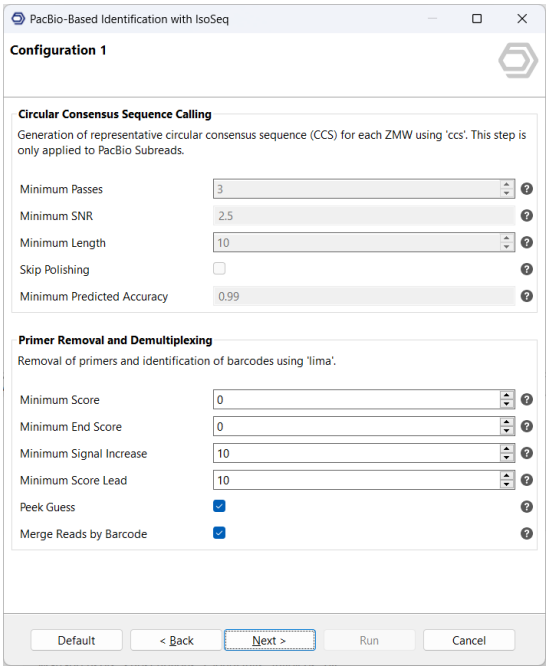


Configuration 1 Circular Consensus Sequence Calling

- **Minimum Passes:** Minimum number of full-length subreads required to generate CCS for a ZMW.
- **Minimum SNR:** Minimum SNR of subreads to use for generating CCS.
- **Minimum Length:** Minimum draft length.
- **Skip Polishing:** Only output the initial draft template (faster, less accurate). It does not refer to the last optional polishing step.
- **Minimum Predicted Accuracy:** Establish the minimum predicted accuracy (0 - 1).

Primer Removal and Demultiplexing

- **Minimum Score:** Reads below the minimum barcode score are removed from downstream analysis.
- **Minimum End Score:** Minimum end barcode score threshold is applied to the individual leading and trailing ends.
- **Minimum Signal Increase:** The minimal score difference, between first and combined, required to call a barcode pair different.
- **Minimum Score Lead:** The minimal score lead required to call a barcode pair significant.
- **Peek Guess:** Try to infer the used barcodes subset, by peeking at the first 50000 ZMWs, whitelisting barcode pairs with more than 10 counts and mean score >= 45. Check this option to remove spurious false-positive signals.
- **Merge by Barcode:** If this option is checked, reads will be merged by barcode. This is useful if the input consists of multiple files, e.g. from multiple SMRTCells, containing multiple barcodes, e.g. when the same samples were ran in multiple SMRTCells.



Configuration 2 Refine

- **Remove Poly(A) Tails:** Check this option if your sample has poly(A) tails. This filters for FL reads that have a poly(A) tail with at least the number of base pairs set in the following parameter. It removes identified tails.
- **Minimum Poly(A) Tail Length:** Establish the minimum poly(A) tail length.

Clustering

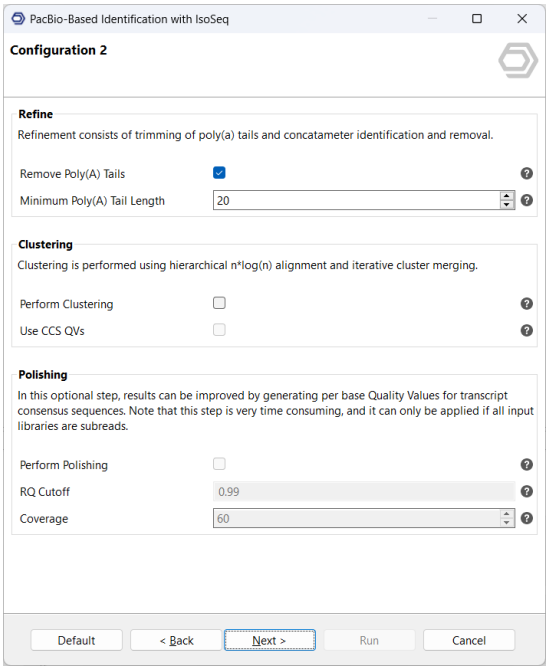
- **Perform Clustering:** Whether the clustering step should be performed. This is required for the subsequent mapping and collapsing steps.
- **Use CCS QVs:** Use CCS QVs. If it is checked, the POA (Partial Order Alignment) Coverage is set to 100.

Polishing

- **Perform Polishing:** If the input files were subreads, this optional step can improve results by generating per base QVs for transcript consensus sequences.

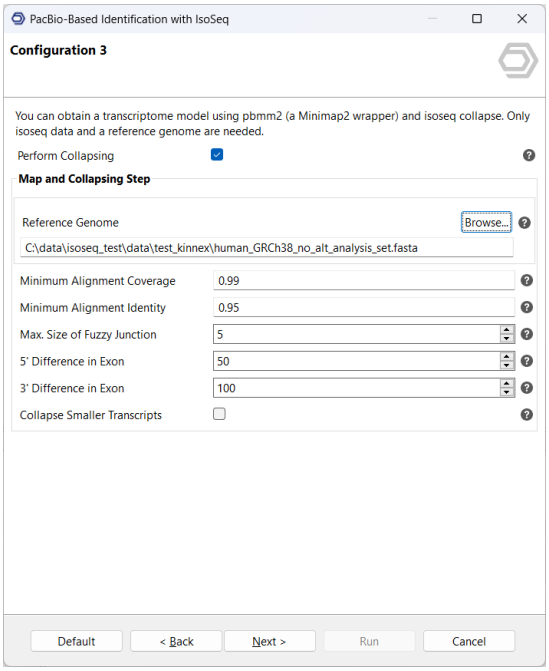
Note that this step is outdated and only offered to support legacy data sets. It is very time consuming and the improvement in quality is often unnecessary.

- **RQ Cutoff:** RQ cutoff for fastx output.
- **Coverage:** Maximum number of subreads used for polishing.



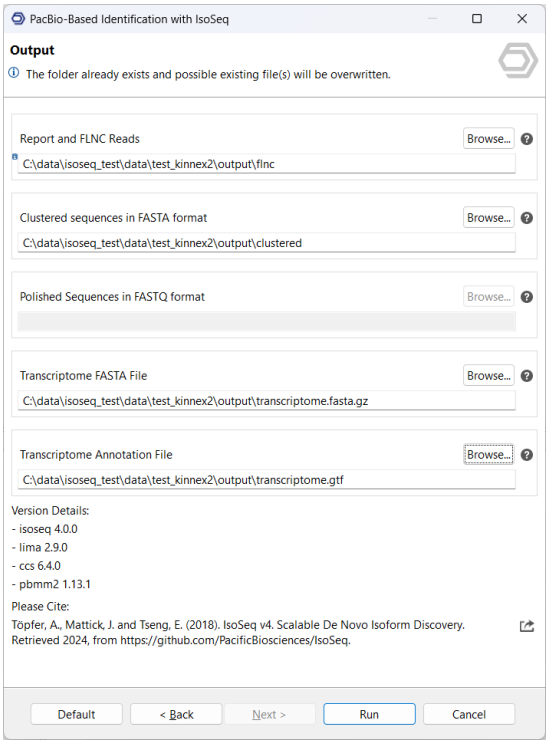
Configuration 3 Map and Collapsing Step

- **Perform Collapsing**: Whether the mapping and collapsing steps should be performed.
- **Reference Genome:** reference to align the clustered reads to make a transcriptome model.
- **Minimum Alignment Coverage:** Ignore alignments with less than minimum query read coverage.
- **Minimum Alignment Identity:** Ignore alignments with less than minimum alignment identity.
- **Max. Size of Fuzzy Junction:** Ignore mismatches or indels shorter than or equal to N.
- **5' Difference in Exon:** Maximum allowed 5' difference on same exon.
- **3' Difference in Exon:** Maximum allowed 3' difference on same exon.
- **Collapse Smaller Transcripts:** Collapse 5' shorter transcripts which miss one or multiple 5' exons to a longer transcript.



Output

- **FLNC Reads:**
- In this directory, the pre-process FLNC (Full-Length Non-Chimeric) reads will be saved.
- If collapsing was performed, an abundance file will also be saved here.
- **Clustered Sequences in FASTA Format**
- In this directory, the clustered high-quality transcript sequences will be saved. This is only available if clustering is performed. If collapsing is performed as well, a clustering report will also be generated.
- **Polished Sequences in FASTQ Format**
- In this directory, the polished sequences will be saved. This is only available if polishing is performed.
- **Transcriptome FASTA File**
- The collapsed transcript sequences. This is only available if collapsing is performed.
- **Transcriptome Annotation File**
- The collapsed transcripts in GTF Format. This is only available if collapsing is performed.

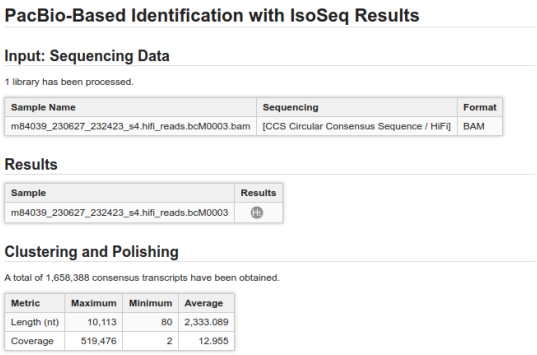


Results

The main output is the clustered/polished .fasta file. It contains the transcripts identified from the input data. Additional BAM and FASTQ files contain the same information in a different format.

The **report.csv** file contains information about how many PacBio reads have contributed to the reconstruction of each transcript.

In addition, a report and a chart are generated with complementary information. The report shows a summary of the IsoSeq results (Figure 6).



PacBio-Based Identification with IsoSeq Results

Input: Sequencing Data

1 library has been processed.

Sample Name	Sequencing	Format
m84039_230627_232423_s4.hifi_reads.bcM0003.bam	[CCS Circular Consensus Sequence / HiFi]	BAM

Results

Sample	Results
m84039_230627_232423_s4.hifi_reads.bcM0003	

Clustering and Polishing

A total of 1,658,388 consensus transcripts have been obtained.

Metric	Maximum	Minimum	Average
Length (nt)	10,113	80	2,333.089
Coverage	519,476	2	12.955

In addition, a report for each input sample can be opened (Figure 7). They contain additional details about the processing of each sample.

PacBio-Based Identification with IsoSeq Results: m84039_230627_232423_s4.hifi_reads.bcM0003

Deconcatenation

Total Reads	6,557,584
Segmented Reads	28,084,667
Mean Length of S-Reads	2,249
Percentage of Reads with Full Array	0%
Mean Array Size	4.283

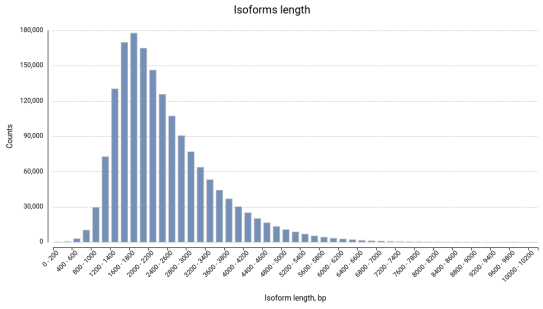
Primer Removal and Demultiplexing

Reads input	28,084,667
Reads above all thresholds (A)	26,872,586
Reads below any threshold (B)	1,212,081
Read marginals for (B)	
Below min length	20,650
Below min score	402
Below min end score	569,945
Below min passes	392
Below min score lead	0
Below min ref span	992,284
Without SMRTbell adapter	0
Wrong different pair	391
Undesired 5p--5p pairs	235,434
Undesired 3p--3p pairs	553,461
Reads for (A)	
With same pair	0
With different pair	26,872,586

Three types of charts are generated:

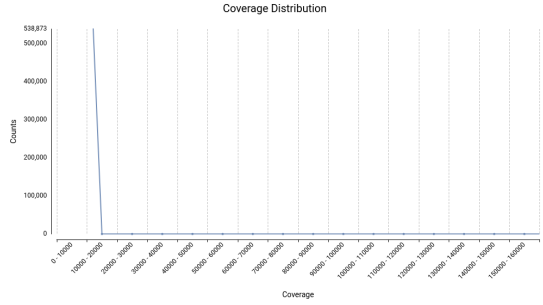
Length Distribution

Show the distribution of the lengths of the resulting transcripts (Figure 8).



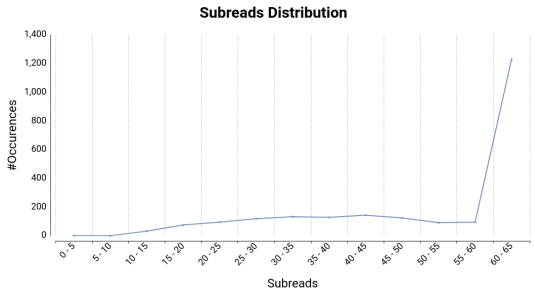
Coverage Distribution

Show the distribution of the coverage, this is, the number of reads supporting each transcript (Figure 9).



Subreads Distribution

This chart is only generated if the polishing step is applied. It shows the distribution of subreads supporting the resulting transcripts (Figure 10).



IDENTIFICATION AND QUANTIFICATION WITH FLAIR

Introduction

Long-read sequencing technologies are becoming increasingly popular for transcriptome analysis because they can capture a full RNA molecule within a single read, enabling more detailed analyses of properties such as alternative splicing. However, whereas short reads have to be assembled in order to form transcript models, long reads require processing in order to separate artifacts and noise from genuine novel transcripts.

FLAIR is a computational pipeline designed to identify both known and novel transcript isoforms in long-read RNA-sequencing data. It consists of the following steps (see Figure 1):

1. *FLAIR-align*: Long reads are aligned to the reference genome using minimap2.
2. *FLAIR-correct*: Information from reference transcriptome annotations and/or short-read data is used in order to correct splice junctions in long reads.
3. *FLAIR-collapse*: Long reads are grouped and *collapsed* by their splice junctions and transcript ends are defined to identify and annotate transcript models.
4. *FLAIR-quantify*: Long reads are quantified on the identified transcript models. In OmicsBox, this step can also be run by itself to 1) quantify long reads based solely on a reference annotation, without discovering novel isoforms; or 2) re-quantify long reads on a custom transcriptome after curation with SQANTI3.

Please cite FLAIR as:

Tang, A. D., Soulette, C. M., van Baren, M. J., Hart, K., Hrabeta-Robinson, E., Wu, C. J., & Brooks, A. N. (2020). Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns. *Nature communications*, *11*(1), 1-12.

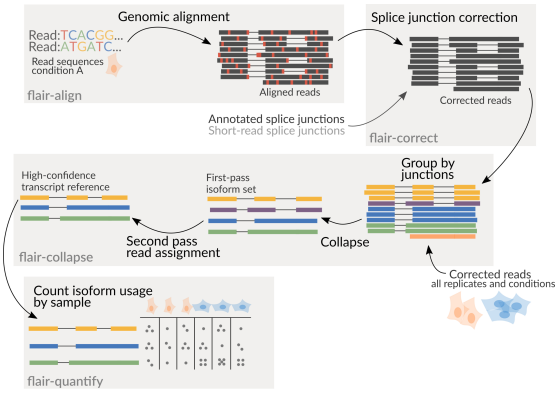


Figure 1. *FLAIR pipeline; graphic adapted from https://flair.readthedocs.io/*

Run FLAIR for Long-Read Isoform Definition

FLAIR can be found under **Transcriptomics → Long-Reads Analysis → Transcript Identification → Identification and Quantification with FLAIR.** The wizard consists of 5 pages and allows the definition of the input and output options as well as the analysis parameters (see Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6).

Required Inputs

This page includes input options for the basic files required by FLAIR:

- **Long Reads Files:** FASTA/Q files containing long reads originating from PacBio or ONT technologies. These should already be pre-processed (e.g. FLNC reads in the case of PacBio).
- **Reference Genome:** FASTA file with the reference genome.

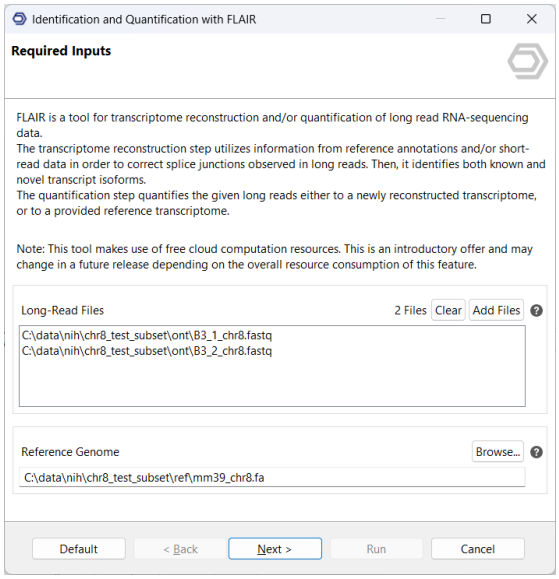


Figure 2. *"Required Inputs" page of the FLAIR wizard in OmicsBox.*

Reconstruction and/or Quantification

Next, the desired analysis has to be configured with the following options:

- **Transcriptome Reconstruction:** Whether or not to perform transcriptome reconstruction, consisting of the FLAIR align, correct, and collapse steps. This step identifies known and novel transcript isoforms and produces a transcriptome in GTF format as its primary output. Performing these steps requires supplying at least one of the following file options: a reference transcriptome annotation in GTF/GFF3 format, and/or short-read data as alignments in BAM format or as SJ.out.tab format as produced by the STAR aligner.
- **Quantify Reads:** Whether or not to perform quantification of long reads to produce a count matrix. When both transcriptome reconstruction and quantification are selected, quantification will be performed on the newly reconstructed transcriptome. When only quantification is selected, quantification will be performed on the provided transcriptome annotation file.
- **Use Annotation File:** Whether or not to supply a transcriptome annotation file.
- **Transcriptome Annotation:** The transcriptome annotation in GTF/GFF3 format. If transcriptome reconstruction is performed, this file will be used to correct splice junctions in long reads. If only quantification is performed, this file will be used as a basis for quantification.
- **Use Short Reads:** Whether or not to supply short-read alignments or splice junctions for the splice junction correction in transcriptome reconstruction.

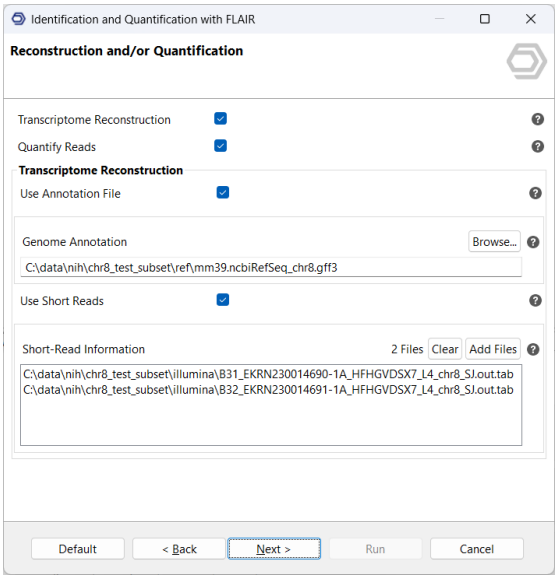


Figure 3. *"Reconstruction and/or Quantification" page of the FLAIR wizard in OmicsBox.*

Alignment

This page defines whether pre-existing alignments are supplied, or whether FLAIR should perform its own alignment in the FLAIR-align step.

- **Existing Alignment:**
- **Use Own Alignment Files:** Whether or not to provide custom alignments. This may be useful if you want to use an aligner other than minimap2 or if you want to configure your alignment in more detail.
- **Aligned Reads:** Alignments of the provided reads in BAM format. These alignments are merged into a single file for the FLAIR correct step and therefore do not necessarily have to correspond 1:1 to the provided read files.

Alignment:

- **Native RNA:** Use native-RNA-specific alignment parameters for minimap2. This flag indicates that the input consists of native RNA sequences rather than pre-processed or adapter-trimmed sequences.
- **Min. Mapping Quality:** Minimum mapping quality score of read alignments to the genome.
- **Retain Secondary Alignments:** Retain that number of secondary alignments from minimap2 (i.e. alignments of the same read in other parts of the genome). Please proceed with caution; changing this setting is only useful if you know there are closely related homologs elsewhere in the genome. It will likely decrease the quality of FLAIR's final results.

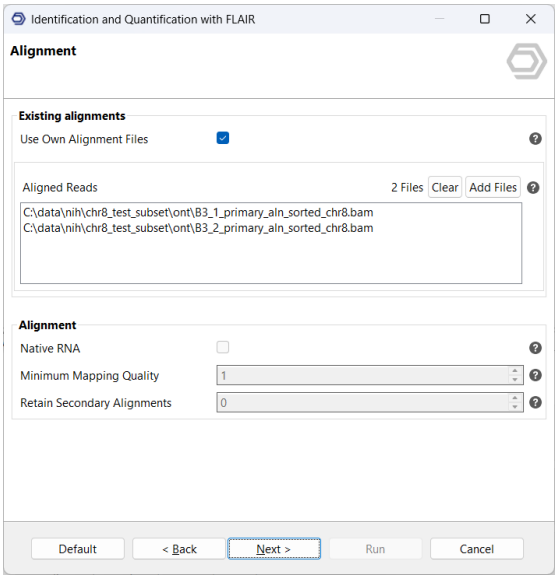


Figure 4. *"Alignment" page of the FLAIR wizard in OmicsBox.*

Configuration

This page allows for the configuration of FLAIR's correct, collapse, and quantify steps.

- **Correct:**
- **Window Size:** Window size for correcting splice sites.
- **Collapse:**
- **Minimum Supporting Reads:** Minimum number of long reads required to call an isoform.
- **Window Size for TSS and TTS:** Window size for comparing transcript starts (TSS) and ends (TTS).
- **Ends Determined at Isoform Level:** When specified, TSS/TTS for each isoform will be determined from supporting reads for individual isoforms rather than from genes.
- **Get TSS and TTS from Supporting Reads:** Do not use TSS/TTS from the input GTF to adjust isoform TSS/TTS. Instead, each isoform's TSS/TTS will be determined from supporting reads.
- **How to Treat Redundant Isoforms:**
    - No redundancy control: best TSS/TTS chosen for each unique set of splice junctions.
    - TSS/TTS that maximize length: choose this to maximize transcript length.
    - Most supported TSS/TTS: single most-supported TSS/TTS by reads.
- **How to Filter Isoforms:**
    - Filter based on support: this is the default filter.
    - Filter out subset isoforms: any isoforms that are a proper subset of another isoform are removed.
    - Both options: as the name suggests, both previous options are used.
    - Both options and remove single-exon isoforms: as above, but also removes single-exon isoforms. These isoforms are typically considered to be noise in transcriptome sequencing data and are often removed.
- **Collapse & Quantify:**
- **Minimum Mapping Quality:** Minimum mapping quality of a read assignment to an isoform.
- **Stringent Mode:** supporting reads must cover 80% of their isoform and extend at least 25 nt into the first and last exons. If those exons are themselves shorter than 25 nt, the requirement is that the read must start within 4 nt from the start or end within 4 nt from the end.
- **Check Splice Sites:** Enforces coverage of 4 out of 6 bp around each splice site and disallows insertions greater than 3 bp at the splice site.
- **Trust Ends:** Specify if reads are generated from a long-read method with minimal fragmentation.

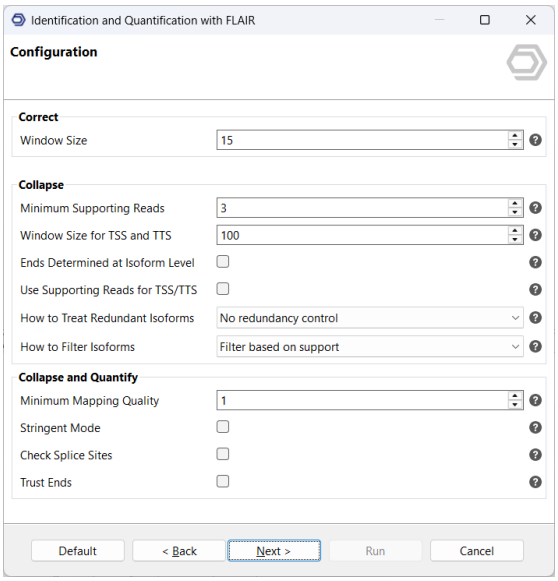


Figure 5. *"Configuration" page of the FLAIR wizard in OmicsBox.*

Output

This page defines where output files are saved.

- **Transcriptome Annotation:** transcriptome annotation in GTF format. SQANTI3 can be used to check the quality of the assembled transcriptome using this file.
- **Transcriptome Sequences:** sequences for each isoform of the transcriptome in FASTA format.
- **Counts File:** only if quantification has been performed. This file can be provided to SQANTI3 as full-length counts.

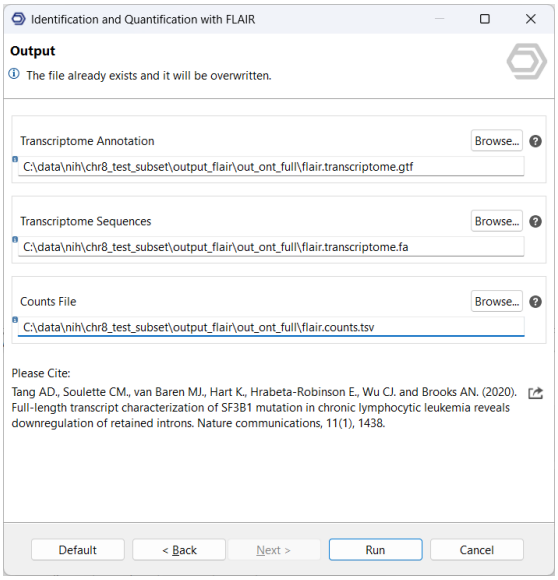


Figure 6. *"Output" page of the FLAIR wizard in OmicsBox.*

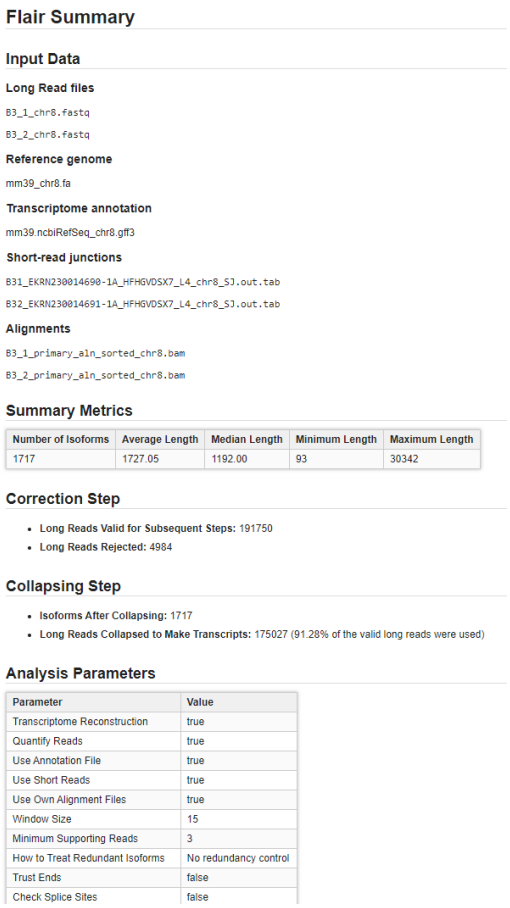
Results

FLAIR has the following outputs:

- **Files**:
- Transcriptome Annotation (GTF file). This file can be included in SQANTI3 for quality control and characterization of transcripts.
- Transcriptome Sequences (FASTA file). File with the sequences of all the defined isoforms.
- Counts File. File with the counts per isoform and per sample. This file can also be provided to SQANTI3 as full-length counts.
- **Report** with information of the input files, as well as the correction and collapsing steps.
- **Length Distribution Chart.**
- **Counts Matrix**

Report

This report first shows the input data used and then some summary metrics of the defined isoforms (number of isoforms and maximum, minimum, and average length). After this summary, it shows the number of valid and dismissed transcripts during the correction step, and then the number of isoforms created and the number of transcripts used to create them. Finally, the chosen parameters are displayed.



Flair Summary

Input Data

Long Read files

B3_1_chr8.fastq

B3_2_chr8.fastq

Reference genome

mm39_chr8.fa

Transcriptome annotation

mm39.ncbiRefSeq_chr8.gff3

Short-read junctions

B31_EKRN230014690-1A_HFHGVDSX7_L4_chr8_SJ.out.tab

B32_EKRN230014691-1A_HFHGVDSX7_L4_chr8_SJ.out.tab

Alignments

B3_1_primary_aln_sorted_chr8.bam

B3_2_primary_aln_sorted_chr8.bam

Summary Metrics

Number of Isoforms	Average Length	Median Length	Minimum Length	Maximum Length
1717	1727.05	1192.00	93	30342

Correction Step

- **Long Reads Valid for Subsequent Steps:** 191750
- **Long Reads Rejected:** 4984

Collapsing Step

- **Isoforms After Collapsing:** 1717
- **Long Reads Collapsed to Make Transcripts:** 175027 (91.28% of the valid long reads were used)

Analysis Parameters

Parameter	Value
Transcriptome Reconstruction	true
Quantify Reads	true
Use Annotation File	true
Use Short Reads	true
Use Own Alignment Files	true
Window Size	15
Minimum Supporting Reads	3
How to Treat Redundant Isoforms	No redundancy control
Trust Ends	false
Check Splice Sites	false

Figure 7. *Summary report with information on the correction and collapsing steps*

Length Distribution Chart

Histogram with the distribution of lengths of the defined isoforms in the collapsing step. This histogram may be useful for determining the acceptable range of isoform lengths and which threshold to set in SQANTI3.

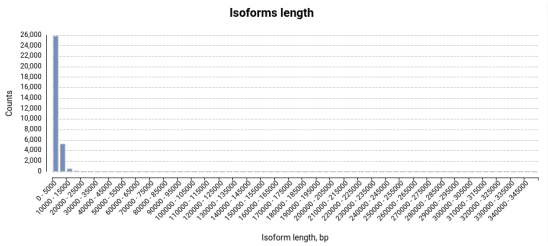
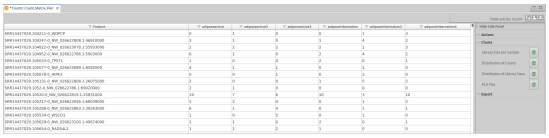


Figure 8. *Distribution of Isoform Lengths*

Count Table

If quantification was performed, a transcript count table will also appear when FLAIR finishes. In the sidebar of this table you will see a button to perform Differential Expression Analysis. The transcript names are the same as those that appear in the transcriptome generated by FLAIR.

The available tools for Differential Expression Analysis are the same as those for short reads.

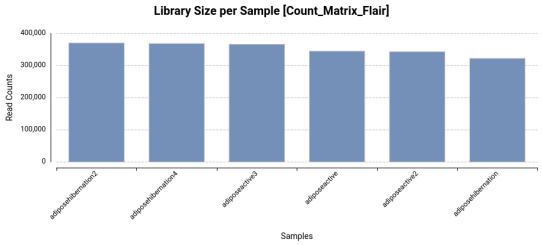


Count Table Charts

Different statistical charts can be generated from the count table. These charts provide additional information about the quantification, as well as a quality assessment of the resulting counts. They can be found in the **Side Panel → Charts** of the Count Table Viewer.

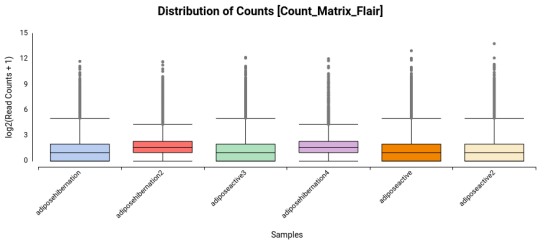
Library Size per Sample

Bar chart showing the number of read counts aligned to genomic features contained in each sample (Figure 10).



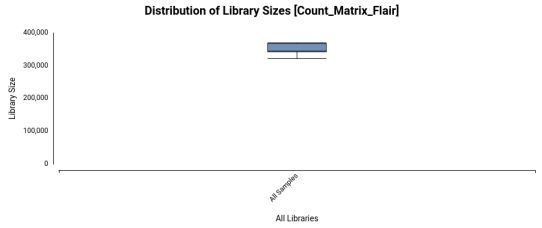
Distribution of Counts

Box plot showing how counts are distributed within each sample for all transcripts (Figure 11). Features with 0 counts in all samples will be discarded for this chart. The binary logarithm (log2) of raw counts is shown.



Distribution of Library Sizes

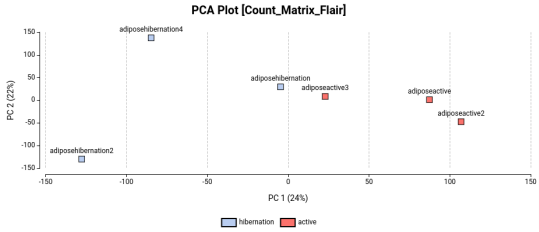
Box plot showing the distribution of library sizes across all samples being quantified (Figure 12).

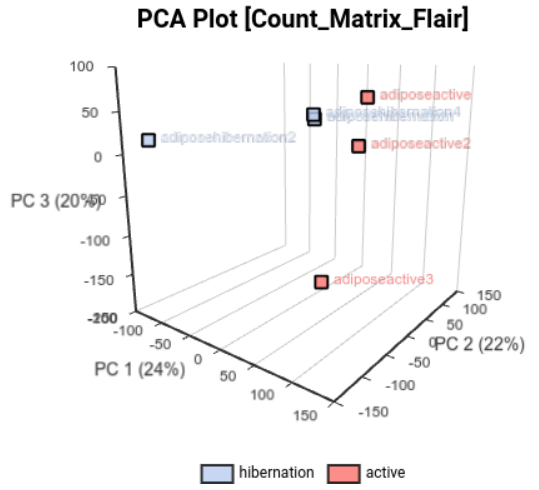


PCA Plot

This feature performs a Principal Component Analysis and generates a 2D (Figure 13) or 3D (Figure 14) plot with the first two or three principal components, respectively. This chart helps to identify which samples are similar to each other in terms of gene expression. Ideally, samples belonging to the same condition should appear closer in the plot.

The 3D PCA plot is available only for datasets with three or more samples.





IDENTIFICATION AND QUANTIFICATION WITH ISOQUANT

Introduction

With the rapid advancements made in the field of long-read sequencing, new computational tools are constantly published to aid in processing and analyzing these data. IsoQuant is one such tool, which allows long reads to be aligned to a reference genome (using Minimap2) and subsequently reconstructs and quantifies transcript models. While this can also be achieved in OmicsBox using FLAIR, IsoQuant offers not just a distinct, alternative algorithm, but also additional features. Whether FLAIR or IsoQuant should be used depends on the nature of the data set and the goal of the analysis.

IsoQuant is a computational tool for the genome-based analysis of long RNA read data originating from technologies such as PacBio or Oxford Nanopore. The tool can be run with or without a reference annotation and consists of two stages:

1. **Reference-based analysis:** If a reference annotation is provided, the provided long reads undergo reference-guided splice site correction, are assigned to the reference transcripts, and then quantified.
2. **Transcript discovery:** IsoQuant reconstructs transcript models based on the provided reads and performs abundance quantification for discovered isoforms.

Please cite IsoQuant as:

Prjibelski, A.D., Mikheenko, A., Joglekar, A. *et al.* Accurate isoform discovery with IsoQuant using long reads. *Nat Biotechnol* **41**, 915–918 (2023).

As IsoQuant utilizes a gffutils database, you may also want to consider citing the gffutils GitHub repository as:

Dale, R. (2023). gffutils v0.12. Retrieved 2024, from https://github.com/daler/gffutils.

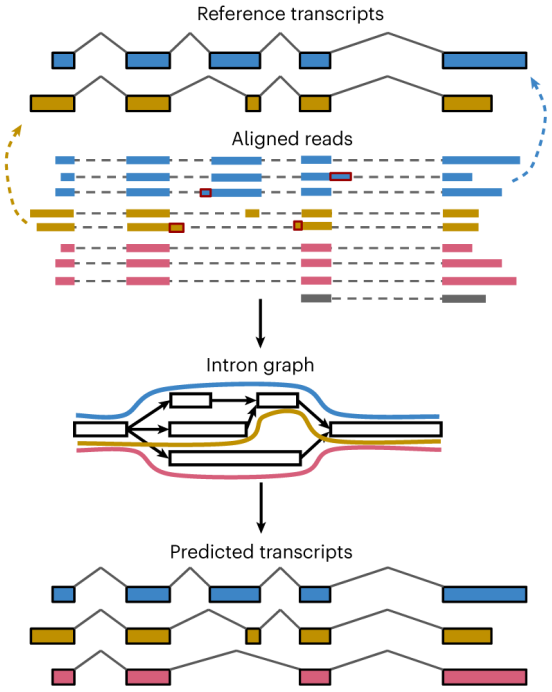


Figure 1: *Outline of the IsoQuant pipeline. When a reference gene annotation is provided, reads are assigned to annotated isoforms and alignment artifacts are corrected (top). The intron graph is constructed from read alignments (middle) and transcripts are discovered via path construction (bottom). Graphic taken from IsoQuant.*

Run IsoQuant for Long-Read Isoform Definition

IsoQuant can be found under **Transcriptomics → Long-Reads Analysis → Transcript Identification → Identification and Quantification with IsoQuant.** The wizard consists of 4 pages and facilitates the definition of the input and output options as well as the analysis parameters (Figure 2, Figure 3, Figure 4, and Figure 5)

Input

IsoQuant requires at least the following files:

- **Long-Reads Files:** Reads in FASTA or FASTQ format, or already aligned reads in BAM format. Note that, when supplying reads in FASTA or FASTQ format, they will be aligned to the given reference genome using minimap2.
- When supplying files in FASTA or FASTQ format, additionally the **Read Strandness** option can be set, which influences the aligning step. This may be important e.g. when working with Nanopore dRNA stranded reads. Otherwise, it should be left at the default value of "None".
- **Reference Genome:** FASTA file with the reference genome.

Info

- If you are supplying BAM files as inputs, ensure that you are supplying the reference genome in the same version as you used to align your reads.
- Note that, for quantification, every file you supply will be treated as an individual sample. If you have multiple files belonging to the same sample, consider concatenating them into a single file first.

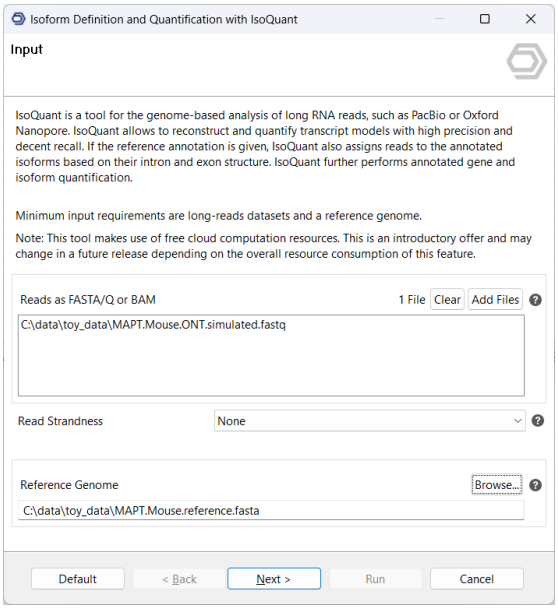


Figure 2: *Input page of the OmicsBox IsoQuant wizard.*

Annotation Inputs

Additionally, a reference annotation file may also be provided:

- **Transcriptome Annotation:** GTF file with annotations of the reference transcriptome.
- When using a transcriptome annotation which already includes "gene" and "transcript" level features, you may also check the checkbox **Detailed Gene Database**. This saves some time during the conversion process, as transcript and gene entries will not have to be inferred. If you are unsure, leave this option off.

Info

- If you are providing a reference annotation, ensure that its version matches that of the provided reference genome.
- While the use of a reference annotation is not mandatory, it is likely to increase precision and recall. If you have access to a fitting reference annotation, it is recommended to use it.

Optionally, short reads may be provided:

- **Short-Read BAM Files:** One or multiple .bam format short-read alignment files which are used to correct the splice junction alignments of the provided long reads. Note that the short reads are NOT used for transcript discovery or quantification.

Info

If you are providing short reads, ensure that they are aligned to the same version of the reference genome as you are supplying.

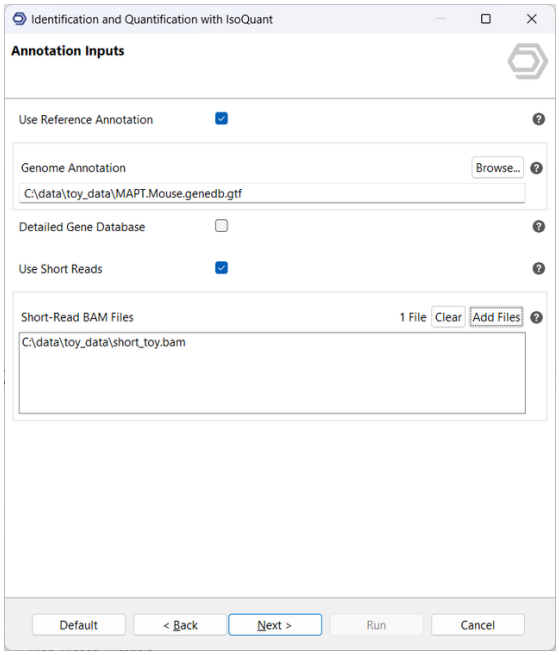


Figure 3: *Annotation Input page of the OmicsBox IsoQuant wizard.*

Algorithm Options

This page provides some more detailed options to configure the algorithm:

- **Transcript and Gene Quantification:** What quantification strategy should be used to assess abundance on both transcript- and gene-level.
  - **Unique Only:** Only count reads that are uniquely assigned and consistent with a transcript (default for transcript-level).
  - **With Ambiguous:** Ambiguously assigned reads are split with equal weights.
  - **Unique Splicing Inconsistent:** Uniquely assigned reads which do not contradict annotated splice sites are included (default for gene-level).
  - **Unique Inconsistent:** Uniquely assigned reads are included, allowing any kind of inconsistency.
  - **All:** Both ambiguous and inconsistent reads are included.
- **Data Type:** Most importantly, the **Data Type** has to be specified:
  - PacBio CCS or FLNC,
  - ONT dRNA or cDNA,
  - assembled / corrected transcript sequences.
- **Full-Length Transcripts:** Whether both ends of the sequences can be considered reliable (e.g. in PacBio FLNC reads).
- **Matching Strategy:** How exact or loose the read-to-isoform-matching algorithm should be.
  - **Exact:** All minor errors are treated as inconsistencies.
  - **Precise:** Only minor alignment errors are allowed. (default for PacBio)
  - **Flexible:** Alignment errors typical for Nanopore are allowed, short novel introns are treated as deletions. (default for ONT)
  - **Loose:** Even more serious inconsistencies are ignored, ambiguity is resolved based on nucleotide similarity.
- **Splice Correction:** Which splice correction strategy should be employed.
  - **None:** No correction is applied.
  - **Default PacBio:** Optimal settings for PacBio CCS reads. (default for PacBio)
  - **Default ONT:** Optimal settings for ONT reads. (default for ONT)
  - **Conservative ONT:** Conservative settings for ONT reads, only incorrect splice junctions and skipped exons are fixed.
  - **Assembly:** Optimal settings for a transcriptome assembly. (default for Transcript Assembly)
  - **All:** Correct all discovered minor inconsistencies, may result in overcorrection.
- **Model Construction:** Which model construction strategy should be employed.
  - **Reliable:** Only the most abundant and reliable transcripts are reported; precise, but not sensitive.
  - **Default PacBio:** Optimal settings for PacBio CCS reads. (default for PacBio)
  - **Sensitive PacBio:** Sensitive settings for PacBio CCS reads, more transcripts are reported possibly at a cost of precision.
  - **Full-Length PacBio:** Optimal settings for full-length PacBio CCS reads.
  - **Default ONT:** Optimal settings for ONT reads. (default for ONT)
  - **Sensitive ONT:** Sensitive settings for ONT reads, more transcripts are reported possibly at a cost of precision.
  - **Assembly:** Optimal settings for a transcriptome assembly: input sequences are considered to be reliable and each transcript to be represented only once, so abundance is not considered. (default for Transcript Assembly)
  - **All:** Reports almost all novel transcripts, loses precision in favor to recall.
- **Report Mono-Exonic Transcripts:** Whether to report novel mono-exonic transcripts. (default to off for ONT, on for PacBio and Transcript Assembly)

Info

When selecting the **Data Type**, the subsequent options will automatically be set to appropriate default settings. However, you may still adjust them for your specific purposes.

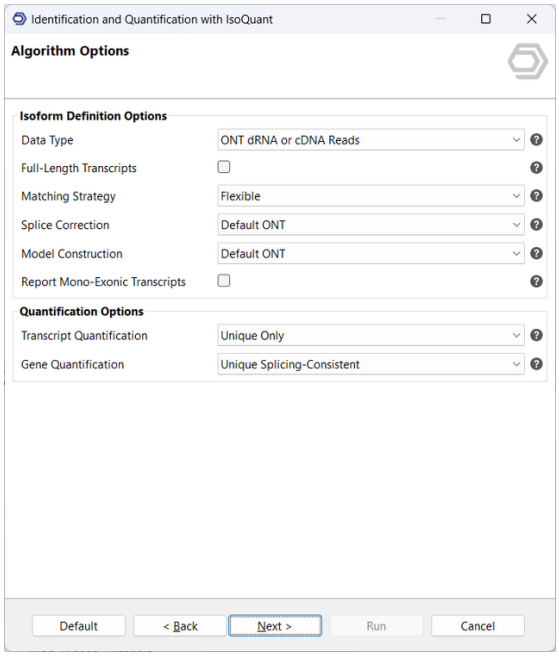


Figure 4: *Algorithm Options page of the OmicsBox IsoQuant wizard.*

Output

- **Output File Prefix:** Set a name which will serve as a prefix for all output files.
- **Save Transcript Model Annotations**: Whether you want to save the transcript models identified by IsoQuant as a .gtf file. If so, select a location for the .gtf file below.
- **Count Table Outputs:** As IsoQuant also performs quantification, you can choose several options on which quantification outputs to receive:
  - Reference Gene Counts: Quantification at gene-level, grouped by input files. This option will be unavailable if you have not provided a reference annotation.
  - Reference Transcript Counts: Quantification at transcript-level, using only the transcript models present in the supplied reference annotation, grouped by input files. This option will be unavailable if you have not provided a reference annotation.
  - Transcript Counts: Quantification at transcript-level, using the transcript models identified by IsoQuant, grouped by input files.

Tip

Note that if only Reference Gene and/or Transcript Counts are selected as desired outputs, IsoQuant runs with the "*--no_model_construction*" flag. This means that only quantification based on the provided reference is performed, and the model construction step is skipped. This saves considerable computational resources, which results in much faster runtime.

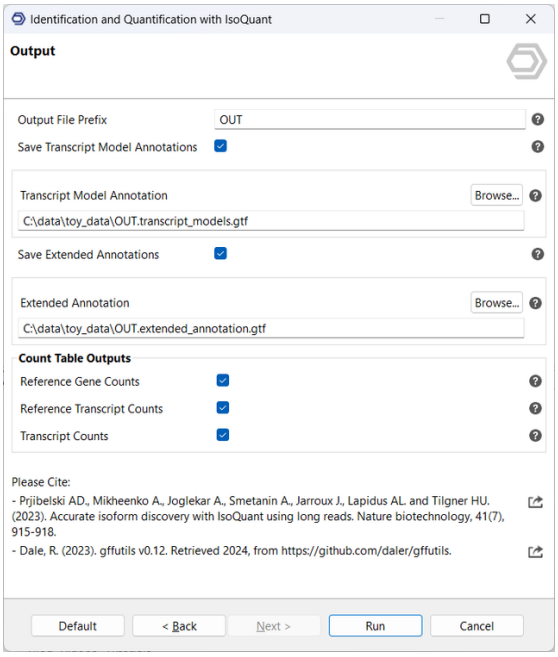


Figure 5: *Output page of the OmicsBox IsoQuant wizard.*

Results

IsoQuant has the following outputs:

- **Transcript Models as Annotation (GTF file):** An annotation of transcript models for which IsoQuant finds sufficient evidence in the given data. This can be used to run a SQANTI3 quality control and filtering analysis.
- **Extended Annoation (GTF file):** When supplying a reference annotation to IsoQuant, an extended annotation can be obtained which contains all reference transcripts (even those not observed in the given data), extended by any novel transcripts discovered by IsoQuant.
- **Report** with information on the assignment and alignment of reads, as well as the categories of identified transcript models.
- **Count Tables** as specified in the output page. These can be used to run differential expression analyses.
- **Length Distribution Chart** which shows the distribution of isoform lengths.

Report

The summary report of an IsoQuant run in OmicsBox first provides a description of all provided input and reference files, as well as the chosen algorithm configuration. The report further gives an overview over some statistics concerning the assignment and alignment of reads, as well as the structural classification of Transcript Models as “Known”, “Novel In Catalog”, or “Novel Not In Catalog”. An example of this section can be seen in Figure 6.

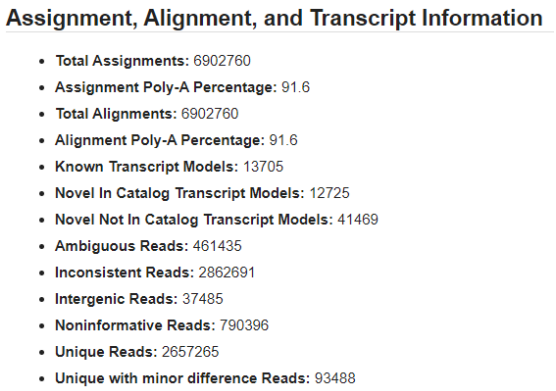


Figure 6: *Example of the statistics section of an IsoQuant report in OmicsBox.*

Transcript Models

The transcript models identified by IsoQuant are provided as a .gtf file. When running IsoQuant with a reference annotation, an extended annotation can also be obtained.

Count Tables

Depending on the selection in the outputs, the different count tables will be provided as OmicsBox objects. These can be used in downstream differential expression analyses through the use of the Sidebar action “Differential Expression Analysis.”

Length Distribution Chart

This histogram shows the distribution of Isoform lengths. This information can be useful in order to judge the acceptable range of isoform lengths, as well as to set the length threshold when running SQANTI3.

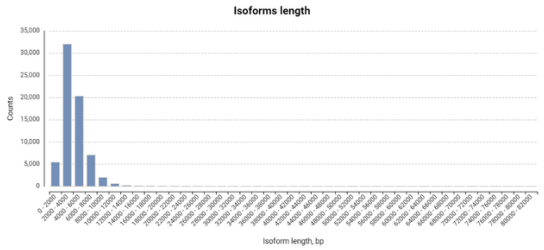


Figure 7: *Example chart of the distribution of transcript isoform lengths. OmicsBox provides a rich interface of options to customize the appearance of this plot, including setting the desired bin size.*

REFERENCE-FREE ISOFORM RECONSTRUCTION

Introduction

With the advancement of transcriptomics through long-read technologies, various reference-guided tools for reconstructing a transcriptome from long reads have become publicly available. As their name suggests, these tools utilize a reference genome and its annotation. However, for non-model organisms, it is common to encounter situations where there is a need for conducting a differential expression analysis without access to a characterized genome or transcriptome. To address this challenge, we have incorporated the isONpipeline (isONclust3, isONcorrect and isONform) into OmicsBox. This pipeline enables the reconstruction of a transcriptome using long reads from Pacific Biosciences (PacBio) or Oxford Nanopore Technologies (ONT).

Please cite the components of the isONpipeline as follows:

- Alexander J. Petri. and Kristoffer Sahlin. (2025). De novo clustering of large long-read transcriptome datasets with isONclust3. bioRxiv. https://doi.org/10.1101/2024.10.29.620862
- Sahlin K. and Medvedev P. (2021). Error correction enables use of Oxford Nanopore technology for reference-free transcriptome analysis. Nature communications, 12(1), 2. https://doi.org/10.1038/s41467-020-20340-8
- Petri AJ. and Sahlin K. (2023). isONform: reference-free transcriptome reconstruction from Oxford Nanopore data. Bioinformatics (Oxford, England), 39(39 Suppl 1), i222-i231. https://doi.org/10.1093/bioinformatics/btad264

Reference-free Isoform Reconstruction

Reference-Free Isoform Reconstruction can be found in the Transcriptomics Module of OmicsBox under **Transcriptomics → Long-Reads Analysis → Transcript Identification → Reference-free Isoform Reconstruction.** The wizard consists of 3 pages and allows to define the input and output options as well as the analysis parameters (Figure 1, Figure 2, Figure 3).

Input Page

In this page you will be able to select the files that contain a transcriptome and some parameters regarding long reads length-filtering.

- **Long-Read Files:** FASTQ files with long reads that come from PacBio or ONT sequencing technologies.

If you select multiple files containing reads, all the reads from those files will be combined into a single file and then this tool will run. If you have multiple transcriptomes sequenced in different files, please ensure to run this tool on each individual file separately.

- **Minimum Read Length:** reads shorter than this value will be filtered out.
- **Maximum Read Length:** reads longer than this value will be filtered out.

While filtering outliers can improve the runtime of the algorithm, it is also vital not to exclude too many reads in order to reconstruct the complete sequenced transcriptome.

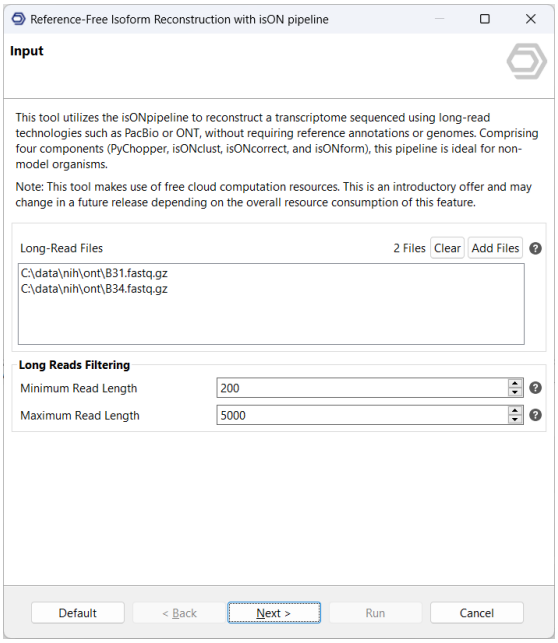


Figure 1: *Input page of the OmicsBox isONpipeline wizard.*

Configuration

- **General Parameters:** these parameters do not belong to a specific step.
  - **Reconstruction Pipeline:**
    - The previously available option to run the isONpipeline with PyChopper integrated has been removed as we work to restructure and improve the implementation of the isONpipeline in OmicsBox. If you need help to run PyChopper to pre-process your ONT data, please contact support for assistance.
    - ONT Pipeline: for pre-processed ONT reads, the full isONpipeline consisting of isONclust3, isONcorrect, and isONform is executed.
    - PacBio Pipeline: for PacBio reads, only isONclust3 and isONform are executed.
  - **Isoform Read Support:** number of reads to call a transcript in isONform.
- **Clustering** (isONclust3)**:**
  - **K-mer Size:** length of the k-mers to make a Hash table before clustering.
  - **Window Size:** length of the sliding window to obtain the k-mers used to create the Hash table.

> **Info**
>
> For ONT data, k-mer size and window size seem to be optimal at 13 and 21, respectively, and at 15 and 51, respectively, for PacBio data.

- **Correction** (isONcorrect)**:**
  - **K-mer Size:** length of the k-mers to make a Hash table before the correction step.
  - **Window Size:** length of the sliding window to obtain the k-mers.
- **Reconstruction** (isONform)**:**
  - **K-mer Size:** length of the k-mers to make a Hash table before clustering.
  - **Window Size:** length of the sliding window to obtain the k-mers used to create the Hash table.
  - **Maximum Difference in 3':** maximum length difference at 3' end, for which subisoforms are still merged into longer isoforms.
  - **Maximum Difference in 5':** same at 5' end.

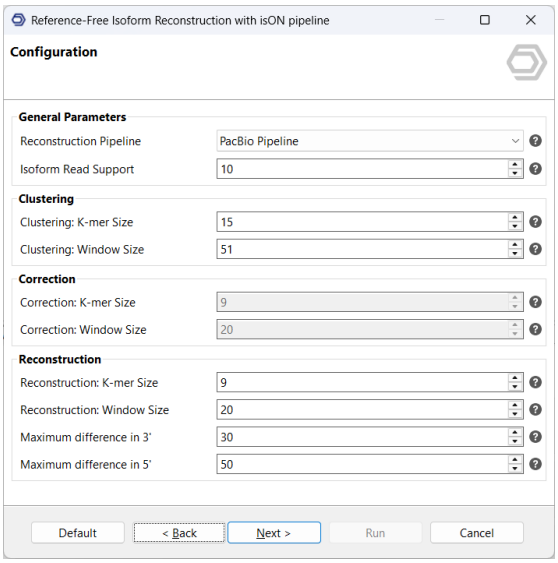


Figure 2: *Configuration page of the OmicsBox isONpipeline wizard.*

Output

Transcriptome FASTA File: location to save the final transcriptome FASTA file.

Quantification File: location to save the transcript count table (.csv file). Quantification is grouped by input files.

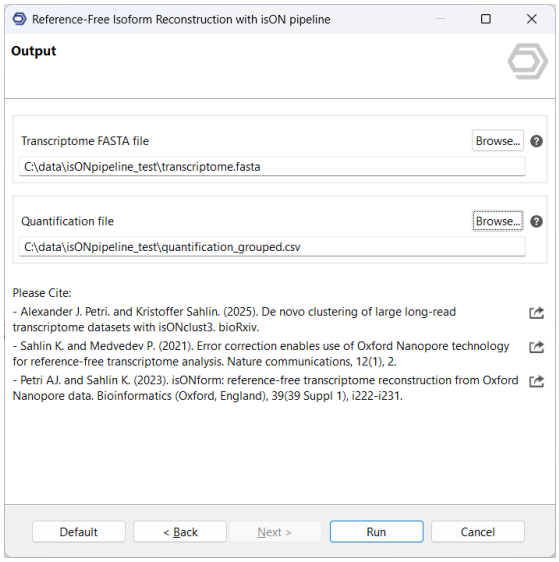


Figure 3: *Output page of the OmicsBox isONpipeline wizard.*

Results

The main output is the reconstructed **transcriptome in FASTA format** as well as the **quantification count table**.

A **summary report** with the input FASTA filenames, information about the reconstructed transcriptome (number of isoforms and maximum, minimum and average length), the parameters set and the references will be generated too (see Figure 4). In addition, two charts will be output:

- **Isoform Length Distribution:** in this chart you can see the distribution of length of the transcriptome (see Figure 5).
- **Isoform Support Distibution:** distribution of the number of reads used to reconstruct a transcript (see Figure 6).

Transcript Reconstruction Report

Input Data

/home/enrique/research/isontformvsflair/prueba_multi_file/subset1.fastq.gz

/home/enrique/research/isontformvsflair/prueba_multi_file/subset2.fastq.gz

Summary Metrics

Number of Isoforms	Average Length	Minimum Length	Maximum Length
55	2118.93	1048	3822

Analysis Parameters

Parameter	Value
Maximum Read Length	4000
Minimum Read Length	1000
Reconstruction Pipeline	PacBio Pipeline
Isoform Read Support	10
K-mer Size	15
Window Size	50

Figure 4: *Example Summary Report of an isONpipeline run in OmicsBox.*

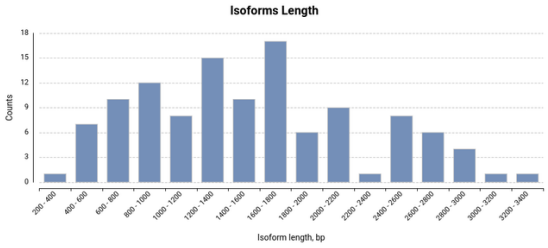


Figure 5: *Customizable chart of the length distribution of discovered transcripts.*

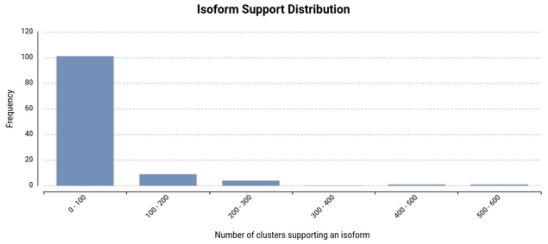


Figure 6: *Customizable chart of the support distribution of discovered transcripts.*

Curation of Long-Read Transcriptomes with SQANTI3

INTRODUCTION

SQANTI3 is a bioinformatics tool designed for the quality control and filtering of full-length transcripts sequenced with PacBio's long-read technology. It is designed as the next step of the IsoSeq pipeline. The interest in this tool comes from the usefulness of long-read transcriptome sequencing to describe eukaryotic transcriptomes and replace the use of second-generation sequencing. Illumina short-reads cannot contain a whole transcript and are not able to well-characterize eukaryotic transcriptomes.

This tool can reveal the nature and novelty found by long-read sequencing by classifying transcripts based on the comparison between their splice junctions and the reference transcriptome provided. It combines the long read-defined transcripts (in a FASTA/Q or GTF format) with the reference annotation and with other optional data to provide a wide range of descriptors of transcript quality. SQANTI3 generates a comprehensive report to facilitate quality control and filtering of the isoform models and performs two different tasks, both of them equally important:

1. **Isoform classification and quality control for long read-defined transcriptomes:** the categories in which transcripts are classified, together with a long list of attributes and descriptors, allow to carefully inspect the properties of their transcriptome and identify potential problems generated data processing.
2. **Filter for long read-defined transcriptomes:** with the supplied information, users can set different parameters to remove potential false positive isoforms.

In SQANTI3 version 5, another step was added. In this step, SQANTI3 rescues isoforms that have been removed from the curated transcriptome but that have evidence in the reference transcriptome. The idea is to avoid losing transcripts and/or genes that are detected as expressed by long read sequencing but could not be confidently validated using orthogonal data.

Please cite SQANTI3 as:

Pardo-Palacios, F. J., Arzalluz-Luque, A., Kondratova, L., Salguero, P., Mestre-Tomás, J., Amorín, R., ... & Conesa, A. (2024). SQANTI3: curation of long-read transcriptomes for accurate identification of known and novel isoforms. *Nature Methods*, 1-5.

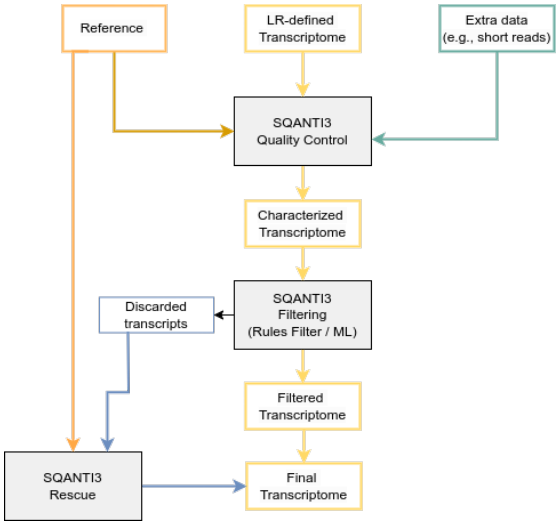


Figure 1. *SQANTI3 workflow*

RUN SQANTI3 FOR TRANSCRIPTOME CHARACTERIZATION

SQANTI3 can be found in the Transcriptomics Module of OmicsBox under **Transcriptomics → Long-Reads Analysis → Transcript Identification → Curation of Long-Read Transcriptomes with SQANTI3.**The wizard consists of 5 pages and allows to define the input and output options as well as the analysis parameters (Figure 2, Figure 3, Figure 4, Figure 5 and Figure 6).

Input 1

- First of all, SQANTI3 can be fed with long-read information in two different ways:
- **Transcriptome Annotation File:** annotation file created in your transcriptome reconstruction pipeline. This option is recommended in order to analyze non-redundant isoforms precisely. If you do not have a long-read annotation file, please use the other option.
- **Transcriptome FASTA/Q File:** FASTA/Q File with Transcripts. If long-read data is introduced this way, transcripts will be mapped using minimap2. The BAM file will be transformed in a GTF file respecting splice sites.
- **Raw Long Reads:** FASTA/Q File with Raw Long Reads both from PacBio or Nanopore Technology. If long-read data is introduced this way, long reads in this file (FASTA or FASTQ format) will be mapped using minimap2, and mapped reads in the SAM file will be collapsed using cDNA Cupcake to generate non-redundant isoforms.

It is recommended that users generate their own transcriptome files (in GTF or FASTX format) so as to know exactly which isoforms are being analysed.

- **Reference Genome:** this tool also needs the reference genome in FASTA format. Please check that the chromosome/scaffolds names are the same in the reference annotation and the reference genome.
- **Annotation of the Reference Genome:** it must be introduced in GTF format. This file will be taken as reference to describe the degree of novelty of each transcript. Make sure your annotation file is based on the correct reference genome version.

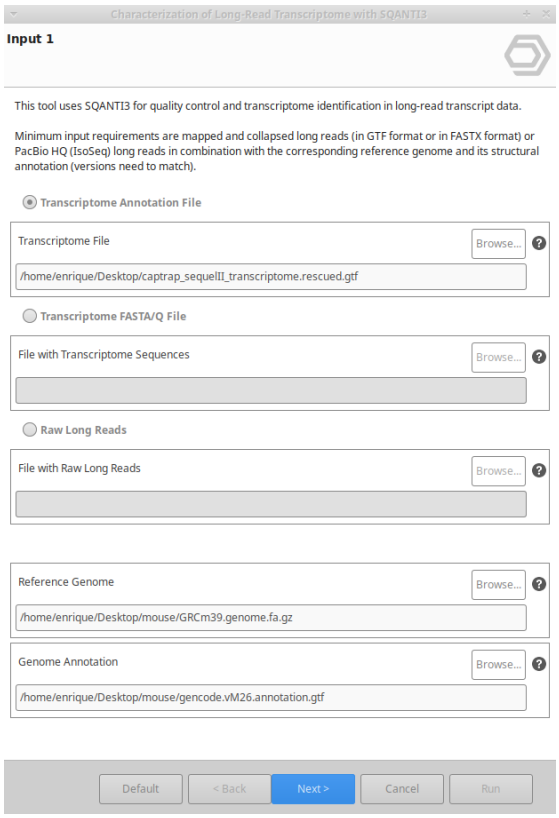


Figure 2. *Input 1 Page*

Input 2

This tool can also accept short reads (paired-end or single-end) to validate isoforms taking into account the coverage in splice junctions. The expression of each isoform can also be estimated with paired-end short-reads (not with single-end because it is very time-consuming).

If paired-end date is introduced, the upstream and downstream files patterns must be typed in order to differentiate each kind of file.

It is strongly recommended to use short-read data, even if it comes from a different experiment, to validate isoforms using splice junctions coverage.

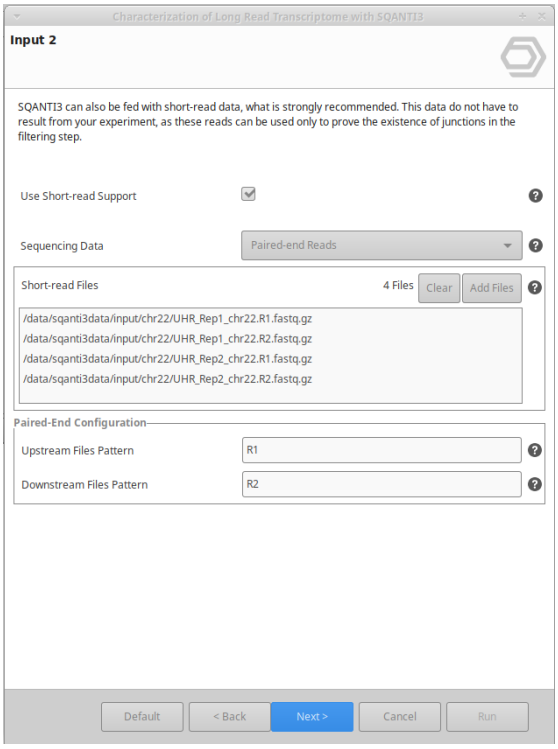


Figure 3. *Input 2 Page*

Input 3

In this page, extra files can be introduced in order to have additional details in the output.

- **Transcription Start Site (TSS) Annotation File:** bed file with information of the TSS in a genome. This kind of information can only be used for human and mouse yet. To have more information please visit the referenceTSS webpage.
- **File with PolyA Motifs:** text file with one polyA motif for that species in each line. In the PolyASite webpage different files for *H. sapiens, C. elegans* and *M. musculus* can be found with PolyA information, and the last column can be parsed to get the polyA motifs.
- **File with PolyA Peaks:** complementary to polyA motif information, polyA site data can be supplied. For human, mouse, and worm, you can download public polyA site data from the PolyASite atlas.
- **File With FL Counts:** text or TSV file with Full Length Counts. This information can be obtained, for example, using a cDNA cupake script (get_abundance_post_collapse.py).

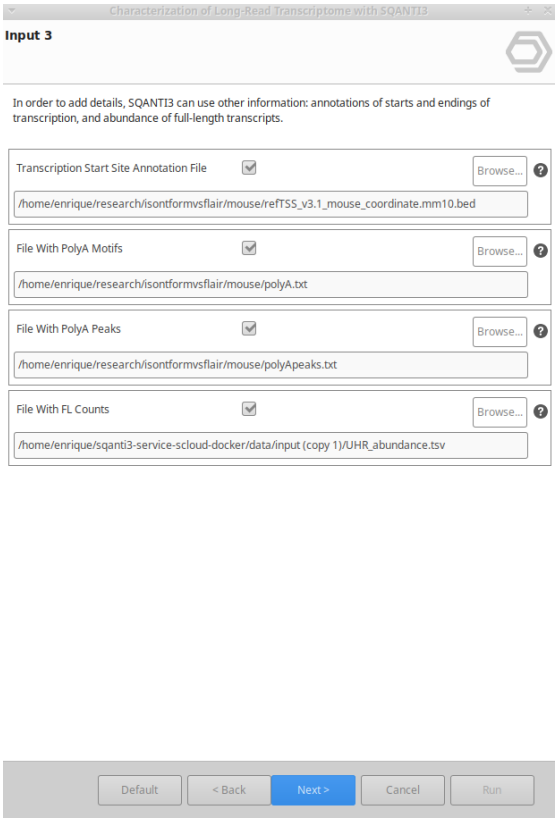


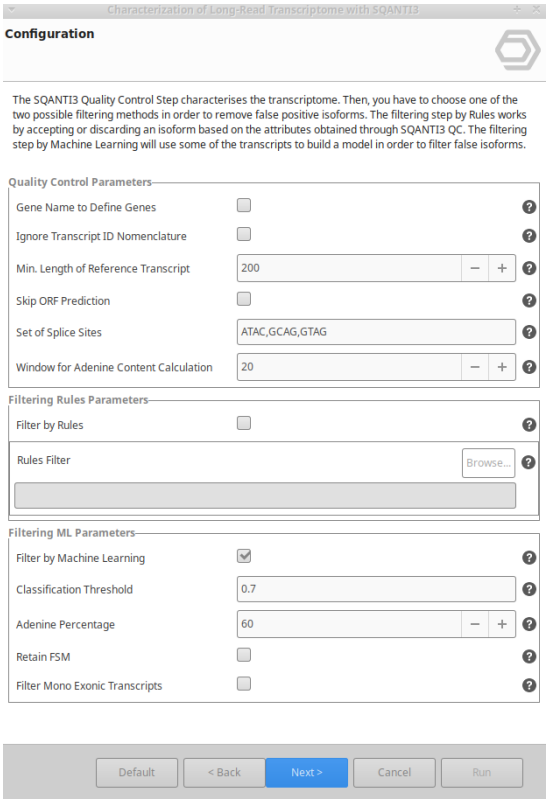
Figure 4. *Input 3 Page*

Configuration

In this page the parameters for SQANTI3 Quality Control and SQANTI3 Filtering Stepcan be set:

- **Quality control:**
- **Ignore Transcript ID Nomenclature:** allow the usage of transcript IDs non related with PacBio's nomenclature (PB.X.Y).
- **Min. length of Reference Transcript:** minimum reference transcript length.
- **Skip ORF Prediction:** check it to skip ORF prediction so as to save time. If ORF prediction is checked, the translated transcriptome could not be returned.
- **Set of Splice Sites:** set of splice sites to be considered as canonical. If the set is going to be changed, type the new splice sites separated with comma and no spaces.
- **Filtering Rules Parameters:**
- **Filter by Rules:** check this option to filter isoforms that do not comply some rules.
- **Rules Filter:** select the rules to filter out isoforms.
- **Filtering ML Parameters:**
- **Filter by Machine Learning:** select this option to filter isoforms with a Random Forestclassifier that labels possible transcripts as true isoforms or artifacts using SQANTI QC descriptors as predictive variables.
- **Classification Threshold:** Machine Learning probability threshold to classify transcripts as positive isoforms. Probability has to be higher than this value to be classified as isoform. Select a higher value to be more stringent on which possible transcripts are considered actual isoforms.
- **Adenine Percentage:** adenine percentage at genomic 3' end to flag an isoform as intra-priming. Intra-priming is the process of generation transcript artefacts due to cDNA poly-dT priming off genomic A stretches that are not true polyA tails.
- **Retain FSM:** when checked, forces retaining FSM transcripts regardless of ML filter result (FSM are threfore automatically classified as isoforms).
- **Filter Mono Exonic Transcripts:** remove transcripts made of just one exon. Normally, this transcripts are removed as they might be truncated isoforms.

Quality control and filtering steps are mandatory. In the case of the filtering step, you have to select one of the two methods: rules filter or ML filter.

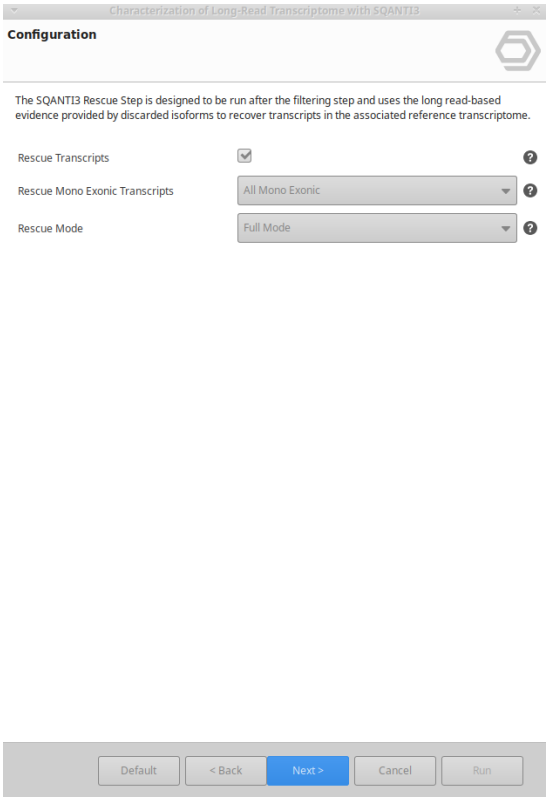


Picture 5. Configuration Page

Configuration

In this page the parameters for the Rescue step can be set:

- **Rescue Transcripts:** the SQANTI3 Rescue Step is designed to be run after the filtering step and uses the long read-based evidence provided by discarded isoforms to recover transcripts in the associated reference transcriptome. The idea behind this strategy is to avoid losing transcripts/genes that are detected as expressed by long read sequencing, but whose start/end/junctions could not be confidently validated using orthogonal data, resulting in the removal of those genes/transcripts from the transriptome. As a result, SQANTI3 rescue will generate an **expanded transcriptome GTF** including a set of reference transcripts as well as the long read-defined isoforms that passed the filter.
- **Rescue Mono Exonic Transcripts:** you can choose whether to enable or disable the rescue of transcripts with only one exon, regardless of their category, or limit the rescue to those in the FSM category.
- **Rescue Mode:**
- **Automatic Mode:** all reference transcripts that were represented by at least one FSM or ISM in the original post-QC transcriptome are retrieved. Then, those reference transcripts for which all FSM representatives were removed by the filter are rescued.
- **Full Mode:** also NIC and NNC categories will be used to rescue transcripts.Since there is no associated transcript information, all transcripts classified as artifacts will be included in the rescue candidate list. As a result, we consider all reference or long read-defined transcripts from genes that have at least one rescue candidate to be rescue targets.



Picture 6. Configuration 2 Page

Output

- **Set Prefix for Output Files:** prefix to add to the output filenames.
- **Information Files:** directory to save files with all information of isoforms and junctions, and the transcriptome in GTF format.
- **Isoform FASTA file:** directory to save isoform sequences in FASTA format.

- **Translated transcriptome FASTA file:** directory to save translated isoform sequences in FASTA format.This is only possible if ORF prediction has not been skipped.

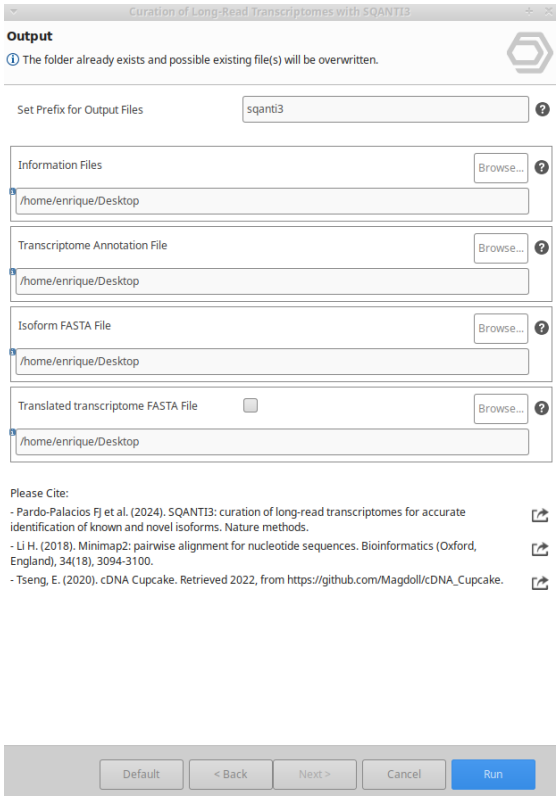


Figure 7. *Output Page*

RESULTS

SQANTI3 has the following outputs:

- **Table** with the main information of the isoforms (Figure 8).
- **Report** with information of the number of genes, isoforms per category and types of junctions before and after filtering isoforms (if not filtering is applied, only information of the first step will be shown) (Figure 9).
- **Charts**:
- General information charts.
- Distance to TSS and TTS.
- Filtering.
- Short and Long Read Coverage.
- Redundancy Analysis.
- Quality Features.
- Rescue Step
- **Files**:
- Transcriptome Annotation File.
- Transcriptome File.
- Translated Transcriptome File (optional).
- Junctions File.
- Classification File.

Table with information of the isoforms

The next columns can be seen (Figure 8):

- **Category:** structural category of the transcript.
- **Subcategory:** deeper classification than the general category for each transcript.
- **Isoform:** ID of the isoform that comes from the collapsing algorithm.
- **Chr:** name of the chromosome/scaffold where the encoding gene is located.
- **Length:** isoform length.
- **Exons:** number of exons of the isoform.
- **Gene ID:** gene ID that appears in the annotation file, if annotated.
- **Transcript ID:** transcript ID that appears in the annotation file, if annotated.
- **Gene exons:** number of exons of the gene if that gene is annotated. If it is novel, this information is not shown.
- **Difference to TSS:** distance of query isoform 5' start to reference transcript start end. Negative value means query starts downstream of reference.
- **Difference to TTS:** distance of query isoform 3' end to reference annotated end site. Negative value means query ends upstream of reference.
- **RT switching:** TRUE if one of the junctions could be a RT switching artifact.
- **All Canonical Junctions:** whether all junctions are canonical or not.
- **Min. coverage:** minimum junction coverage based on STAR algorithm. If no short reads are given, this column is not shown.
- **FL counts:** FL count associated with this isoform if a FL counts file is provided. Otherwise this column is not shown.
- **Isoforms expression:** short read expression for this isoform if paired-end reads are provided, otherwise this column is not shown.
- **Gene expression:** short read expression for the gene associated with this isoform (summing over all isoforms) if paired-end reads are provided, otherwise this column is not shown.
- **Coding Type:** whether the isoform encodes a protein or not.
- **Predicted NMD:** TRUE if there's a predicted ORF and CDS ends at least 50 bp before the last junction; FALSE if otherwise. NA if non-coding.
- **CAGE:** TRUE if the PacBio transcript start site is within a CAGE Peak. If a TSS annotation file is not given, this column is not shown.
- **PolyA:** shows the location of the last base of the hexamer. Position 0 is the putative polyA site. This distance is hence always negative because it is upstream.
- **PolyA motif:** if a polyA motif list is given, shows the top ranking polyA motif found within 50 bp upstream of end. Otherwise, this column is not shown.
- **TSS ratio:** the short-read mean coverage of the 100bp upstream and downstream a reported TSS are measured. Then the ratio coverage inside isoform + 0.01/ coverage outside isoform + 0.01 is calculated.

Report

In the report, there are two main parts: in the first one, some Job Information can be seen (Figure 9). There are four tables:

- **Table at the gene level:** information of the number of annotated genes and novel genes before and after filtering.
- **Table at the isoform level:** information of the number of isoforms in the different structural categories before and after filtering and description of each structural category.
- **Table at the splice junction level:** information of the number of each type of splice junction before and after filtering and description of each type of splice junction.
- **Filtering Information:** if rules filter is applied, a table with the rules applied will be shown. If the ML filter is used, different ML metrics are displayed.
- **Rescue Information:** if the rescue step has been applied, two additional tables will be shown in the report. The first one will show the category and the number of isoforms rescued. The second one will show the reasons why other transcripts were excluded from being rescued:
- Reference Already Present: if the mapping hit is a reference transcript that is already represented by an isoform, be it an FSM that passed the filter or a transcript that was obtained during automatic rescue.
- ML Probability: if the mapping hit did not pass the supplied ML probability threshold or the rules in the JSON file.
- Long Read Transcript: if the mapping hit is a long read-defined isoform, and is therefore already present in the transcriptome.

Intergenic and Genic Intron transcripts are the structural categories that belong to novel genes. These categories, along with Genic Genomic and Antisense, are not common, so they should not be frequent in the report.

In the second part, the main parameters used in SQANTI3 job are shown (Figure 10).

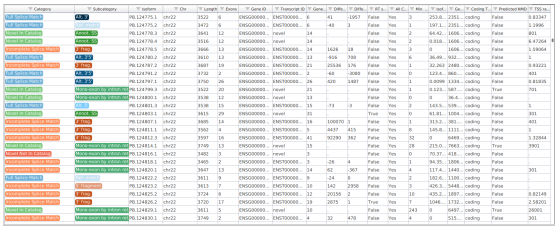
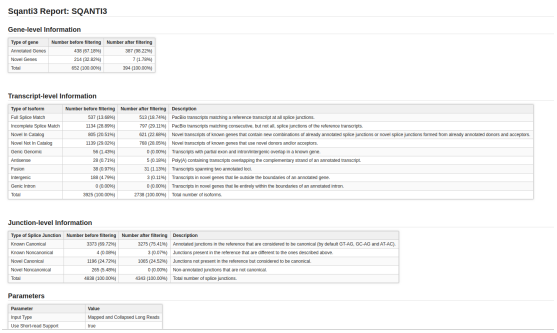


Figure 8. *Classification table of transcripts in OmicsBox*



Sqanti3 Report: SQANTI3

Gene-level Information

Type of gene	Number before filtering	Number after filtering
Annotated Genes	439 (67.18%)	387 (98.22%)
Novel Genes	214 (32.82%)	7 (1.78%)
Total	652 (100.00%)	394 (100.00%)

Transcript-level Information

Type of isoform	Number before filtering	Number after filtering	Description
Full Splice Match	537 (13.68%)	513 (16.74%)	PacBio transcripts matching a reference transcript at all splice junctions.
Incomplete Splice Match	1134 (28.89%)	707 (23.11%)	PacBio transcripts matching consecutive, but not all, splice junctions of the reference transcripts.
Novel In Catalog	665 (25.51%)	621 (22.69%)	Novel transcripts of known genes that contain new combinations of already annotated splice junctions or novel splice junctions formed from already annotated donors and acceptors.
Novel Not In Catalog	1139 (29.02%)	789 (28.05%)	Novel transcripts of known genes that use novel donors and/or acceptors.
Genic Genomic	56 (1.43%)	0 (0.00%)	Transcripts with partial exon and intron/intergenic overlap in a known gene.
Antisense	28 (0.71%)	5 (0.18%)	PolyA's containing transcripts overlapping the complementary strand of an annotated transcript.
Fusion	38 (0.97%)	31 (1.13%)	Transcripts spanning two annotated loci.
Intergenic	188 (4.79%)	3 (0.11%)	Transcripts in novel genes that lie outside the boundaries of an annotated gene.
Genic Intron	0 (0.00%)	0 (0.00%)	Transcripts in novel genes that lie entirely within the boundaries of an annotated intron.
Total	3925 (100.00%)	2738 (100.00%)	Total number of isoforms.

Junction-level Information

Type of Splice Junction	Number before filtering	Number after filtering	Description
Known Canonical	3373 (69.72%)	3275 (75.41%)	Annotated junctions in the reference that are considered to be canonical (by default GT-AG, GC-AG and AT-AC).
Known Noncanonical	4 (0.08%)	3 (0.07%)	Junctions present in the reference that are different to the ones described above.
Novel Canonical	1186 (24.72%)	1065 (24.53%)	Junctions not present in the reference but considered to be canonical.
Novel Noncanonical	265 (5.48%)	0 (0.00%)	Non-annotated junctions that are not canonical.
Total	4828 (100.00%)	4343 (100.00%)	Total number of splice junctions.

Parameters

Parameter	Value
Input Type	Mapped and Collapsed Long Reads
Use Short-read Support	true

Figure 9. *Report with information of genes, isoforms and splice junctions*

Parameters

Parameter	Value
Input Type	Mapped and Collapsed Long Reads
Short Reads Input	Use Supporter Short Reads
Sequencing Data	Paired-end Reads
TSS File	true
PolyA-motif File	true
File with Full Length Counts	true
Ignore Transcript ID Nomenclature	false
Min. Length of Reference Transcript	200
Skip ORF Prediction	false
Set of Splice Sites	ATAC,GCAG,GTAG
Saturation Curves	false
Filtering	true
FASTA File of Translated ORFs	true
Adenine Percentage	0.6
Adenines in a Row	6
Distance to Annotated TTS	50
Minimum Short-Read Coverage	3
Filter Mono Exonic Transcripts	false
Set Prefix for Output Files	example

Figure 10. *Parameters information in the report*

Charts

In the side panel of the table (Figure 8) there are different action buttons with different chart categories:

- **General Information Charts:** these charts are related to general characteristics.
- Isoforms per Gene: frequency of the number of isoforms.
- Isoforms per Structural Category: number of isoforms in the different structural categories.
- Transcript Length per Structural Category: boxplots of the length of the different structural categories.
- Isoforms per Type of Gene: stacked barcharts with the number of isoforms per gene in annotated genes and novel genes.
- Exons per Structural Category: boxplots of the number of exons that are present in the different structural categories.
- **Distance to TSS and TTS:** these charts are related to the distance of Full Splice Match (FSM) and Incomplete Splice Match (ISM) transcripts to annotated Transcription Start Sites (TSS) and Transcription Termination Sites (TTS).
- Distances of FSM to TTS: histogram of the distribution of the distance of FSM isoforms to a TTS. If a polyA motif file is introduced, only isoforms with a polyA motif found will be shown in this chart.
- Distances of ISM to TTS: idem with ISM isoforms.
- Distances of FSM to TSS: histogram of the distribution of the distance of FSM isoforms to a TSS. If a CAGE annotation file is introduced, only isoforms within a CAGE will be shown in this chart.
- Distances of ISM to TSS: idem with ISM isoforms.
- **Filtering:** these charts are related to the removal of transcripts after filtering.
- Reasons of Transcripts Removal: percentage of isoforms removed because of intrapriming, RT Switching, or because of Low Coverage/non-canonical junctions. The current filtering rules are as follow:
    - If a transcript is FSM, then it is kept unless the 3' end is unreliable (intrapriming).
    - If a transcript is not FSM, then it is kept only if all of below are true:
    - 3' end is reliable.
    - Does not have a junction that is labeled as RTSwitching.
    - All junctions are either canonical or has short read coverage above the threshold that was set by the user.
    - Isoforms Before and After Filtering: stacked barcharts with the number of isoforms for each structural category before and after filtering.
- **Short and Long Read Coverage:** these charts are related to coverage analysis of transcripts by short reads and long reads.
- Short Read Coverage:stacked barchart with the number of transcripts whose splice junctions are covered by short reads by structural category.
- Long Read Counts on Genes: boxplots with the number of FL counts per type of gene (annotated or novel). The plotted value is the logarithm to base 2 of the Counts per Million plus 1.
- Long Read Counts on Categories: idem with structural categories.
- Saturation Plot: saturation plots are used to assess the performance of RNA-seq experiments and determine the optimal sequencing depth for a given experiment. Saturation plots are generated by plotting the number of unique transcripts detected versus the total number of reads (sequencing depth) for a given RNA-seq experiment. The vertical line shows the saturation point, that is to say, the point where the curve starts flattening (e.g when more sequencing depth is not going to make you find new transcripts).
- Increment Plot: this plot is similar to the other one, but shows the increase of discovered transcripts as sequencing depth is increased. Because of that, after the saturation point, the bars showing the increase of discovered transcripts should be shorter.
- **Redundancy Analysis:** these charts are related to redundancy analysis of transcripts. Redundancy reflects the fact that transcripts, after being validated, can often incorporate 3′ and 5′ end variability, which can lead to the detection of multiple FSM and/or ISM isoforms per reference transcript differing in their start and/or end positions. If a CAGE annotation file and/or a polyA motif file is/are introduced, that data is taken into account.
- Redundancy Analysis of FSM Transcripts
- Redundancy Analysis of ISM Transcripts
- Redundancy Analysis of Both Types
- **Quality Features:** these charts summarize features that represent good/bad quality of transcripts per structural categories.
- Good quality features: percentage of transcripts per structural category with different features that show good quality:
    - All_CJ: every splice junction is a canonical junction.
    - Annotated: Distance of query isoform 3' start to the closest start end of *any* transcripts of the matching gene is smaller than the distance introduced in Configuration 2 wizard (Figure 5)
    - Cage_support: if CAGE annotation file is added, this bar appear in the chart. Isoform is within a CAGE site.
    - Covered: if short reads files are introduced, this bar is added to the chart. Isoform has short read coverage.
    - PolyA_support: if polyA motif file is introduced, this bar is added to the chart. Isoform has a polyA motif.
    - Bad quality features: percentage of transcripts per structural category with different features that show bad quality:
    - RTSwitching: isoform has at least one junction that might be marked as RT Switching. This information only appear if the filtering step has run.
    - Non_all_CJ: not every splice junction is a canonical junction.
    - Non_covered: if short reads files are introduced, this bar is added to the chart. Isoform has not read coverage.
    - Predicted_NMD: isoform's predicted ORF and CDS ends at least 50bp before the last junction. This information only appear if the filtering step has run.
    - **Rescue Step:** this charts give information about the rescue step:
- Categories used to rescue transcripts: barchart with the number of isoforms per category used to rescue a reference transcript.
- Reasons of artifact exclusion: reasons to finally not use an artifact to rescue a reference transcript.

Files

- **Transcriptome Annotation File:** this GTF annotation file shows the location in the genome of every transcript and exon.
- **Transcriptome File:** Multifasta file with the nucleotide sequence of every isoform.
- **Translated Transcriptome File (optional):** Multifasta file with the translated amino acidic sequence of every isoform. This file can only be returned if ORF prediction is not deactivated.
- **Junctions File:** Text file with information of every splice junction found.
- **Classification File:** Text file with information of every isoform found.

If the rescue step is performed, the Transcriptome Annotation File will be an extended version with rescued reference transcripts.

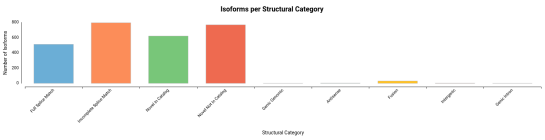


Figure 11. *Isoforms per Structural Category Chart*

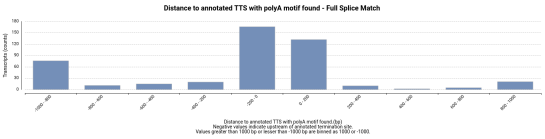


Figure 12. *Histogram of Distances of FSM to annotated TTS*

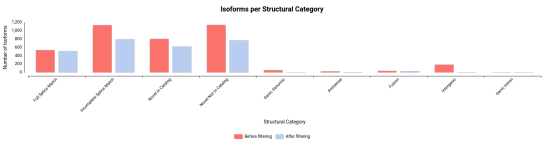


Figure 13. *Isoforms per Structural Category Before and After Filtering*

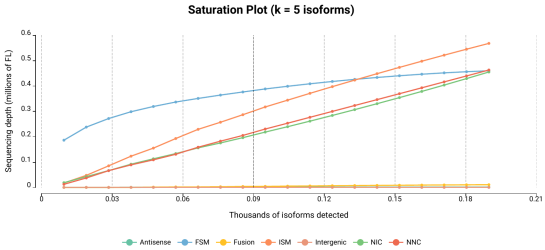


Figure 14. *Saturation Lines per Structural Category with 5 FL-reads required to call an isoform*

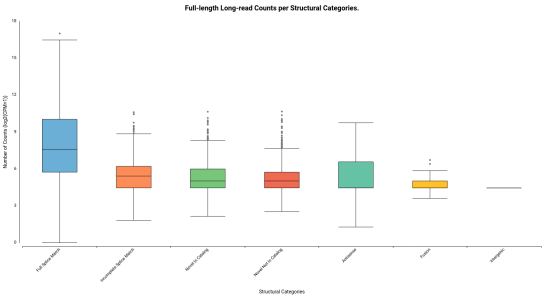


Figure 15. *Boxplots of FL Counts per Structural Category*

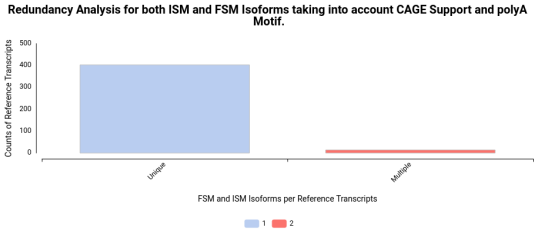


Figure 16. *Redundancy Analysis for FSM and ISM*

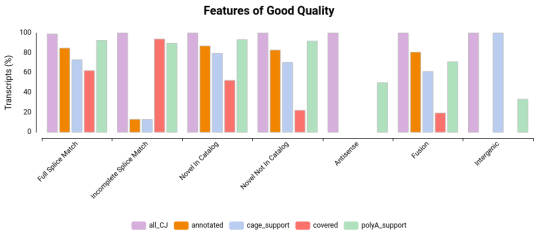
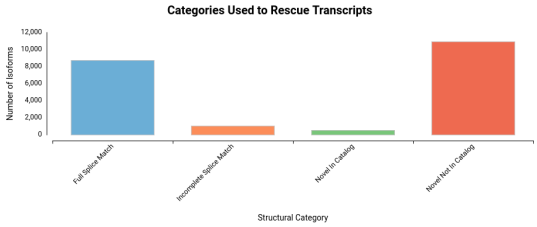


Figure 17. *Stacked barcharts with percentages of features of good quality*



Isoform extraction from FASTA

Another useful tool that has been implemented is the extraction of isoforms from the transcriptome file that SQANTI3 returns. To use this tool, just select every isoform whose sequence is needed and right click in one of them and then click in "Extract Isoforms from FASTA". Then a wizard will open (Figure 19), and the FASTA transcriptome file and the output folder must be introduced. The output filename will be *extracted*_transcriptome-file-name.fasta.

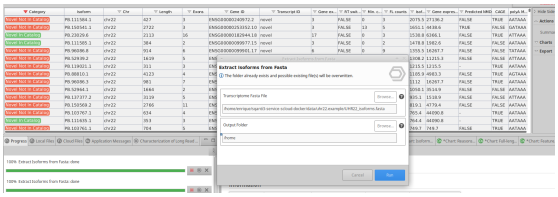


Figure 19. *Extract Isoforms from FASTA Wizard*

Combining Transcriptomes with TAMA Merge

INTRODUCTION

TAMA Merge is a computational tool which can merge multiple transcriptomes into a combined one. It does this by merging transcript models within a given range of similarity. This tool may be useful for the following purposes:

- Extending a reference annotation with novel transcript models, e.g. as defined by FLAIR.
- Comparing two transcriptomes by merging them, thereby creating a joint set of transcript model identities.
- On data sets with biological replicates, merging multiple transcriptomes defined on individual samples into a common transcriptome ("Call & Join" approach to isoform identification for biological replicates).

Please cite TAMA Merge as:

Kuo, Richard I., et al. "Illuminating the dark side of the human transcriptome with long read transcript sequencing." *BMC genomics* 21 (2020): 1-22.

TODO: add images

RUN TAMA MERGE

TAMA Merge can be found under **Transcriptomics → Long-Reads Analysis → Combining Transcriptomes with TAMA Merge.** The wizard consists of 3 pages and facilitates the definition of the input and output options as well as the merging parameters.

Input

TAMA Merge receives the following inputs:

- **Transcriptome:** Transcript annotations in BED12 or GTF format. At least one file needs to be provided.
- **Give Priority to...:** Out of the provided input files, one may be chosen as a to give priority to. Doing this will give its transcription start and end sites, as well as splice sites priority over the other files. This is recommended e.g. for combining a reference transcriptome annotation with a custom generated one. It will also cause the transfer of gene and transcript IDs into the merged transcriptome. If "None" is selected here, all provided transcriptomes will have equal priorities.
- Note that supplied BED12 files have to use the following format in their 4th column ("name"): "gene_id;transcript_id". As this format is not very commonly used, we generally recommend the use of .gtf files as inputs, which will automatically be converted into a suitable .bed file.
- Though generally intended to merge multiple transcriptomes, running TAMA Merge on only one transcriptome is also possible and may make sense if the goal is to collapse similar transcript models.

Algorithm Options

This page provides some more detailed options to configure the algorithm:

- **Capped:** Defines whether transcript start sites in the provided transcriptomes can be trusted, or whether shorter transcripts should always be merged into longer transcripts. This is generally recommended for merging transcriptomes created from tools such as FLAIR or IsoQuant, as these already implement their own logic for determining transcription start sites.
- **Exon Ends:** Whether the last exons (start and end) of transcript models should be chosen based on the most common or the longest exon.
- **5' Threshold:** The threshold in base pairs for the five prime end within which transcript models should be merged.
- **Splice Junction Threshold:** The threshold in base pairs for the splice junctions of transcript models.
- **3' Threshold:** The threshold in base pairs for the three prime end within which transcript models should be merged.

Note that transcript models within one transcriptome which fall within the given thresholds will also be combined, even for the selected reference.

Output

- **Output File Prefix:** Set a name which will serve as a prefix for all output files.
- **Output Directory:** Define a directory (existing or new) in which to save the output files.
- **Save Merged Transcriptome as .bed:** Whether to save the merged transcriptome as a .bed file.
- **Save Merged Transcriptome as .gtf:** Whether to save the merged transcriptome as a .gtf file.

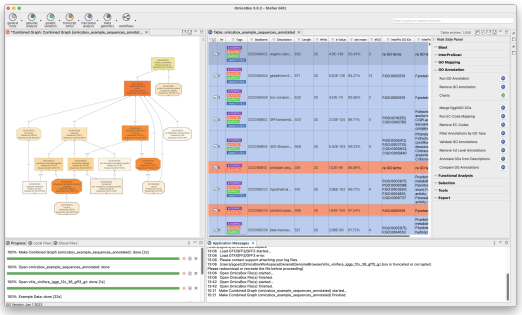
RESULTS

TAMA Merge has the following outputs:

- **Merged Transcriptome as .gtf and/or .bed file:** The main output of TAMA Merge is the set of transcript annotations produced by the merge.
- **Summary Report** with information on the number of genes as well as the number of transcripts before and after the merging process.
- **Merge Report .txt file** which maps the transcript IDs of the input files to the transcript IDs in the merged transcriptome.
- **Gene Report .txt file** which contains information about each gene, e.g. how many transcripts it had before and after the merge.
- **Transcript Report .txt file** which contains information about each transcript, e.g. which source transcripts from which files were merged to create it.

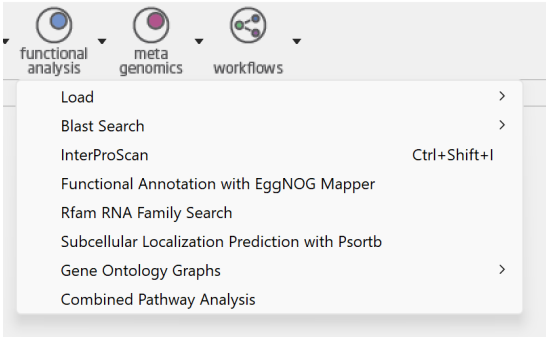
4.5 Module Functional Analysis

4.5.1 Module Functional Analysis



The OmicsBox Functional Analysis module is a well-known comprehensive bioinformatics tool for functional annotation.

- **High-Throughput Blast and InterProScan**: Utilize CloudBlast and CloudInterProScan for rapid sequence alignments and domain searches against customizable reference datasets.
- **Gene Ontology Mapping:** Link potential homologs and domains with available functional annotation from the latest well-curated databases provided by UniProt and Gene Ontology consortia.
- **Blast2GO Annotation:** The Blast2GO methodology allows to flexibly assign the most reliable functional labels to novel sequence datasets, taking into account source annotation quality and ontology hierarchies.
- **Enrichment Analysis:** Use different enrichment analysis approaches (Fisher Exact Test and GSEA) to identify over and under-represented molecular functions.
- **Combined Pathway Analysis:** Identify Reactome and KEGG pathways for any set of sequences, utilize the differential expression data to compute pathway enrichment using (GSEA), and benefit from a combined visualization for easier insights.



Additional Resources

- Functional Annotation Analysis use case: https://www.biobam.com/whole-genome-functional-annotation-of-solanum-lycopersicum/ .

Functional Analysis Example Dataset: https://resources.biobam.com/omicsbox/example_data/version_2_0_0/FunctionalAnalysis.zip

4.5.2 Load Data

Introduction

In order to start the analysis with the Functional Analysis module, there is the need to load data to OmicsBox.
It is possible to load different types of files such as FASTA, XML Blast results, XML InterProScan results as well as annotation (GOs) files.
When loading one of the above-mentioned files to OmicsBox a new project will be generated and the functional analysis features will become available.

FASTA FILE FORMAT

A sequence in FASTA format begins with a single-line description or header starting with a ">" character. The rest of the header line is arbitrary but should be informative. Subsequent lines contain the sequence, one character per residue. Lines can have different lengths. Be sure your file is in this format and avoid strange characters in the sequence header, such as '&' or '\', and use 'N' to denote in-determinations in the sequences.

An example of the FASTA format:

```
>gi|121664|sp|P00435|GSHC BOVIN GLUTATHIONE PEROXIDASE
MCAAQRSAAALAAAAPRTVYAFSARPLAGGEPFNLSSLRGKVLLIENVASLUGTTVRDYTQMND
LQRLGPRGLVVLGFPCNQFGHQENAKNEEILNCLKYVRPGGG
```

XML FILE FORMAT

An Extensible Markup Language (XML) file is a markup language file that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. In an XML file, there are both tags and text. The tags provide the structure to the data. The text in the file that you wish to store is surrounded by these tags, which adhere to specific syntax guidelines.

An example of an XML file:

```
<?xml version="1.0"?>
<BlastXML2
xmlns="http://www.ncbi.nlm.nih.gov"
xmlns:xi="http://www.w3.org/2003/XInclude"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:schemaLocation="http://www.ncbi.nlm.nih.gov http://www.ncbi.nlm.nih.gov/data_specs/schema_alt/NCBI_BlastOutput2.xsd">
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_1.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_2.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_3.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_4.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_5.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_6.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_7.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_8.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_9.xml"/>
    <xi:include href="0e76513c-1bfa-11ea-ad7e-06dd694a34b4_10.xml"/>
</BlastXML2>
```

Load files

The load feature can be found under **Functional Analysis → File → Load**.

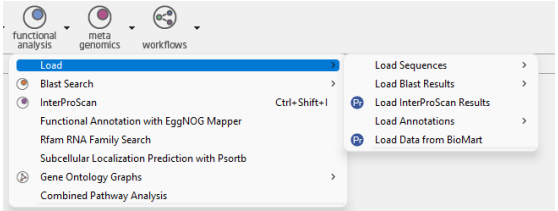


***Figure 1**: Load*

Load Sequences

LOAD FASTA FILE (.FASTA)

The application accepts text files containing one or more DNA or protein sequences in FASTA format. These files must have the extension *.fasta,.fnn,.faa,.fna,.ffn, or .txt.* to be accepted by the application.

LOAD FASTA FROM REFERENCE + GFF/GTF

Extract and import sequences from a genome FASTA and a GFF/GTF file (figure 3).
For further information, please the blog here.

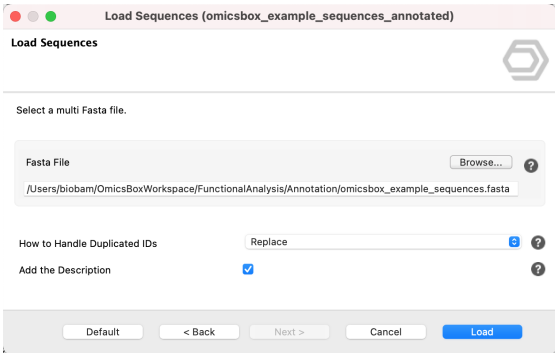


Figure 2: *Load Sequences Dialog: Choose Fasta file*

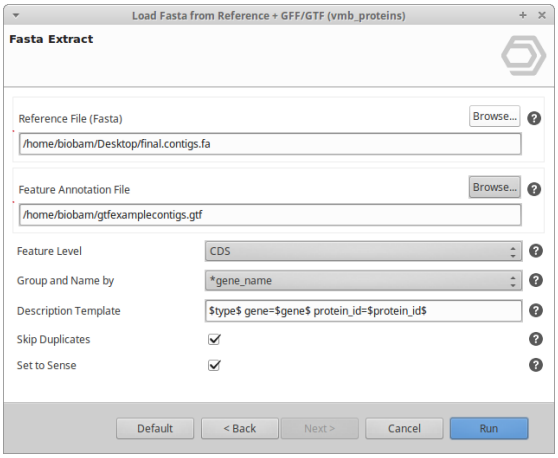


Figure 3: *Extract and import sequences from a FASTA and a GFF/GTF file.*

SHOW SEQUENCE RESULTS

Once the sequences have been loaded to OmicsBox, it is possible to see them by right-clicking on the Sequence Table. The "Single Sequence Menu" (context menu) will appear (figure 4). This menu provides some functions for sequences individually, i.e. will apply to the sequence at that position of the Table.
With the sequence viewer, it is possible to copy the sequence to the clipboard.

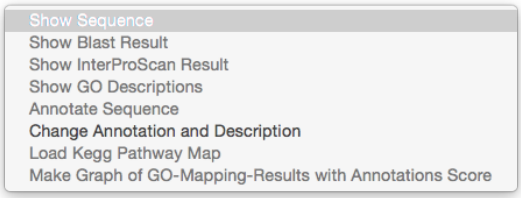


Figure 4: *Context Menu: Show Sequence*

Figure 5: *Sequence Viewer*

Load BLAST Results

If a BLAST result is already available in XML format, it can be directly loaded into OmicsBox by using the Load Blast Results. You can choose the Blast format to import as an XML (Load Blast XML Legacy) or as the new XML2/JSON files (Load Blast Results XML/Zip). These new formats can be loaded as a Zip file.
In the Load Blast Results dialog, a whole directory containing a collection of BLAST XML files or a single XML file can be selected figure 6. The BLAST results will be added to your current OmicsBox session.
OmicsBox also allows the input of TimeLogic DeCypher Blast results.

If a new project is being generated when loading the XML Blast results then no sequence information is available. To confirm this, the show sequence will be disabled in the context menu. The sequences can be added to the existing project.

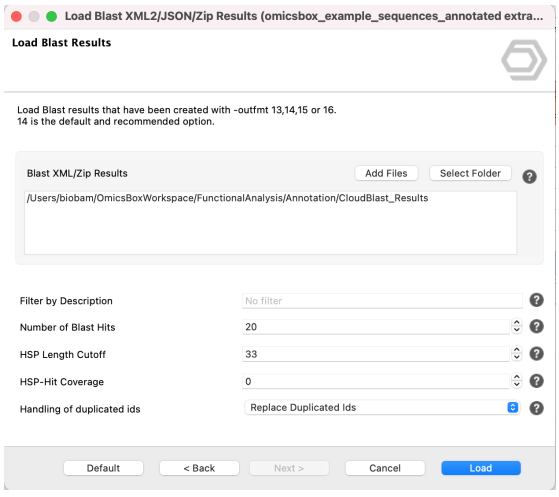


Figure 6: *Load / Import Blast Results*

Load InterProScan Results

The InterProScan results saved in XML format can be loaded into the current OmicsBox project or generate a new one.

When loading the InterProScan results it is possible to select the input format.

- Protein - If InterProScan has been performed inside OmicsBox (OmicsBox translates the nucleotide sequences to the longest ORF peptides)
- Nucleotides - If InterProScan has been performed with nucleotide sequences and InterProScan binaries.

If a new project is being generated when loading the InterProScan results, then the sequence information loaded to OmicsBox is protein. o confirm this, the show sequence will be disabled in the context menu.

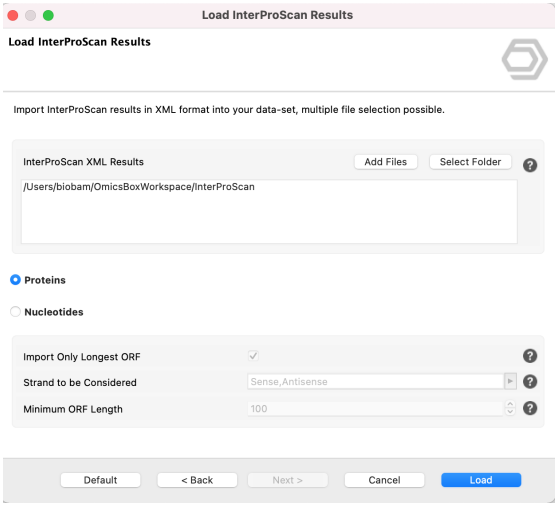


Figure 7: *Load InterProScan Results*

Load Annotations

LOAD ANNOTATIONS (.ANNOT)

Already made or existent annotation can be imported using the .annot format. For import purposes only, the .annot format allows also multiple annotations of the same sequence to be given in one single row, separated by commas, as shown above (Schema: Seq-Name GO(s) or EC(s) Sequence description).

LOAD SEQUENCE DATA/ ANNOTATION

This load option expects a text file with identifiers and connects directly to NCBI and retrieves the corresponding sequence information and annotations.
The text file provided to load should have two columns separated by a tab, where the first column should be the identifiers (locus, proteins) and the other the taxonomy identifier. For further information, please visit the blog here.

LOAD NETAFFY ANNOTATIONS

It is possible to load annotation files provided by Affymetrix. These files have to be in CSV format and contain the probe IDs annotated.

An example can be downloaded from here: ATH1-121501 Annotations, CSV format, Release 36 (8.7 MB, 4/13/16)

In all these options when requesting to only load the annotation information then no sequence information is available. To confirm this, the show sequence will be disabled in the context menu. The sequences can be added to the existing project.

OmicsBox Annotation File (.annot):

```
Seq1 GO:0001234 glycolipid transfer protein-like
Seq1 GO:0001264,GO:0004567,...
Seq1 GO:0034567
Seq1 EC:2.1.2.10
Seq2 GO:0001234,... sorbitol transporter
Seq2 GO:0001244
Seq3 GO:0001234,GO:0004567,GO:0009123
Seq3 EC:1.2.4.1, EC:3.1
....
```

Example text file to be used with Load Sequence Data/ Annotation:

```
AT1G15520 3702
AT1G18900 3702
AT5G14970 3702
```

Load Data from BioMart

This feature allows retrieving gene/protein sequences as well as the annotation directly from Ensembl BioMart using a list of identifiers (figure 8).
With this tool, there is the need to know the Mart, database, and the type of identifiers one has.
For further information, please the blog here.

Once the sequences have been loaded to OmicsBox it is possible to see the sequences as well as see the Sequence Length Distribution chart.

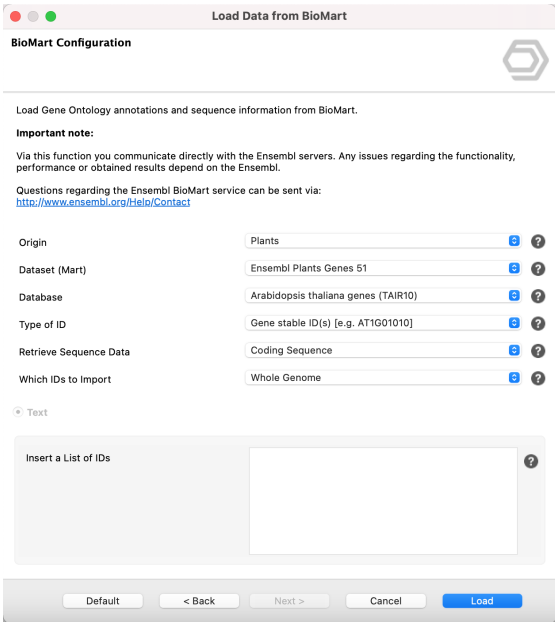


Figure 8: *Load Data from BioMart*

Add sequences to the existing OmicsBox project

In case the loaded project file has only Blast results and no sequence information it is still possible to add the corresponding sequences to the OmicsBox project by clicking on the arrow next to the **Start** icon and selecting **Load Sequences (.fasta)**. Now two options will be displayed Create a new project and Add to the existing project, see figure 9. The "Add to the existing project" option should be selected and on the next page, you can browse for the FASTA file, and the "Replace" option should be selected, see figure 2.

Use the File Manager context menu to merge into the project. Select two or more projects, open the context menu with a right-click on one of the files and select **Merge**.

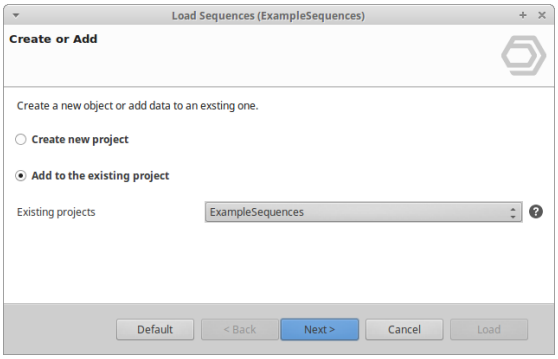


Figure 9: *Load Sequences Dialog: Add sequences to an existing project*

Export FASTA Sequence with Annotation Results

After executing the whole functional annotation of the sequences in OmicsBox (BLAST, Mapping and Annotation) it is possible to export the sequence in FASTA format with the corresponding sequence description and GO ID or GO term from the Side Panel → Export → Export as FASTA.

An example of the Exported FASTA format with Sequence Description and GO IDs:

```
>C04018C10|mitogen-activated protein kinase 3|GO:0005634;GO:0004707;GO:0005515
acaaacgagagcgtagaaaattaattagagagaaaaagagagagagtaaaatggctgacgtggcgcaggtcaacg
gcgtaggtcaaacggctgattttcctgcggtaccgacgcacggcggtcagtttatacagtacaatatatttggaa
acttgtttgaaatcacggccaagtatcggcctccgatcatgccgattggtcgcggcgcgtacgggatcgtttgct
cggtgttgaatacggagacgaatgagctcgttgcgatgaagaaaatagcgaacgcttttgataatcacatggatg
ccaagcgaacgcttcgtgagattaagcttctgcaacatttcgatcatgaaaatgtgatagctgtaaaagatgtgg
ttcccccaccgttacgaagagaattcactgatgtctatattgctgcggaactcatggacactgacctttaccaaa
ttattcgctcaaatcaaagtttatccgaggagcactgccagtatttcttgtatcaacttcttcgaggactcaagt
atatccattcagcaaatgttattcatcgggatttgaagcccagcaatctcttgttaaatgcaaattgtgatttaa
```

```
aaatttgtgattttggtcttgctcgtccaacctcggagaatgagttcatgacagagtatgttgtcacaagatggt
accgagcccctgagttgttattgaactcatctgactacactg
```

4.5.3 BLAST

BLAST

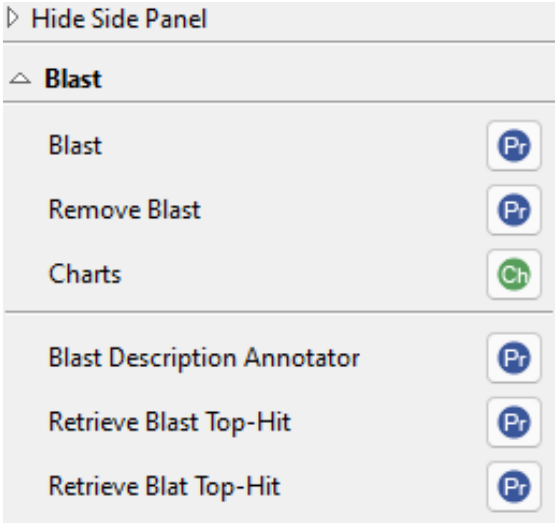


Figure 1: *Blast side panel section*

INTRODUCTION

OmicsBox uses the Basic Local Alignment Search Tool (BLAST) to find sequences similar to your query set. Please, refer to http://www.ncbi.nlm.nih.gov/BLAST for details on the BLAST function. Figure 6, shows the BLAST Configuration Dialog Window that controls the BLAST step.

BLAST in OmicsBox can basically be performed in 5 different ways:

1. Diamond Blast. Utilize the OmicsBox dedicated cloud infrastructure to run Diamond Blast, an effective community resource for quick and secure sequence alignments designed for larger datasets (5000+ sequences).
2. CloudBlast. This is a cloud-based OmicsBox Community Resource for massive sequence alignment tasks. It allows you to execute standard NCBI Blast+ searches directly from within OmicsBox in a dedicated computing cloud. CloudBlast is a high-performance, secure and cost-optimized solution for your analysis. This is a blast service totally independent from the NCBI servers to provide fast and reliable sequence alignments. Please see Run Blast using CloudBLAST section for more information.
3. QBlast@NCBI. NCBI offers a public service that allows searching molecular sequence databases with the BLAST algorithm. The main advantages of making use of this service are its versatility and that no database maintenance is required. Therefore by selecting this option at OmicsBox no additional installations have to be done.
4. Local BLAST against its own database. It is possible to use BLAST+ executable to query a local/own database. At **https://www.blast2go.com/make-own-database-and-blast** and at the Make Blast Database section one can see how to prepare and blast locally an own fasta database.
5. Custom Database CloudBlast. It is possible to run BLAST against a database made of a custom protein fasta file using the OmicsBox Cloud resources.

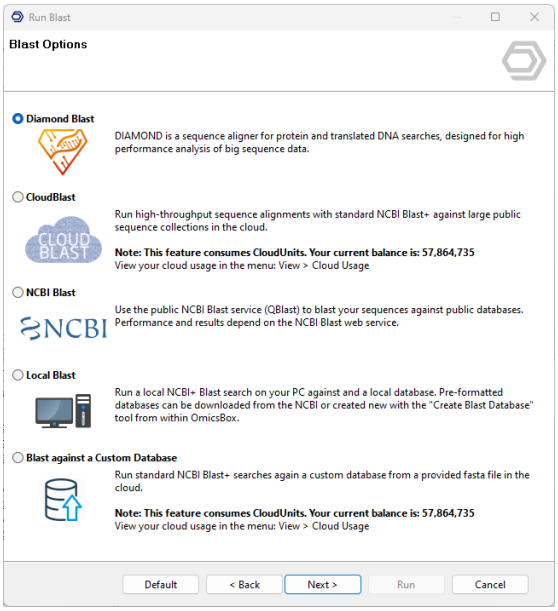


Figure 2: *Choose between Diamond, CloudBlast, NCBI, Local, or Custom Database*

BLAST

The Blast functionalities can be found under **functional analysis → Blast → Run Blast** or from the **Side Panel** if a sequence has been loaded in OmicsBox.The wizard allows for adjustment of analysis parameters, which are divided into three different sections: Blast Configuration in figure 6, Advanced in figure 7 and Save Results Page figure 8.

DIAMOND BLAST

Diamond is an alternative to the official NCBI Blast software. Developed for high-performance analysis of large sequence data, DIAMOND is a sequence aligner for protein and translated DNA searches. Key characteristics include:

- Protein and translated DNA pairwise alignment at speeds 100x–10,000x faster than BLAST.
- Alignments for frameshifts in long read analyses.

This makes Diamond especially useful when dealing with bigger datasets (5000+ query sequences).

DIAMOND is currently developed by Benjamin Buchfink at the Drost lab, Max Planck Institute for Biology, Tübingen, Germany (since 2019).

Diamond Blast Configuration Page

- BLAST Mode: The algorithm you want to use:
- blastp - Compares an amino acid query sequence against a protein sequence database.
- blastx - Compares a nucleotide query sequence translated in all reading frames against a protein sequence database. Used to find potential translation products of an unknown nucleotide sequence
- BLAST DB: The name of the database to search in eg. nr, SwissProt, RefSeq.
- Taxonomy Filter: Search for Blast results only in the selected taxonomy.

- BLAST expect value: The statistical significance threshold for reporting matches against database sequences. If the statistical significance ascribed to a match is greater than the EXPECT threshold, the match will not be reported. Lower EXPECT thresholds are more stringent, leading to fewer chance matches being reported. Increasing the threshold shows less stringent matches.
- Number of BLAST hits: The number of alignments you want to achieve (0-100).
- HSP Length CutOff: A Cutoff value for the minimal length of the first HSP of a blast hit, used to exclude hits with only small local alignments from the BLAST result. The given length corresponds to amino acids or nucleotides depending on the type of performed BLAST.
- HSP-Hit Coverage

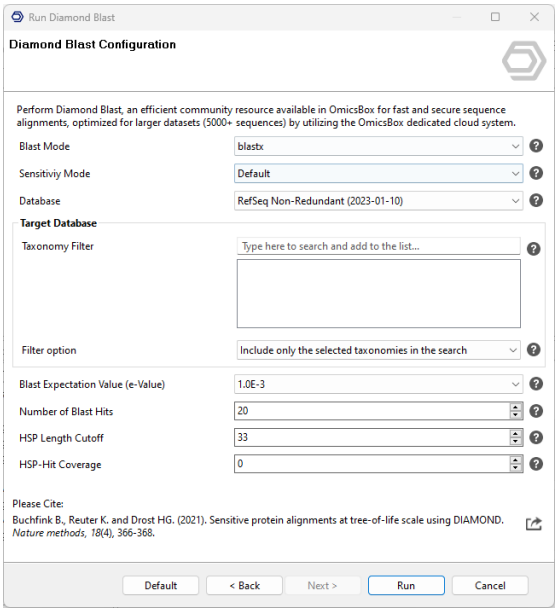


Figure 3: *Diamond Blast Configuration Page*

Custom Diamond Database

This is a step-by-step instruction on how to upload and use a custom Diamond database for Diamond Blast in OmicsBox.

Prerequisites: Before you begin, please ensure the following:

- You have access to a custom Diamond database in the form of a single .dmnd file.
- The custom database has been properly formatted for Diamond Blast. OmicsBox does not accept raw data or Fasta files for database uploads.

Procedure: Uploading a Custom Diamond Database

1. **Access the Cloud Files Tab:**
2. Open OmicsBox.
3. On the left-hand side of the OmicsBox interface, you will find a panel with different tabs. Click on the "Cloud Files" tab to access your cloud storage.
4. **Create a New Folder:**
5. In the "Cloud Files" tab, navigate to the location where you want to upload your custom Diamond database.
6. Right-click on the desired location, and a context menu will appear.
7. Select "New Folder" from the context menu. A new folder will be created.
8. **Upload the Custom Diamond Database:**
9. Locate the custom Diamond database file on your local computer.
10. Simply drag and drop the .dmnd file into the newly created folder in the "Cloud Files" tab.

Using the Custom Diamond Database for Diamond Blast

To perform a Diamond Blast search using your custom database, start by selecting the annotation project and navigating to the Diamond Blast dialog from the side panel. Within the Diamond Blast settings, locate the "Database" drop-down control, and upon clicking it, a list of available databases will appear. From this list, you will

find the custom Diamond database that you uploaded in a previous step. Simply select your custom database by clicking on it. Once your custom database is chosen, proceed to configure any other pertinent parameters and settings to tailor your Diamond Blast analysis to your specific requirements.

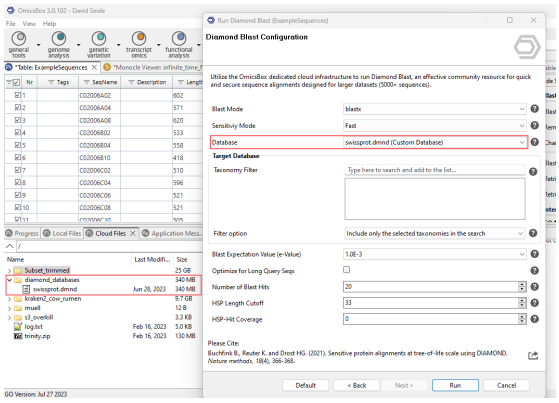


Figure 4: *Diamond Blast Configuration Page selecting a Custom Database from the user's Cloud Files.*

CLOUD BLAST

CloudBlast offers a highly optimized, self-sustained HPC solution to address a very specific need of the OmicsBox community.
CloudBlast is a BLAST service totally independent of the NCBI servers to provide fast and reliable sequence alignments. It consists of a high performance computing cluster dedicated exclusively to Blast searches.

All OmicsBox subscriptions include "Cloud Units" to make use of this resource and allows you to perform blast searches for tens of thousands of sequences within a few days against a large collection of protein databases.

These units correspond directly to the usage of the cluster (used CPU seconds and network traffic/data volume).

Each sequence alignment performed in the system consumes a certain amount of computation time depending on the sequence length and the blast algorithm (blastx, blastp) and the parameters used. The smaller the database you blast against the more sequences you can analyse with 6.000.000 Cloud Units (see Cloud Usage in the View Menu section to know how to monitor the Cloud Units). This means that e.g. if you blast against the vertebrate NR-subset you would be able to blast approx. one million (1.000.000) sequences. If you decide to blast against the NR database, the largest protein database available, it should allow you to blast approx. 80.000 sequences (with an average length of 800nt per sequence). One has to add the Species taxonomy id to blast against an NR-subset.

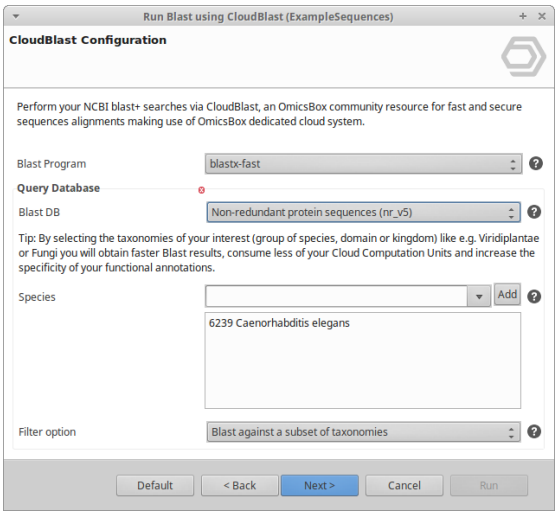


Figure 5: *CloudBlast Configuration Page*

For the advanced and save parameters page please see Advanced Page and Save Results Page sections for detailed information.

NCBI BLAST

Blast Configuration Page

- Your e-mail address in case you are using the NCBI BLAST web service.
- BLAST program: The algorithm you want to use:

- blastp - Compares an amino acid query sequence against a protein sequence database.
- blastn (-task blastn) - Compares a nucleotide query sequence against a nucleotide sequence database.
- blastx - Compares a nucleotide query sequence translated in all reading frames against a protein sequence database. Used to find potential translation products of an unknown nucleotide sequence
- tblastn - Compares a protein query sequence against a nucleotide sequence database dynamically translated in all reading frames.
- blastx-fast
- blastp-fast
- blastp-short
- blastn (-task megablast)
- blastn (-task dc-megablast)
- blastn-short
- tblastn-fast
- BLAST DB: The name of the database to search in (eg. nr, SwissProt, PDB). To see a list of possible DBs at NCBI seehttp://data.biobam.com/ncbi_blast_dbs_protein.pdf
- Taxonomy Filter: Search for Blast results only in the selected taxonomy.
- BLAST expect value: The statistical significance threshold for reporting matches against database sequences. If the statistical significance ascribed to a match is greater than the EXPECT threshold, the match will not be reported. Lower EXPECT thresholds are more stringent, leading to fewer chance matches being reported. Increasing the threshold shows less stringent matches.
- Number of BLAST hits: The number of alignments you want to achieve (0-100).

BLAST Description Annotator: The BDA finds the best possible description for a new sequence based on a given BLAST result.

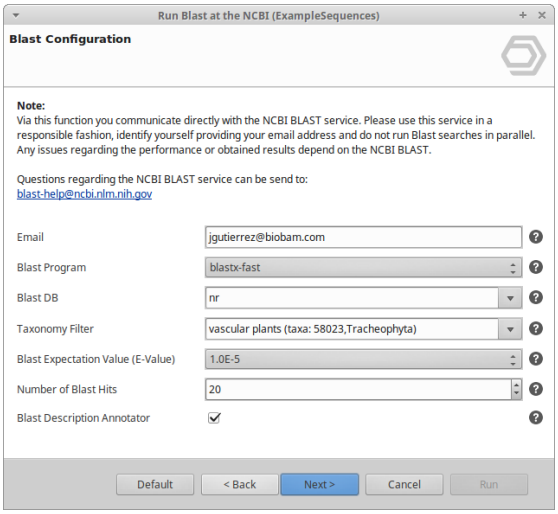


Figure 6: *NCBI Blast Configuration Page*

Advanced Page

- Blast Parameters:
- Word size: One of the important parameters governing the sensitivity of BLAST searches is the length of the initial words. The word size is adjustable in blastn and can be reduced from the default value to increase sensitivity. This word size can also be increased to increase the search speed and limit the number of database hits.
- Low complexity filter: The BLAST programs employ the SEG algorithm to filter low complexity regions from proteins before executing a database search. The default is ON.
- Filter Options:
- HSP length cutoff: A Cutoff value for the minimal length of the first HSP of a blast hit, used to exclude hits with only small local alignments from the BLAST result. The given length corresponds to amino acids or nucleotides depending on the type of performed BLAST.
- HSP-Hit Coverage

• Filter by description: Filter-out Blast hits by a description

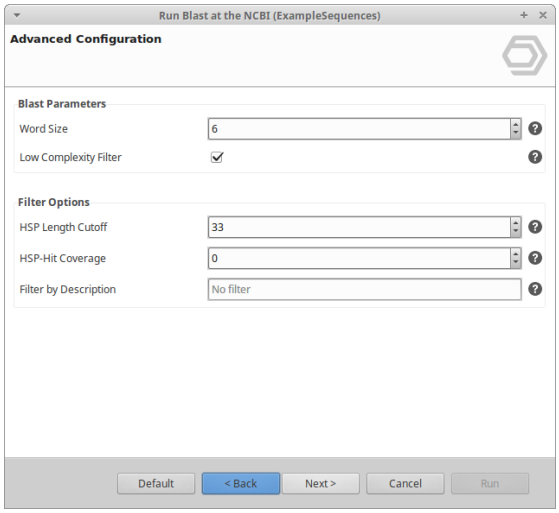


Figure 7: *Advanced Configuration Page*

Save Results Page

The results of the BLAST queries can also be directly saved to a file in different formats by selecting the corresponding checkboxes at the BLAST Save Results Page. If the chosen file already exists, upcoming results will be appended. Choose a format type to additionally save your BLAST results.

• XML2: This is a new BLAST result provided by NCBI and can also be loaded into OmicsBox.

• XML: It is recommended to save your BLAST results as XML as this format is supported by the OmicsBox Load BLAST Results function.

• TXT: It saves the blast results of each sequence in text file format.

• HTML: For each sequence, a file in HTML format will be saved.

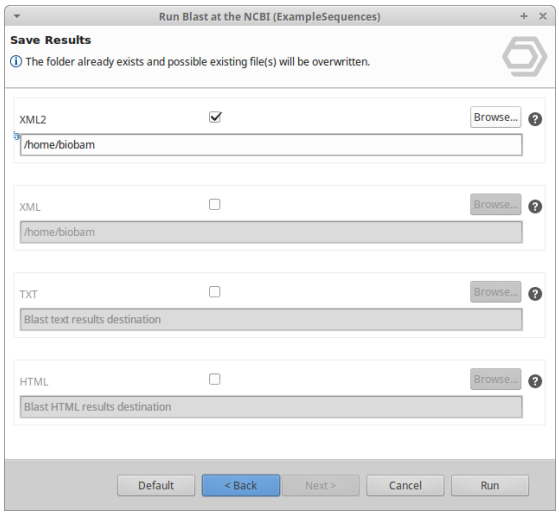


Figure 8: *Save Results Page*

LOCAL BLAST

With Local BLAST you can blast the sequences against your own database. OmicsBox allows creating a Blast database from a FASTA file with the option "Make Blast Database" (see Make Blast Database section). Download and format your database and choose the corresponding folder to see figure 9. Databases have to be formatted for NCBI Blast+.

The main parameters in the Local BLAST Configuration page are very similar to the ones in NCBI and CloudBlast. The main difference is when choosing the database as OmicsBox is expecting a *.pal*' file or .psq. On the Advanced Page at the "Run Parameters," it is possible to select the number of threads to be used. This field has not to be set up as OmicsBox detects the number of threads in the computer. The Advanced Page section provides a detailed description of each parameter. As in CloudBlast, the BLAST results will be saved in XML file format.

Visit the following tutorial on how to download NCBI pre-formatted databases.

Please cite NCBI for Local Blast and pre-formatted databases https://www.ncbi.nlm.nih.gov/books/NBK569850/ .

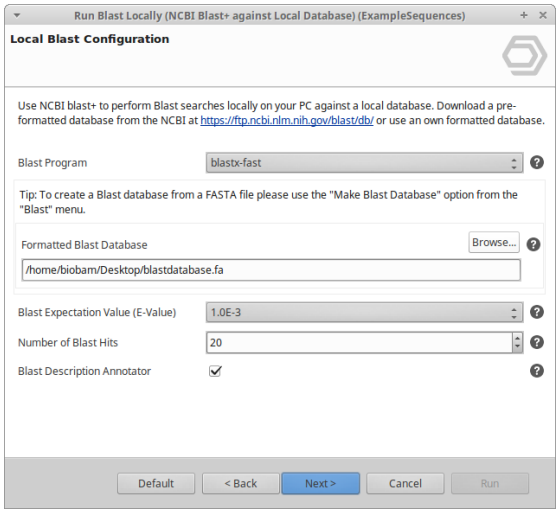


Figure 9: *Local Blast Configuration Page*

CUSTOM DATABASE CLOUDBLAST

OmicsBox offers the possibility to generate your own custom database from a .FASTA file and run Blast on the OmicsBox Cloud.
The database will be automatically generated in the Cloud using the Fasta file and the parameters provided. When running Custom Database CloudBlast Cloud Units will be consumed. More information on Cloud Units can be found online or under the CloudBlast section.

RESULTS

As the BLAST search progresses, sequences with successful BLAST results change their color on the Main Sequence Table from white to **orange** and the BLAST result-related columns will be filled. In case no results could be retrieved for a given sequence, this row will turn **dark-red**.

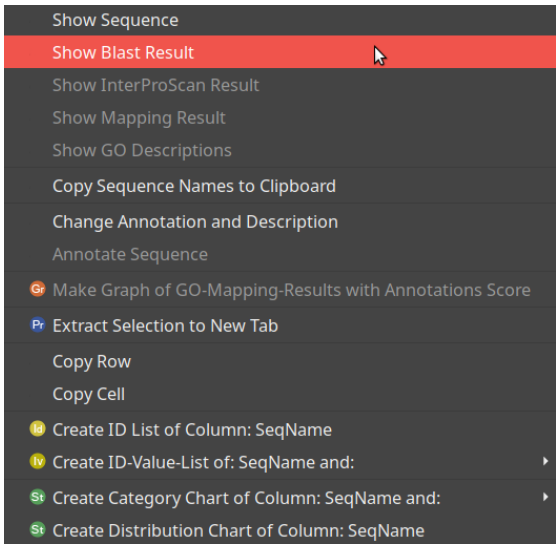


Figure 10. *Show BLAST results.*

Individual Blast Results

With a mouse the right click on a sequence, the Single Sequence Menu will be displayed and it is possible to see the BLAST results for each sequence individually. Show BLAST Results (figure 10) will generate a tab in the Results containing information on the results of the similarity search of the selected sequence. For each of the

obtained hits, the following information is given: Hit id and definition Gene name assigned to the hit by its accession e-value of the alignment Alignment length of the longest hsp Positive matches of the longest hsp Hsp similarity of hit: Number of hsps mapped GO-Terms with its evidence code UniProt codes of the hit sequences.

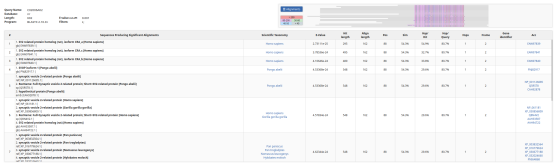


Figure 11. *Individual BLAST Result Table View*

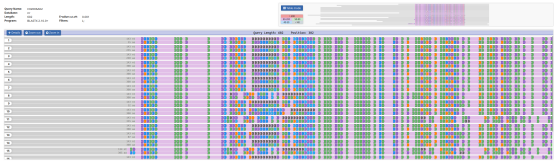


Figure 12*. Individual BLAST Result in Alignment View.*

Remove Blast

This option will remove the BLAST results from the selected sequences. It is possible to also remove the description of the sequences or leave them.

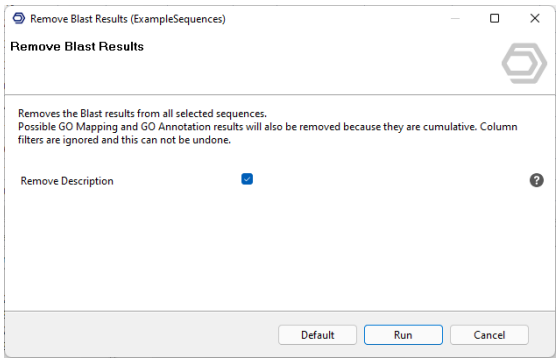


Figure 13: *Remove Blast Results*

Charts

Different BLAST statistics charts (figure 15, figure 16 and figure 17) can be generated for a global visualization of the results. These charts provide a general view of the similarity of the query set with the selected databases and can be used to choose cut-off levels for the e-value, similarity and annotation threshold parameters at the annotation step.

- E-Value Distribution: This chart plots the distribution of E-values for all selected BLAST hits. It is useful to evaluate the success of the alignment for a given sequence database and help to adjust the E-Value cutoff in the annotation step. It is possible to see that in figure 14 there are almost 250 hits with an e-value around 1e-25. It can be used in the annotation rule.
- Similarity Distribution: This chart displays the distribution of all calculated sequence similarities (percentages), shows the overall performance of the alignments and helps to adjust the annotation score in the annotation step. By looking at figure 15 it is possible to get an idea of how similar the query is to the hits. Knowing the overall similarity of the query sequences to the dataset can help decide whether to use a more or less restrictive Annotation CutOff. The smaller the similarity, the smaller the Annotation CutOff should be. This is not the only factor influencing the Annotation Score.
- Species Distribution: This chart gives a listing of the different species to which most sequences were aligned during the BLAST step.
- Top-Hit Species Distribution: Bar chart showing the species distribution of all Top-Blast hits.
- Hit Distribution: This chart shows a distribution of the number of hits for the blasted sequences in a data set.
- Hsp Distribution: This bar chart shows the distribution of hsps per hit.
- Hsp/Seq Distribution: This chart shows a distribution of percentages that represents the coverage between the hsps and their corresponding sequences.
- Hsp/Hit Distribution: Same as above but for hits instead of sequences.

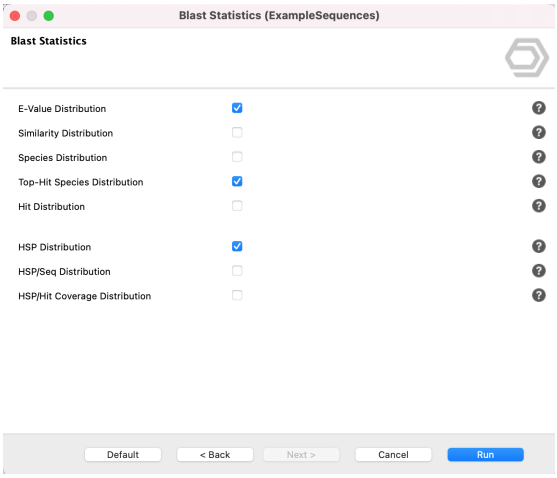


Figure 14: *Blast Statistics*

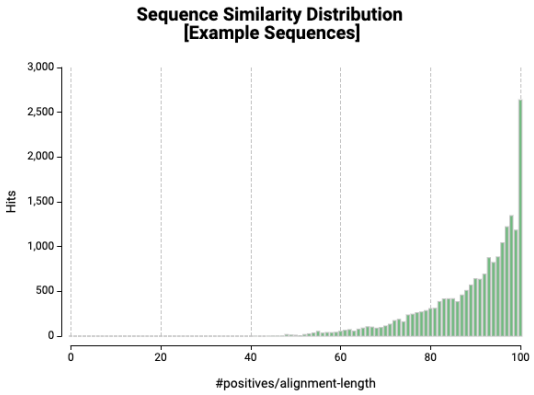


Figure 15: *Similarity Distribution*

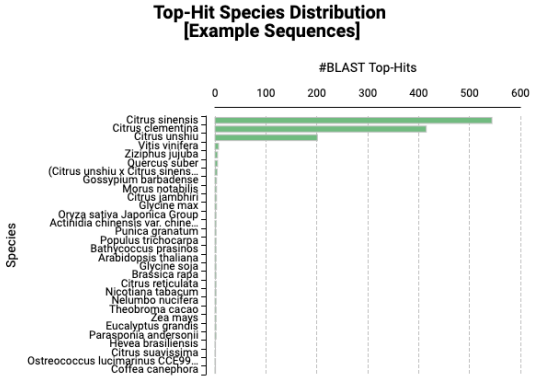


Figure 16: *Top-Hit Species Distribution*

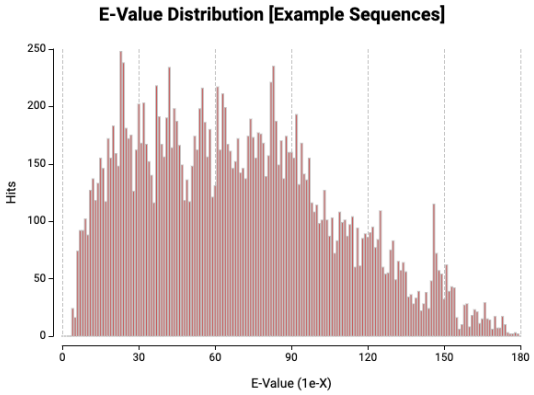


Figure 17: *E-value Distribution*

Blast Descripton Annotator

This will run the BDA algorithm. It also allows recovering the original description: When this option is marked the sequence description column on the Main Sequence Table will contain the top blast hit description and not the one from the BDA. For further details, please see the Blast Configuration Page section.

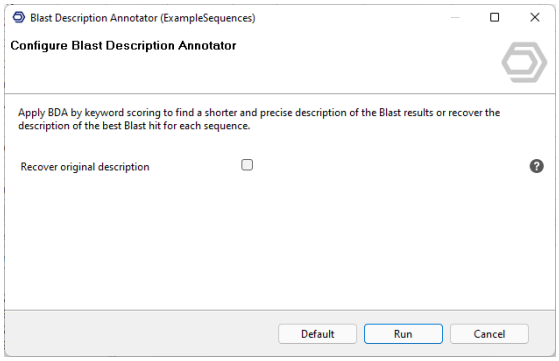


Figure 18: *Blast Description Annotator*

Retrieve Blast Top-Hit

This feature allows retrieving the sequence information of Top Blast Hits from an OmicsBox project to improve the annotation of a dataset.
A possible use case scenario would be a so-called "Double-Blast": The blast results of a first-run are used to replace the sequence data for a second run against a different set of query sequences. Imagine an RNA-seq data-set with a high percentage of sequences without any alignments against a protein database (e.g. blastx against NR). This feature could be used to select and extract the sequences without hits (red ones) into a new project. These sequences could be basted first against a set of EST sequences. The initial unaligned sequences are now replaced with the ESTs. Now the initial blastx search is repeated against the protein database.

It can be found on the side panel **Blast → Retrieve Blast Top-Hit.**

Configuration

Data can be obtained from the NCBI, Ensembl or Uniprot web services and stored in a new project or replace the existing IDs/sequences (figure 19).

- **Action:** Allows to either replace the sequence from the data set or extract them into a new data set.
- **Sequence Name:** It is possible to keep the original sequence names or to rename them to the names in the FASTA file. The latter will add a small note to the sequence description, telling the original name.
- **Replace Query With Top-Hit:** If checked the original sequence will be replaced by the one with a similar sequence found in the fasta file. This option is activated by default.
- **Filters Applied to Top-Hit:** For each Top-Hit (first significant alignment from an already performed BLAST), apply the filters (bottom part of the dialog) and search them in the corresponding database (online).

Depending on the configuration a new project will be generated or the current one will be changed.

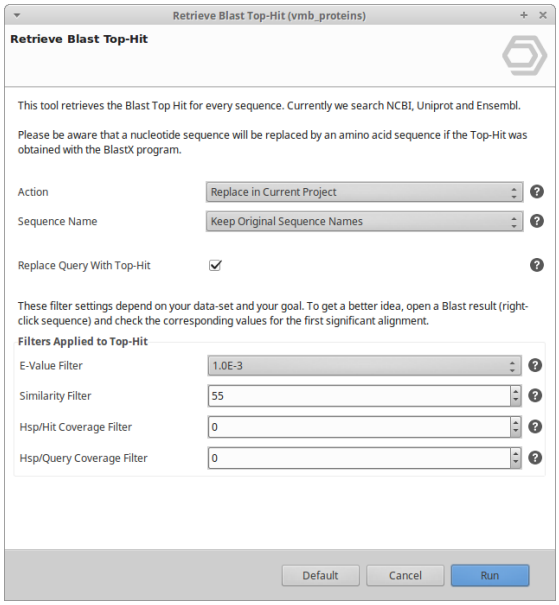


Figure 19: *Retrieve Blast Top-Hit Dialog*

Retrieve Blat Top-Hit

The BLAT algorithm is short for "BLAST-like alignment tool." BLAT is similar in many ways to BLAST. The program rapidly scans for relatively short matches (hits), and extends these into high-scoring pairs (HSPs). BLAT builds an index of the database and then scans linearly through the query sequence. In addition, BLAT can trigger extensions on any number of perfect or near-perfect hits. Furthermore, BLAT has a special code to handle introns in RNA/DNA alignments.

Please cite BLAT: Kent WJ. BLAT--the BLAST-like alignment tool. *Genome Res.* 2002;12(4):656-664. doi:10.1101/gr.229202

In OmicsBox BLAT is used to replace query sequences of a dataset with the top-hit one found in a reference FASTA file. It can be found on the side panel **Blast → Retrieve Blat Top-Hit.**

Configuration

This tool creates a BLAT database with a reference FASTA file and then finds a similar sequence in the project.
The following parameters can be configured.

- **Action:** Allows to either replace the sequence from the data set or extract them into a new data set.
- **Sequence Name:** It is possible to keep the original sequence names or to rename them to the names in the FASTA file. The latter will add a small note to the sequence description, telling the original name.
- **Replace Query With Top-Hit:** If checked the original sequence will be replaced by the one with a similar sequence found in the fasta file. This option is activated by default.
- **Reference Fasta:** BLAT needs a reference FASTA file which is used to search for similar sequences.
- **Similarity:** Filter by similarity
- **Check for Reverse Strand:** If checked BLAT will also consider the reverse strand to find similar sequences.

Depending on the configuration a new project will be generated or the current one will be changed.

A possible use case scenario would be a so-called "Double-Blast": The blast results of a first run are used to replace the sequence data for a second run against a different set of query sequences.
This tool can be useful after running Prokaryotic Gene Finding, in order to replace the sequence names retrieved from Glimmer with the top-hit from a reference fasta.

Visit the online tutorial here to see how to replace the sequence names.

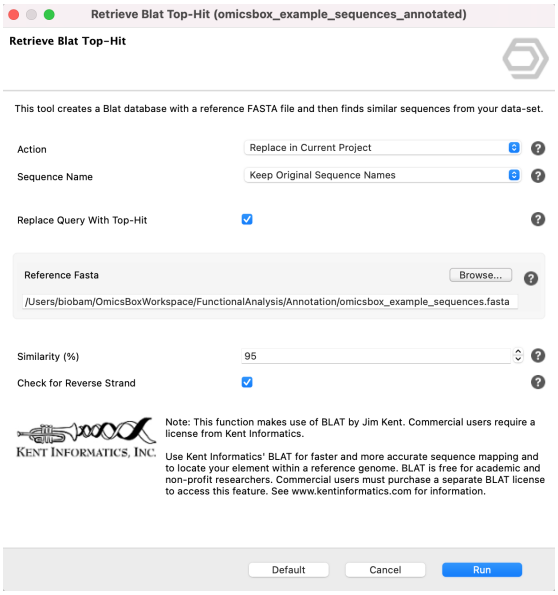


Figure 20: *Retrieve Blat Top-Hit Wizard*

Export Blast Top-Hits

A tab separator text file can be exported with the Blast top hit of each sequence (**Side Panel → Export → Export Blast Top-Hits**).

Create Blast Database

INTRODUCTION

This page describes how to create a BLAST database to be used with local BLAST available from the BLAST dialog in OmicsBox. This database can not be used in combination with Diamond or CloudBlast.

The makeblastdb application, provided by NCBI, produces BLAST databases from FASTA files. It is possible to use completely unstructured FASTA files, but this is not the recommended procedure. Assigning a unique identifier to every sequence in the database allows retrieving the sequence by identifier and also associating every sequence with a taxonomic node (through the taxid of the sequence).

Please cite Make Blast Database: https://www.ncbi.nlm.nih.gov/books/NBK569841/

OmicsBox offers the possibility to build a Blast database with your own sequences.

CREATE BLAST DATABASE

This functionality can be found under **functional analysis → Blast Search → Create Blast Database.** This option allows the creation of a BLAST database from the sequence of any OmicsBox project or from a FASTA file (figure 1).

Configuration

The Make Blast Database Configuration require the following parameters.

- Current project: OmicsBox will use the loaded sequences to create the Blast database. Note: If the resulting database will be used for further GO mapping a proper ID and description line with "GO mappable" information are needed.
- FASTA file: This option allows choosing own FASTA file. The FASTA file has to be correctly formatted for NCBI Blast+.
- Output Folder: Select the directory where to save the created Blast database.
- Blast Database Name: Provide a name for the Blast database
- Taxonomy Options:
- Taxonomy ID: Introduce the NCBI species ID
- Mapping file: If the sequences come from different species, it is possible to generate a text file with the sequence names and its species id to map to the corresponding sequence in the FASTA file.
- Example:

```
TR|A0A022PMT6|ERYGU 4155
TR|A0A022PMU0|ERYGU 4155
TR|A0A059BJ72|EUCGR 71139
TR|A0A059BJ72|EUCGR 71139
TR|A0A061FDU3|THECC 3641
TR|A0A067DJ79|CITSI 2711
```

Visit the following tutorial for more information on how to create the Taxonomy ID file.

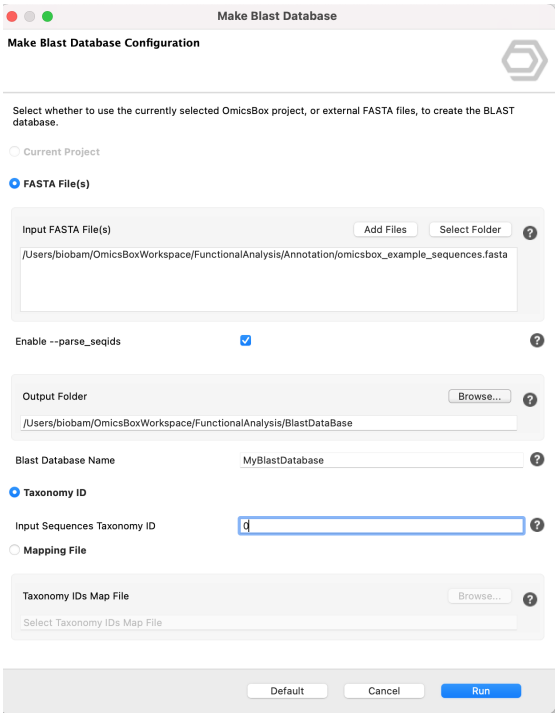


Figure 1: *Make Blast Database*

RESULTS

Once the execution has finished the Blast database folder contains all the files needed to run Local Blast.

4.5.4 InterProScan

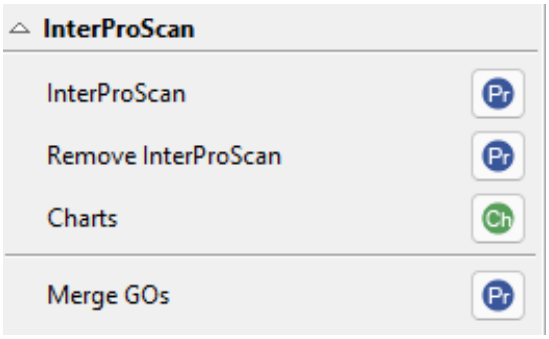
Introduction

InterPro provides functional analysis of proteins by classifying them into families and predicting domains and important sites. To classify proteins in this way, InterPro uses predictive models, known as signatures, provided by several different databases (referred to as member databases) that make up the InterPro consortium. InterPro combines protein signatures from these member databases into a single searchable resource, capitalising on their individual strengths to produce a powerful integrated database and diagnostic tool.

Please Cite InterProScan:

Blum M, Chang H, Chuguransky S, Grego T, Kandasaamy S, Mitchell A, Nuka G, Paysan-Lafosse T, Qureshi M, Raj S, RichardsonL, Salazar GA, Williams L, Bork P, Bridge A, Gough J, Haft DH, Letunic I, Marchler-Bauer A, Mi H, Natale DA, Necci M, Orengo CA, Pandurangan AP, Rivoire C, Sigrist CJA, Sillitoe I, Thanki N, Thomas PD, Tosatto SCE, Wu CH, Bateman A and Finn RD The InterPro protein families and domains database: 20 years on. Nucleic Acids Research, Nov 2020, (doi: 10.1093/nar/gkaa977)

The functionality of InterPro annotations in OmicsBox allows to retrieved domain/motif information in a sequence-wise manner. Corresponding GO terms are then transferred to the sequences and merged with already existing GO terms. InterProScan results can be viewed through the Single Sequence Menu (right-click on a sequence) and saved in TXT and XML format (figure 4). When working with nucleotide sequences, OmicsBox translates them to the longest open reading frame and then sends them to InterProScan.



InterProScan options

InterProScan

The following options can be found under **functional analysis → InterProScan** or from the **Side Panel** when a project has been loaded.

- **InteProScan.** Start sending sequences to the EBI or OmicsBox Cloud.
- **Remove InterProScan.** Delete InterProScan results for the selected sequences.
- **Charts.**
- **Merge GOs.** Add GO terms obtained through motifs/domains to the current annotations.

There are two options to run InterProScan in OmicsBox, either with CloudIPS or via the public web service at EBI.

CloudIPS is a cloud-based OmicsBox community resource for fast and reliable InterPro analysis for everything from small to big data sets. It allows executing the original InterPro algorithms against up-to-date databases in our dedicated computing cloud. This is a high-performance, secure and cost-optimized solution for your analysis.
The public **EMBL-EBI InterPro** web service scans your sequences against InterPro's signatures and performance and results depend on the EBI web server.

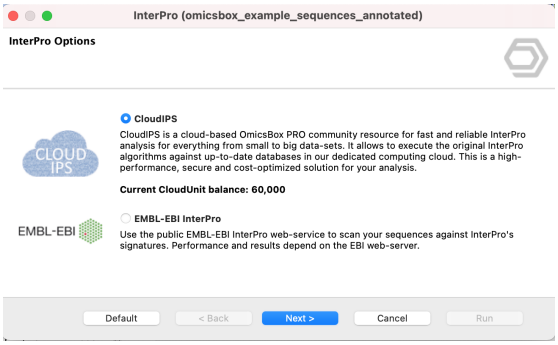


Figure 1: *Choose InterProScan*

CONFIGURATION

The first two configuration pages (figure2 and 3) show the databases that will be used to retrieve the protein families, domains, etc.

The last page allows to save the InterProScan results in different file formats, XML, and GFF3 which are the default outputs,in tab-separated values (TVS), and the input (query) sequence itself (figure 4).

Once the InterProScan has finished it is possible to view the results of each sequence via the context menu (figure 5). The sequences will turn **violet** if no other analysis has been executed before.

InteProScan can only be performed if the sequences are shown in the sequence table that contains the actual sequence information (loaded via fasta file). You have to be careful if you created a project via a blast XML file or if you loaded a .annot file.
To add the sequences to the current OmicsBox project see Add sequences to existing OmicsBox project section.

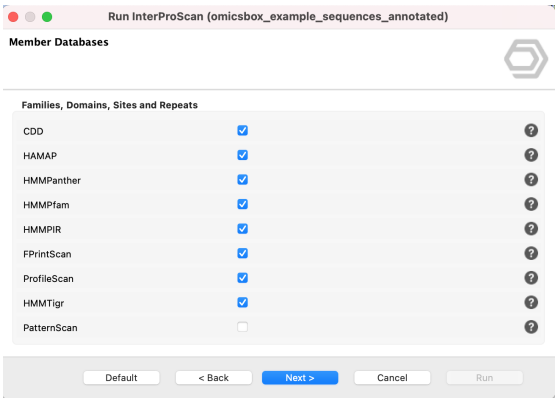


Figure 2: *Selection of Member Databases*

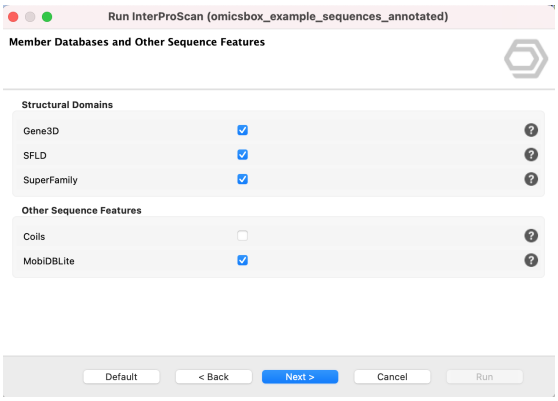


Figure 3: *Selection of Member Databases*

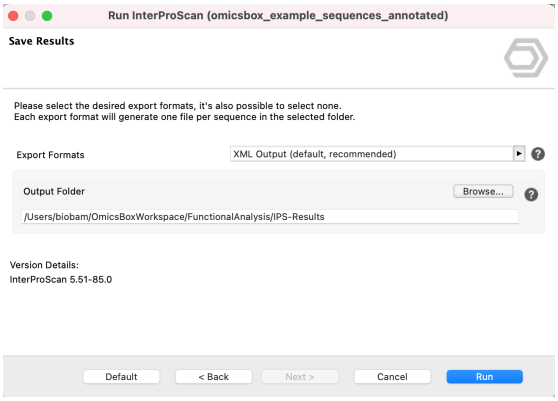


Figure 4: *Save InterProScan Results*

Figure 5: *InterProScan Results*

Charts

It is possible to select InterProScan statistics to see how many sequences still do or do not have IPS results and how many sequences have GOs resulting from InterProScan.

- **InterProScan Results:** This chart reflects the effect of adding the GO terms retrieved through the InterProScan results (figure 7). When comparing this chart with the chart in figure 4 "Analysis Progress" the bar "Only with InterProScan" includes the number of sequences "With and Without IPS" in figure 7.
- **InterProScan Families Distribution:** Bar chart representing the number of sequences that belong to a particular IPS family.
- **InterProScan Domains Distribution:** Bar chart showing the number of sequences that belong to a particular IPS domain.
- **InterProScan Repeats Distribution:** Bar chart reflecting the number of sequences that belong to a particular IPS repeat.
- **InterProScan Sites Distribution:** Bar chart representing the number of sequences that belong to a particular IPS site.
- **InterProScan IDs Distribution:** Bar chart showing the number of sequences that have been annotated with that InterProScan IDs.
- **InterProScan IDs by Database:** Pie chart reflecting the number of sequences of the InterProScan IDs for a particular InterProScan Database. In figure 6 the Pfam database is selected.

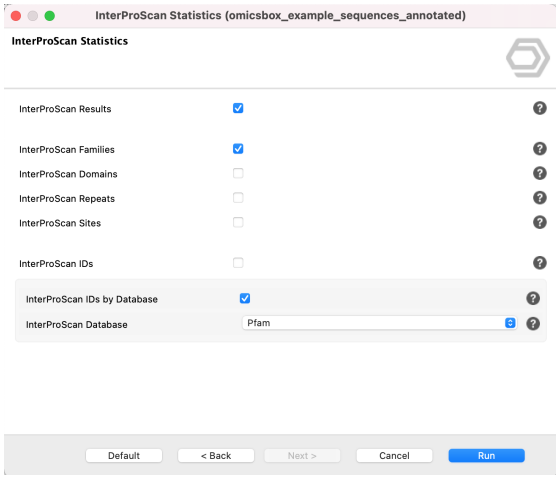


Figure 6: *InterProScan Statistics Configuration Window*

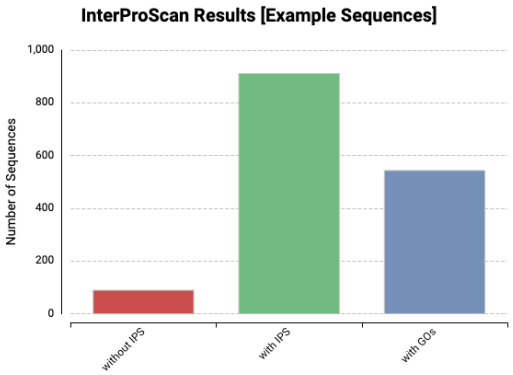


Figure 7: *InterProScan Statistics*

Merge GOs

The InterProScan GOs results can now be added to the already existing annotations based on the BLAST results. This option is available from the InterProScan submenu.
Once the merge has finished a distribution chart is displayed in the Results menu showing the number of GOs that have been added to (or confirmed) the current annotation results.

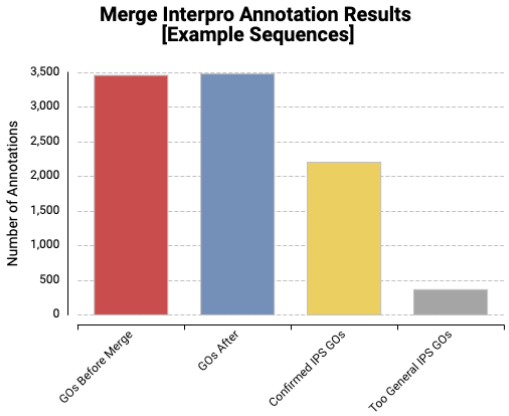


Figure 8: *Statistics after merging InterProScan to GO Annotation*

4.5.5 Gene Ontology Mapping

Introduction

Mapping is the process of retrieving GO terms associated with the Hits obtained by the BLAST search. OmicsBox performs four different mappings steps:

1. BLAST result accessions are used to retrieve gene names or symbols making use of two mapping files provided by the NCBI (gene_info, gene2accession). Identified gene names are then searched in the species-specific entries of the gene-product table of the GO database.
2. GeneBank identifiers (gi), the primary blast Hit ids, are used to retrieve UniProt IDs making use of a mapping file from PIR (Non-redundant Reference Protein Database) including PSD, UniProt, Swiss-Prot, TrEMBL, RefSeq, GenPept and PDB.
3. Accessions are searched directly in the dbxref table of the GO database.
4. BLAST result accessions are searched directly in the gene-product table of the GO database.

Please cite:

Gotz S., Garcia-Gomez JM., Terol J., Williams TD., Nagaraj SH., Nueda MJ., Robles M., Talon M., Dopazo J. and Conesa A. (2008). High-throughput functional annotation and data mining with the Blast2GO suite. Nucleic acids research, 36(10), 3420-35.

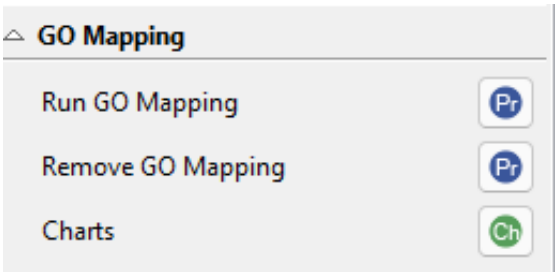


Figure 1:*Mapping options*

Run GO Mapping

The GO Mapping functionality can be found in the **Side Panel → GO Mapping** after loading a fasta file.

The GO Mapping options are:

- **GO Mapping**. Mapping will start using online mapping database.
- **Remove Mapping**. Delete Mapping results for the selected sequences.
- **Charts.** Generate Mapping statistics charts.

The mapping step needs protein ids to run. Make sure you ran blast against a protein database.

blastx - if one has nucleotide sequences

blastp - if one has protein sequences

Results

SHOW MAPPING RESULTS

For each sequence, it is possible to see the mapping results individually.

1. Show Mapping Results. A new table will be displayed (see figure 3). The resulting table shows the GO mapping results for a particular sequence. See Table section to manipulate/extract the results from this table.
2. Show GO Descriptions. GO ID, description, type, and definition are given for all GO terms associated with the selected sequence. The GO ID is linked to the AmiGO browser at the Gene Ontology site while the show option displays the DAG representation of the GO term.
3. Annotate Sequence. This function allows changing annotation parameters for the selected sequence and re-running automatic annotation.
4. Change Annotation and Description. This function edits the annotation of the selected and allows typing and deleting of annotation or sequence description. A manual annotation check-box (see figure 5 in Gene Ontology Annotation section) is available for marking sequences with manual annotation. The sequence will get the pink label on the Main Sequence Table.
5. Make Graph of GO-Mapping-Results with Annotation Score. Displays a DAG with all GO terms related to one sequence. Shows all the GOs from the mapping step as well as final annotations (highlighted). The wizard (figure 4 allows filtering the hits which will be taken into account (see Gene Ontology Graphs section for more details about visualization in OmicsBox)
6. Hit Filter. Nodes can be filtered out by a number of hits: only nodes with more than a given number of BLAST-Hits will be shown in the graph.
7. HSP-Hit Coverage CutOff: Includes only those hits which are overage with the HSP for a given percentage.

For Mapping statistic charts see the Charts and Statistics page of this user manual.

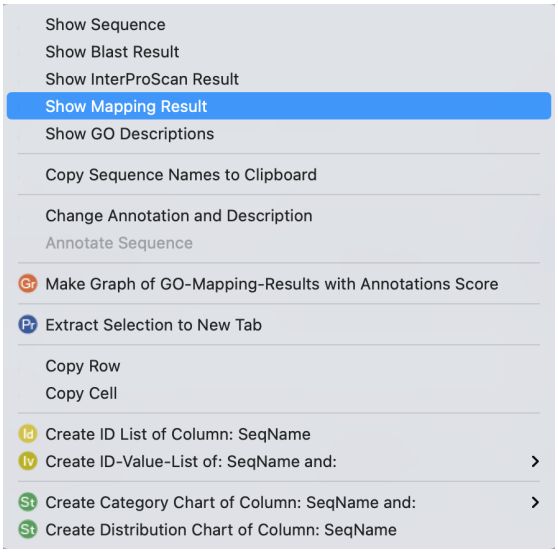


Figure 2: *Show Mapping Results*

Figure 3: *Mapping Results for sequence C02006A02*

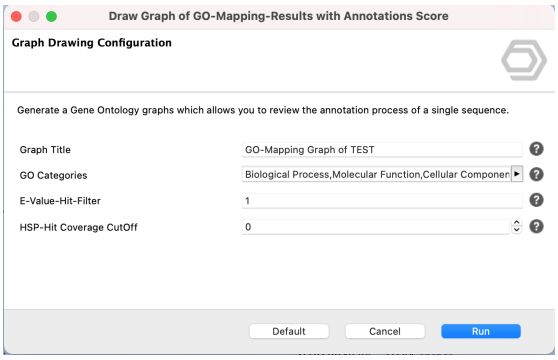


Figure 4: *Single Graph Drawing Configuration*

CHARTS

Three different charts are available to summarise the mapping step:

- **GO Mapping Distribution:** This shows the distribution of the number of Gene Ontology candidate terms assigned to each sequence during the GO Mapping step.
- **EC Distribution for Sequences:** This chart shows the distribution of GO evidence codes for the functional terms obtained during the mapping step. It gives an idea about how many annotations derive from automatic/computational annotations or manually curated ones.
- **EC Distribution for Blast Hits:** Evidence Codes associated with the obtained GO pool.

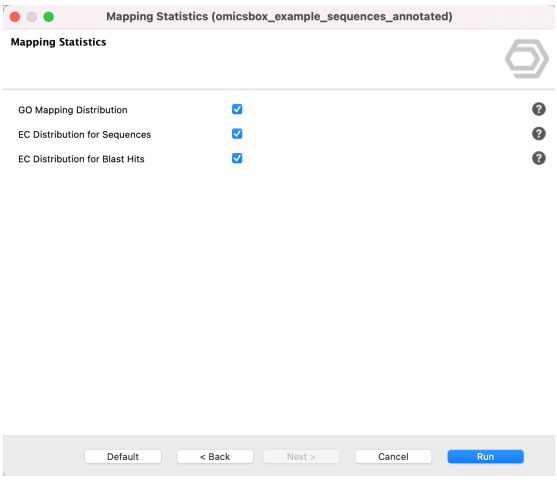


Figure 5: *Mapping Statistics*

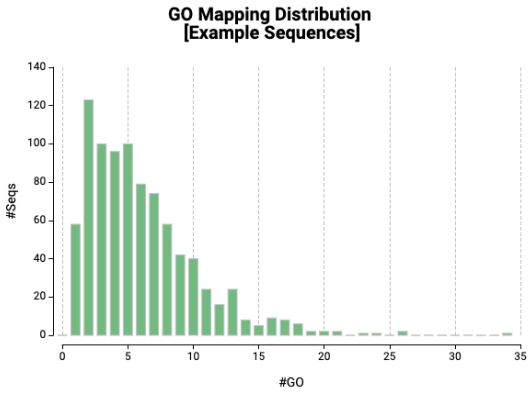
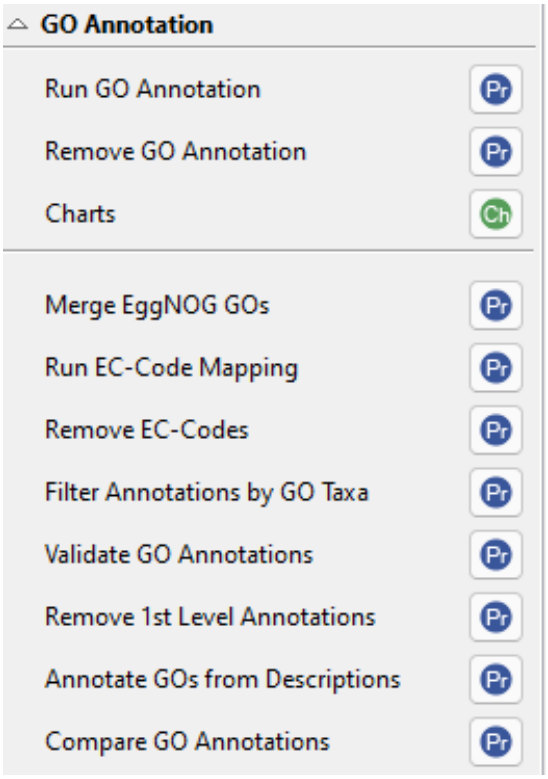


Figure6: *GO Mapping Distribution*

EXPORT MAPPING RESULTS

A tab separator text file can be exported with the corresponding mapping results (**Side Panel > Export > Export Mapping Results**).

4.5.6 Gene Ontology Annotation



GO Annotation options

Introduction

Annotation rule is the process of selecting GO terms from the GO pool obtained by the Mapping step and assigning them to the query sequences. In the current OmicsBox version, this is the core type of functional annotation.

GO annotation is carried out by applying an annotation rule (AR) on the found ontology terms. The rule seeks to find the most specific annotations with a certain level of reliability. This process is adjustable in specificity and stringency.

For each candidate GO an annotation score (AS) is computed. The AS is composed of two additive terms.

The first, direct term (DT), represents the highest hit similarity of this GO weighted by a factor corresponding to its EC.

The second term (AT) of the AS provides the possibility of abstraction. This is defined as an annotation to a parent node when several child nodes are present in the GO candidate collection. This term multiplies the number of total GOs unified at the node by a user-defined GO weight factor that controls the possibility and strength of abstraction. When GO weight is set to 0, no abstraction is done.

Finally, the AR selects the lowest term per branch that lies over a user-defined threshold. DT, AT, and the AR terms are defined as given in figure 1.

To better understand how the annotation score works, the following reasoning can be done: When EC-weight is set to 1 for all ECs (no EC influence) and GO-weight equals zero (no abstraction), then the annotation score equals the maximum similarity value of the hits that have that GO term and the sequence will be annotated with that GO term if that score is above the given threshold provided. The situation when EC-weights are lower than 1 means that higher similarities are required to reach the threshold. If the GO-weight is different to 0 this means that the possibility is enabled that a parent node will reach the threshold while its various children nodes would not.

The annotation rule provides a general framework for annotation. The actual way annotation occurs depends on how the different parameters at the AS are set. These can be adjusted in the Annotation Configuration Dialog (figure 2) and in the Evidence Code Weight Configuration Dialog (figure 3).

Please cite:

Gotz S., Garcia-Gomez JM., Terol J., Williams TD., Nagaraj SH., Nueda MJ., Robles M., Talon M., Dopazo J. and Conesa A. (2008). High-throughput functional annotation and data mining with the Blast2GO suite. Nucleic acids research, 36(10), 3420-35.

$$DT = \max(similarity \times EC_{weight})$$
$$AT = (\#GO - 1) \times GO_{weight}$$
$$AR : lowest.node(AS(DT + AT)) \geq threshold$$

Figure 1: *OmicsBox Annotation Rule*

Run GO Annotation

The GO Annotation functionality can be found under the **Side Panel → GO Annotation** after loading a fasta file or a project**.**

ANNOTATION CONFIGURATION

- **Annotation Cut-Off (threshold):** The annotation rule selects the lowest term per branch that lies over this threshold (default=55).
- **GO-Weight:** This is the weight given to the contribution of mapped children terms to the annotation of a parent term (default=5).
- **Filter GO by taxonomy:** The filter will remove the Gene Ontology terms known not to be in the given taxonomy using the restrictions defined by Gene Ontology. You can select one of the given options or simply write a taxonomy id.
- **E-Value-Hit-Filter:** This value can be understood as a pre-filter: only GO terms obtained from hits with a greater e-value than given will be used for annotation and/or shown in a generated graph (default=1.0E-6). The value to use will depend on how restrictive or permissive the annotation should be.
- **Hsp-HitCoverage CutOff:** Sets the minimum needed coverage between a Hit and his HSP. For example, a value of 80 would mean that the aligned HSP must cover at least 80% of the longitude of its Hit. Only annotations from Hit fulfilling this criterion will be considered for annotation transference.
- **Hit Filter:** This option allows you to consider only the first N hits during annotation. This option is correlative with the "Only hits with GOs" feature.
- **Only hits with GOs:** This option together with the "Hit Filter" option allows you to apply it only on hits that have a GO term candidate.

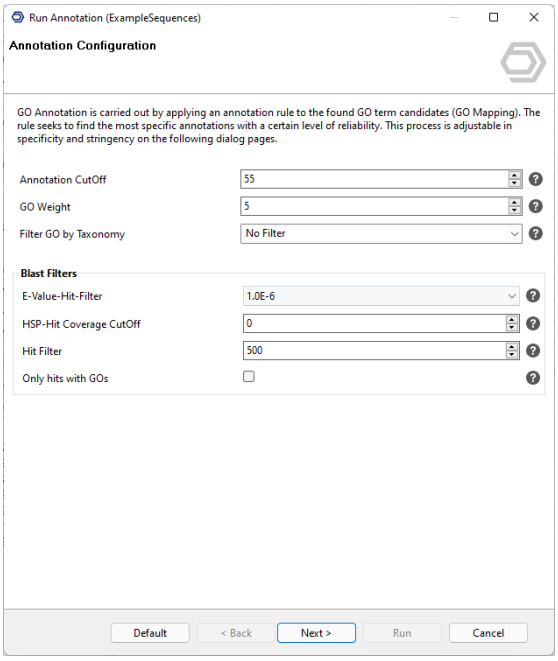


Figure 2:*Annotation Configuration*

EVIDENCE CODE WEIGHTS

Employing ECs promotes the assignment of annotations with experimental evidence and penalizes electronic annotations or low traceability.

EC code weights can be modified depending on what you want. Note that in case of influence by evidence codes is not wanted, you can set them all at 1. Alternatively, when you want to exclude GO annotations of a certain EC (for example IEAs), you can set this EC weight at 0.

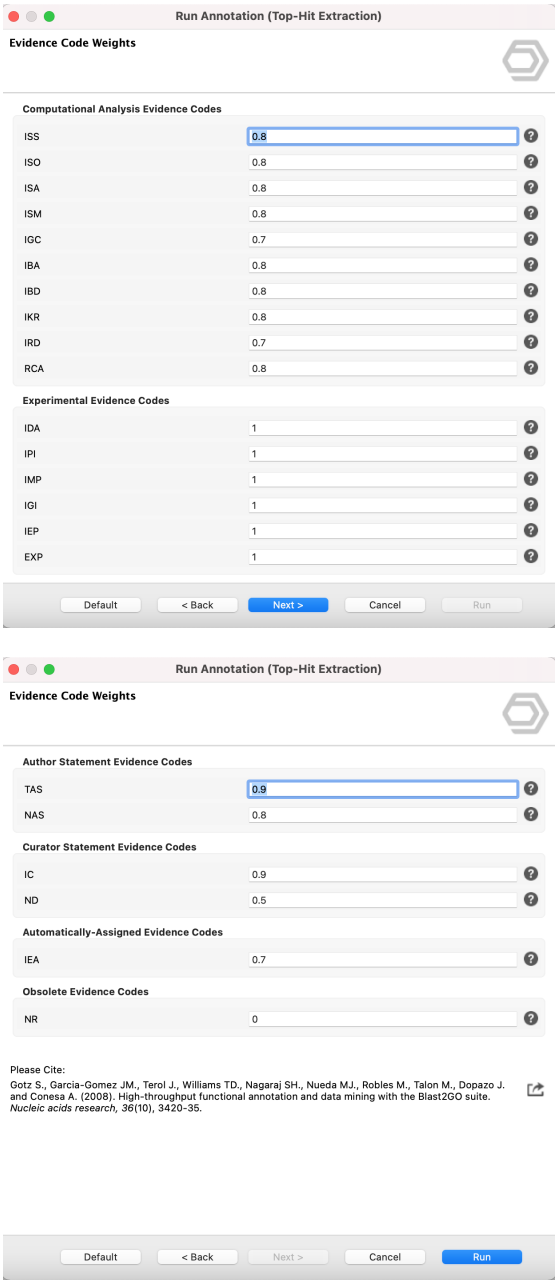


Figure 3: *Evidence Code weight configuration*

Results

Successful annotation for each query sequence will result in a color change for that sequence from light-green to **blue** at the Main Sequence Table, and only the annotated GOs will remain in the GO IDs column.

RESULT TABLE

- **Selection CheckBox:** This checkbox can be used to filter or select the table and to apply actions (extract data, generate charts) only to the selected part of the table.
- **Nr:** A consecutive number for each row.
- **Tags:** Depending on the status of a given sequence the row will show different tags like BLASTED, INTERPRO, MAPPED, ANNOTATED or GOSLIM .
- **SeqName:** The unique name of the sequence. Duplicates are not allowed.
- **Description:** The description line of a sequence. This description will be imported from the fasta file and can be overwritten during the annotation process or manually.

- **Length:** The length of the sequences in bases. This can be amino-acids or nucleotides depending on the type of the sequence.
- **#Hits (blast related):** The number of hits obtained by blast.
- **e-Value (blast related):** The lowest e-value obtained by blast.
- **sim-mean (blast related):** The mean similarity obtained by blast.
- **#GO (mapping and annotation related):** The number of gene ontology terms obtained during the mapping or annotation process.
- **GO IDs (mapping and annotation related):** The gene ontology IDs obtained during the mapping or annotation process.
- **GO Names (mapping and annotation related):** The gene ontology names obtained during the mapping or annotation process.
- **Enzyme Codes (annotation related):** The enzyme codes linked to the GO terms of a given sequence
- **Enzyme Names (annotation related):** The enzyme code names linked to the GO terms of a given sequence
- **InterPro IDs (interproscan related):** The IDs obtained during the InterProScan step.
- **InterPro GO IDs (interproscan related):** The GO IDs linked to the InterPro IDs obtained during the InterProScan step.
- **InterPro GO Names (interproscan related):** The GO names linked to the InterPro IDs obtained during the InterProScan step.

INDIVIDUAL ANNOTATION RESULTS

Annotation results for each sequence can also be visualized on the GO DAG by selecting "Draw Graph of GO-Mapping with Annotation Score" in the context menu. Additionally, the "Change Annotation and Description" figure 4 options of this menu offer also the possibility to adjust annotations specifically for a single sequence. This function edits the annotation of the selected and allows typing and deleting of annotation or sequence description. A manual annotation check-box (see figure 5) is available for marking sequences with manual annotation. The sequence will get the **pink** label on the Main Sequence Table.

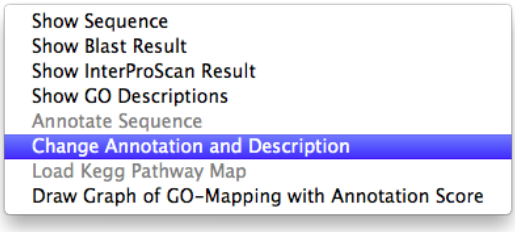


Figure 4: *Manually change Annotation and Description*

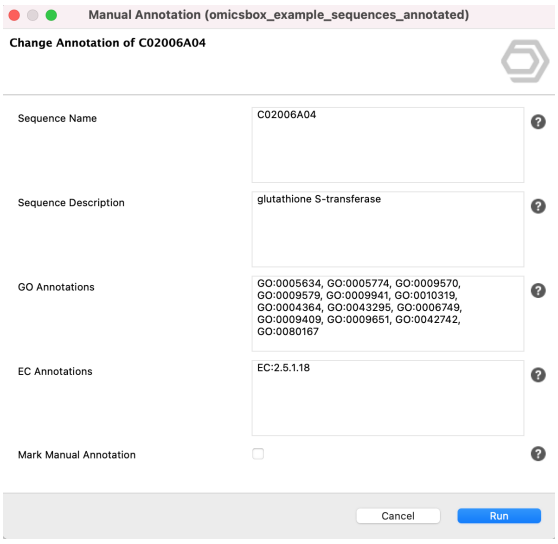


Figure 5: *Mark Manual Annotation*

CHARTS

GO Annotation Statistics

It is possible to summarise the number of sequences that have been annotated with the annotation rule and the following statistics are available:

- **Annotation Distribution:** This chart informs about the number of GO terms assigned per sequence.
- **GO Annotation Level Distribution:** A bar chart that shows all GO terms for all 3 categories for a given GO level taking into account the GO hierarchy (parent-child relationships).
- **Annotation Score Distribution:** A chart that shows the number of sequences per annotation score.
- **Annotated Seqs/Seq-Length:** This shows the relation between the amount of annotated sequences and sequence lengths.
- **Number of GOs/Seq-Length:** This shows the relation between sequence length and the number of GOs.
- **GO Distribution by Level:** A bar chart that shows all the GO terms for all 3 categories for GO level 2, taking into account the GO hierarchy.
- Direct GO Count:
- **Molecular Function:** A chart for the Molecular Function GO category, which shows the most frequent GO terms within a data-set without taking into account the GO hierarchy.
- **Biological Process:** A chart for the Biological Process GO category.
- **Cellular Component:** A chart for the Cellular Component GO category.

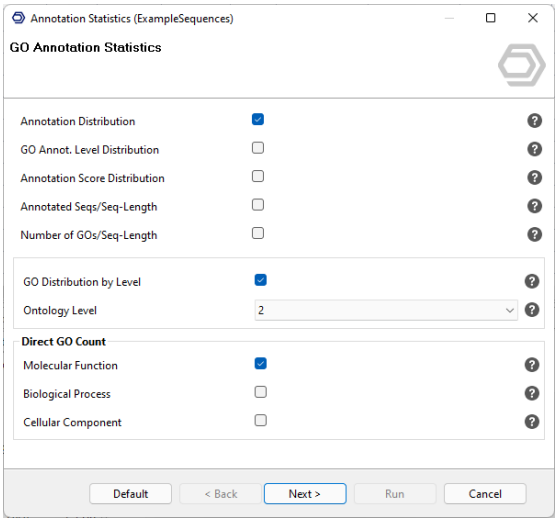


Figure 6: *Annotation Statistics*

An overview of the extent and intensity of the annotation can be obtained from the Annotation Distribution Chart (figure 7), which shows the number of sequences annotated with different amounts of GO-terms.

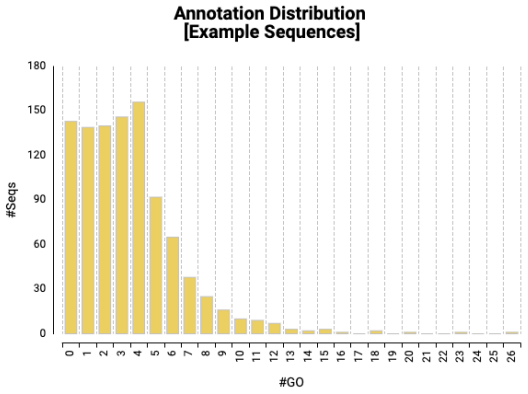


Figure 7: *Annotation Distribution*

EC- Code Statistics

To see the main Enzyme classes in the dataset it is possible to generate a distribution Enzyme Code chart.

- **Main Enzyme Classes:** This shows the distribution of the 7 main enzyme classes' overall sequences.
- **Second Level Classes:** It is possible to create a distribution chart of the enzyme subclasses.

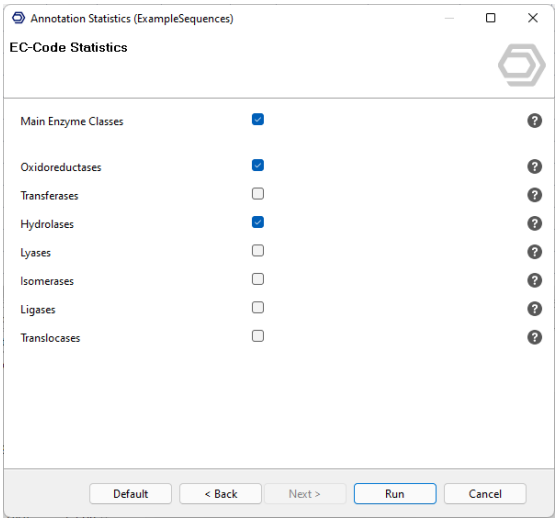


Figure 8:*Enzyme Code Statistic*

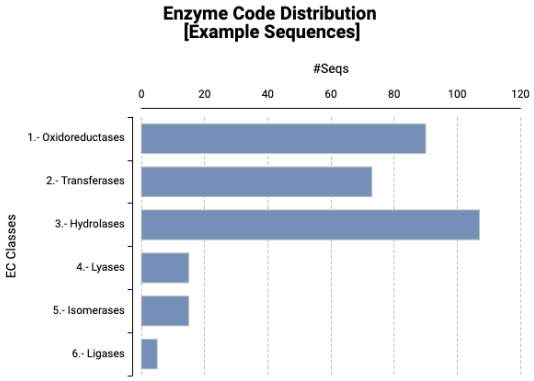


Figure 9: *Enzyme Code Distribution*

EXPORT ANNOTATION RESULTS

The annotation results can be exported in a variety of formats. This function is available under **Side Panel→ Export → Export GO Annotations**.

1. .annot. This is the default option for Annotation export and the exchange annotation format in OmicsBox. Annotations are provided in a three-column fashion. The first column contains the sequence name, the second the annotation code and the third the sequence description. When multiple annotations for the same sequence are available, these come in subsequent rows. GO and EC annotations are exported jointly in the same format.
2. Custom: It is possible to customize the exportation of the annotation file according to the information desired or the column separator see the next figure.
3. Genespring format. One single row is given by sequence where three different columns are provided for Molecular Function, Biological Process, and Cellular Component. GO terms are denoted by their description rather than by their code.
4. GoStats format. One single row is given by sequence and GO terms are only denoted by entire numbers ("GO:" and left zero's are skipped)
5. WEGO format (native). One single row is given by sequence, including those without annotated GOs. Belonging GOs are added to each sequence separated by tabs. The format corresponds to the "WEGO native format", shown in this example: http://wego.genomics.org.cn/docs/input01.lst.
6. Export Annotations in GO Annotation File Format (GAF v.2), which is the primary format currently used by the GO Consortiumhttp://geneontology.org/page/go-annotation-file-formats.
7. Export GO Propagation: Exports the GO parents up to the root for the annotated sequences.
8. Export Sequences per GO (Gene Sets).

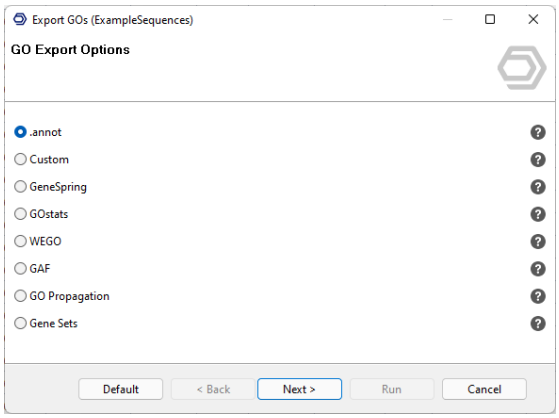


Figure 10: *Export Annotation Configuration*

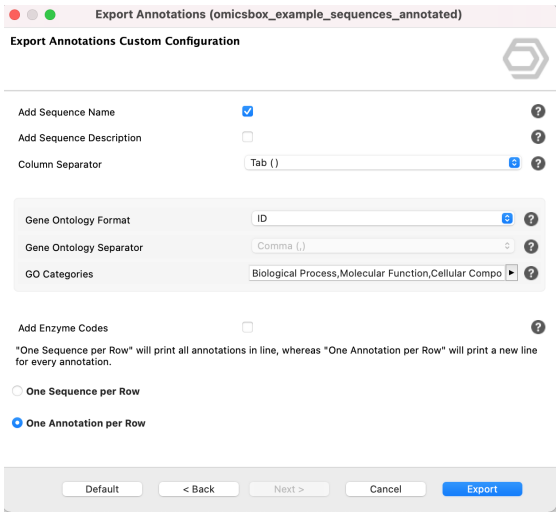


Figure 11: *Export Annotations Custom Configuration*

REMOVE GO ANNOTATION

Delete Annotation results for the selected sequences.

Merge EggNOG GOs

Once the sequences are annotated via EggNOG, it is possible to merge the GO terms and the EC codes (Enzyme Commission Codes) to a sequence project in order to add the new annotations. This can be done by clicking on project **Side Panel → GO Annotation → Merge EggNOG GOs** (figure 12).

In the wizard, you have to select the EggNOG project that has the GO annotations to merge with the current project. If the sequences already have annotated GO terms and/or ECs, the new information generated from EggNOG will be added to the annotations found in the project.

In addition, you can filter the annotations by E-value or Bit-Score.

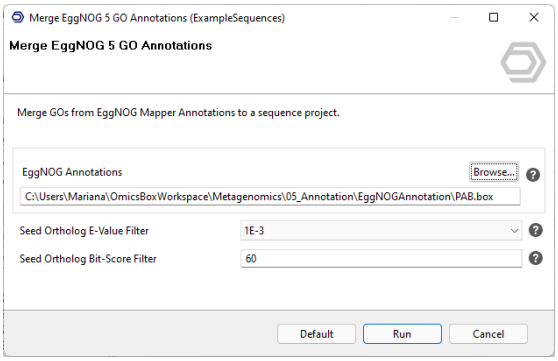


Figure 12: *Merge EggNOG GO Annotations wizard.*

Once finished, this step generates a bar chart showing the total number of GOs and ECs added to the original sequence project (figure 13).

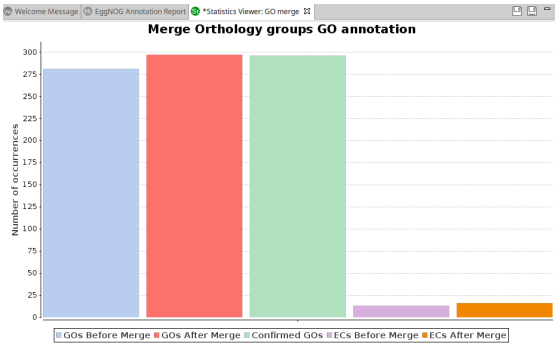


Figure 13: *Merge EggNOG GO Annotations graph.*

Annotate GOs from Descriptions

This tool looks at every significant alignment (**Right-Click → Show Blast Result** on a sequence) for each sequence and searches their description lines for GO ids. These GOs are now directly annotated to the sequence if the alignments similarity passes the desired minimum. Validation can also be applied and is recommended, it will remove intermediate GO terms.

There are still other annotation functions available in the submenu:

Other Annotation Functions in the Side Panel

- Run EC-Code Mapping: This will map GO annotations to EC-Codes for fully annotated sequences. The mapping data is provided by the Gene Ontology Consortium.
- Remove EC-Codes: This will remove the Enzyme Codes from the project.
- Filter Annotation by GO Taxa
- Validate Annotations. OmicsBox annotation generates the lowest node annotations. This is not always guaranteed when Annotations have been imported or changed manually. This function can be run to ensure that no parent-child redundancy is present in the annotated set.
- Remove 1. Level Annotations
- Annotate GOs from Blast Descriptions allows to transfer of GOs from the Blast hit descriptions to their sequences.
- Compare GO Annotations: Compare a set of annotations for a given group of sequences against the annotations already loaded in OmicsBox.

4.5.7 Functional Annotation with EggNOG Mapper

Introduction

EggNOG-mapper is a tool for fast functional annotation of novel sequences (genes or proteins) using precomputed eggNOG-based orthology assignments. Obvious examples include the annotation of novel genomes, transcriptomes or even metagenomic gene catalogs. The use of orthology predictions for functional annotation is considered more precise than traditional homology searches, as it avoids transferring annotations from paralogs (duplicate genes with a higher chance of being involved in functional divergence).

Details and methodology about the tool and its database are best explained on their website: http://eggnogdb.embl.de/#/app/methods .

EggNOG-mapper can be found under **Functional Analysis → EggNOG Annotation → EggNOG Mapper**. The wizard allows to select the parameters for the functional annotation (figure 1).

Wizard Page

- **Target Orthologs**: Define what type of orthologs should be used for functional transfer.
- **GO Evidence**: Define what type of GO terms should be used for annotation:
- experimental = Use only terms inferred from experimental evidence.
- non-electronic = Use only non-electronically curated terms.

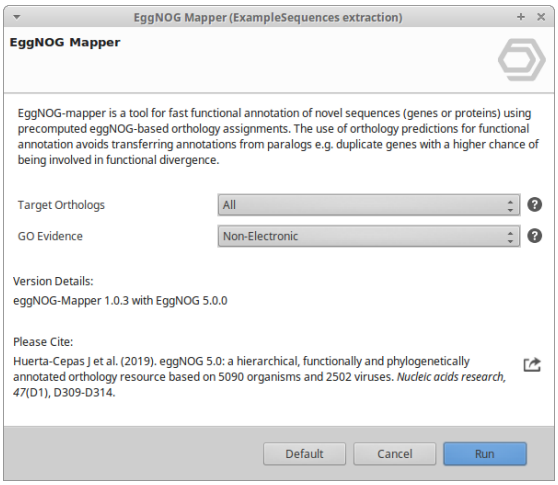


Figure 1. *EggNOG Mapper wizard page.*

Results

The **result table** summarizes all annotations that could be transferred with EggNOG Mapper. Besides ordering and filtering, the context menu allows taking a closer look at certain results (figure 2). This annotation process also generates a **Summary Report** with information about the total number of GOs, the COG categories and the orthologous groups distribution.

Figure 2. *EggNOG results table.*

The **annotation details** (right-click on an annotated sequence → **Show Annotation Details**) provide link outs where possible and give detailed information about annotated GOs (figure 3).

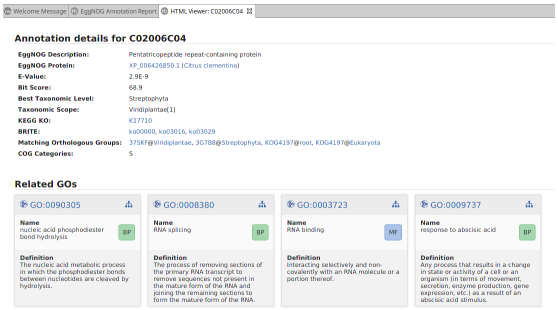


Figure 3. *Annotation details.*

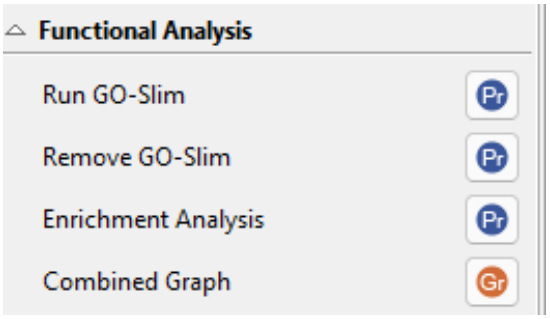
Reference

Fast genome-wide functional annotation through orthology assignment by eggNOG-mapper. Jaime Huerta-Cepas, Damian Szklarczyk, Lars Juhl Jensen, Christian von Mering and Peer Bork. Submitted (**2016**).

eggNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. Jaime Huerta-Cepas, Damian Szklarczyk, Kristoffer Forslund, Helen Cook, Davide Heller, Mathias C. Walter, Thomas Rattei, Daniel R. Mende, Shinichi Sunagawa, Michael Kuhn, Lars Juhl Jensen, Christian von Mering, and Peer Bork. Nucl. Acids Res. (04 January **2016**) 44 (D1): D286-D293. doi: 10.1093/nar/gkv1248

4.5.8 Functional Analysis (Side Panel)

Functional Analysis (Side Panel)



GO-Slim

INTRODUCTION

GO-Slim is a reduced version of the Gene Ontology that contains a selected number of relevant nodes.
GO slims are cut-down versions of the Gene Ontologies that contain a subset of the GO terms. GO slims summarise a set of GO annotations from a genome, microarray, or cDNA collection to a simpler functional schema.

OmicsBox offers the possibility to run GO slim on the data using existing files from the Gene Ontology webpage or provide customised GO Slim files.

For more information please visit Gene Ontology page.

RUN GO-SLIM

The GO Slim feature can be found under **Functional Analysis Side Panel → Run GO Slim.**

GO Slim Configuration

Different GO-Slims are available which are adapted to specific organisms. OmicsBox supports the following GO-Slim mappings: General, Plant, Yeast, GOA (GO-Association) and TAIR.

- **Obo File from GO-Website**
- **GO-Slim file:** Choose the GO slim to use. The file will be directly downloaded from the Gene Ontology webpage.
- **Custom Obo File**
- **GO Obo file:** Browse for the Obo file to use. It can be a custom obo file or downloaded from Gene Ontology.
- **Custom GO Slim file:** Browse for the GO slim file to use. It can be a custom GO slim file or downloaded from Gene Ontology.

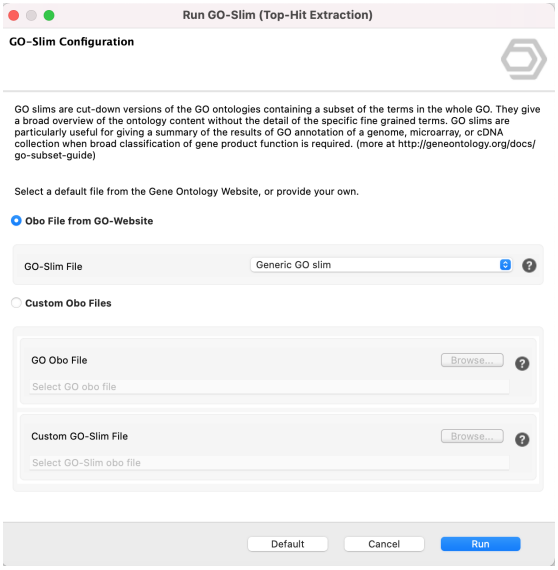


Figure 1: *GO Slim Configuration*

RESULTS

Successful GO Slim for each query sequence will result in a color change for that sequence from blue to **yellow** at the Main Sequence Table, and only the GO slim GOs will remain in the GO IDs column.

Use the **Remove GO-Slim** option to return to the original annotations.

Enrichment Analysis

ENRICHMENT ANALYSIS

Introduction

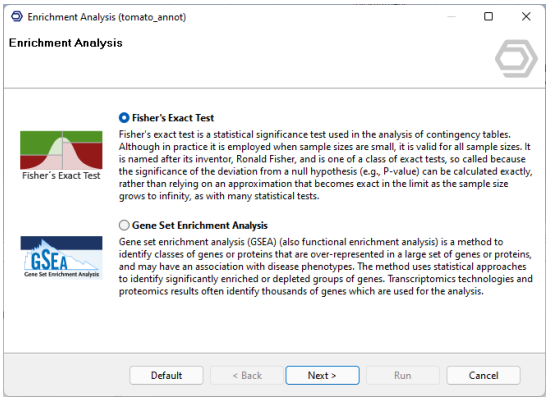
A functional enrichment analysis is the procedure of identifying functions that are over- or under-represented among a set of genes and may have an association with an experimental condition like for example phenotype or drug treatment. To obtain functional profiles for list of genes helps to gain a better understanding of the underlying biological processes. Enrichment analysis methods use statistical approaches to identify **significantly** enriched or depleted functions among a group of genes.

There are two main types of enrichment analysis:

- **Over Representation Analysis:** This method **compares the functional annotations of two lists** of genes against each other. It tests for each function, e.g. a GO term, if it is more frequent in one list compared to another list i.e. a reference set or background. This type of enrichment is calculated via a contingency table used in statistical tests like e.g. a **Fisher's Exact Test.**
- **Gene Set Enrichment Analysis (GSEA):** This method tests if genes of a gene set (i.e. a group of genes annotated with the same GO term) accumulate in the upper or lower part of **a ranked list of genes**. A gene list can be ranked by any metric with biological meaning e.g. expression values or methylation level, etc.

The Functional Analysis module in OmicsBox allows performing both types of tests. The input for each type of enrichment analysis is different. The Fisher's Exact Test requieres a list of IDs for the test-set sequences. On the other hand, the GSEA requires a ranked list of gene IDs where the first column is the sequence identifier and the second column (tab-separated) is the metric with biological meaning e.g the logFC.

Both types of lists can be generated from within OmicsBox or imported as a plain text file. For a detailed tutorial on how to prepare these lists please continue reading here.



Enrichment Analysis options

References

- Rivals I, Personnaz L, Taing L, Potier MC. Enrichment or depletion of a GO category within a class of genes: which test? *Bioinformatics*. 2007;23(4):401-407. doi: 10.1093/bioinformatics/btl633
- Subramanian A, Tamayo P, Mootha VK, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*. 2005;102(43):15545-15550. doi:10.1073/pnas.0506580102

FISHER EXACT TEST

Introduction

Fisher's Exact Test (FET) can be used to find GO terms, or other annotations, that are over and under-represented in a set of genes (test set) with respect to a reference group (reference set). This set of genes can be the differentially expressed genes of differential expression analysis, a set of genes related to a phenotype of interest, etc. Fisher's Exact Test uses a contingency table-based method to examine the association between two kinds of classification.

When the proportion of genes annotated with a determined GO term in the test set is significantly higher than the proportion in the reference set, this GO term will be detected as **over-represented**, and otherwise, it will be declared **under-represented**.

OmicsBox has integrated the FatiGO package for statistical assessment of annotation differences between 2 sets of sequences. This package uses Fisher's Exact Test and corrects for multiple testing. For this analysis, the completion (but not exclusively) of the involved sequences with their annotations must be loaded in the application. This can either be the result of a OmicsBox annotation or the imported annotation by file (*.annot*), see Gene Ontology Annotation of this manual.

Run Fisher's Exact Test

This functionality can be found as a side panel button in the following tables:

- Annotated sequences from Functional Analysis.
- Combined Pathway Analysis results.
- Pairwise Differential Expression results (with and without replicates).
- Time Course Differential Expression results.
- Single Cell RNA-Seq Differential Expression results.

If the FET analysis has been launched from an annotation project, Test and Reference Sequences can be selected by uploading text files or ID-List *.box* files containing the lists of sequence IDs for the two groups (Figure 1). When there is no reference set chosen, the whole dataset present in the project will be taken as Reference. A detailed description of each parameter is available by clicking the help icon next to the parameter.

If the Fisher's Exact Test is applied to differential expression results, the Test Set can be selected from the significant differential expressed features (genes/transcripts). Reference Set would be the rest of annotated features from the Reference Set file provided. See the specific manual section for a more detailed information (linked in the bullet points above).

The Fisher's Exact Test implementation is sensitive in the direction of the test: the sequences that are present in the test-set and also in reference-set will be deleted from the reference, but not from the test-set.

For further details please refer to the FatiGO publication: Al-Shahrour, F., Díaz-Uriarte, R., and Dopazo, J. (2004). Fatigo: a web tool for finding significant associations of gene ontology terms with groups of genes. Bioinformatics, 20(4):578–580.

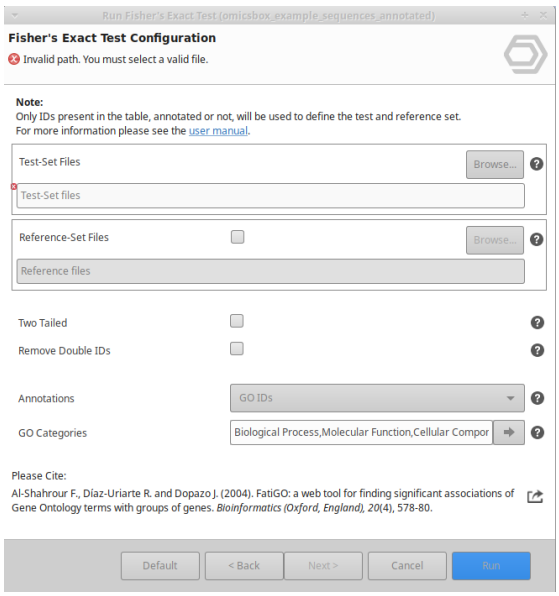


Figure 1: *Fisher's Exact Test configuration wizard*

Input parameters

- **Test-set Files.** ID-list with sequences belonging to the test-set (Annotation project).
- **Test-set Genes.** Subset of significant genes to be considered as the test set. It allows to pick between up-regulated or down-regulated genes (Pairwise Differential Expression and Combined Pathway Analysis results).
- **Type of List.** Subset of significant genes to be considered as the test set. In this case, you can select the subset of genes according to either the regression variables or the experimental groups (Time Course Differential Expression).
- **Reference-set Files.** ID-list with sequences belonging to the reference-set.
- **Two-tailed test.** This option allows us to test for over and under-representation: the test-set will be tested against the reference-set and vice versa.
- **Annotations.** You can select how gene sets are selected for the enrichment analysis: group genes by GO term, by Enzyme Code, InterPro ID, etc.

Click on the *Run* button to start the analysis. It may take a while depending on the number of annotations.

Results Table

Once completed the results table will be shown in a new tab (Figure 2) containing all the annotation terms, displaying a tag only where the adjusted p-values are below given threshold. The columns are:

- **Tag.** It indicates if the GO term has been declared over or under-represented in the test-set.
- **GO Term.** The GO Term ID.
- **GO Name.** The more descriptive GO Name.
- **GO Category.** The are three GO categories: Molecular Function (MF), detailing the biochemical activities of genes; Cellular Component (CC), specifying the physical locations within a cell where functions occur; and Biological Process (BP), encompassing the larger cellular pathways and processes genes are involved in.
- **Adj. P-value.** Corrected p-value by the multiple test correction method chosen (False Discovery Rate control according to Benjamini-Hochberg procedure by default).
- **P-value.** Raw p-Value without multiple testing corrections.
- **Nr Test.** The number of sequences annotated with the GO and in the Test Set.
- **Nr Reference.** The number of sequences annotated with the GO and in the Reference Set.
- **Not Annot Test.** The number of sequences not annotated with the GO and in the Test Set.
- **Not Annot Ref.** The number of sequences not annotated with the GO and in the Reference Set.

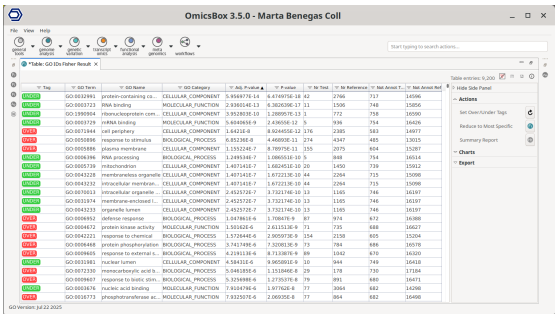


Figure 2: *Enrichment Results Table*

Context Menu

A context menu appears by right-clicking on any row of the results table. The options listed will be applied to the selected rows. The specific options for FET results are:

- **Show Details.** It opens a new tab with more details about the GO term, containing a link to the GO database.
- **Create ID List of TestSet Sequences.** It opens a new tab with a list containing the names of the sequences annotated to the GO that are in the Test Set.
- **Create ID List of RefSet Sequences.** It opens a new tab with a list containing the names of the sequences annotated to the GO that are in the Reference Set.

Sidebar Options

In the sidebar there are located all possible action that can be performed for this enrichment result, including three options for the visual display of the results:

Actions

- **Set Over/Under Tags**: this option allows to define which column (raw p-value or adjusted) should be used to display the enrichment tag, as well as the threshold value. The adjusted p-value column can also be updated by selecting a different multiple test correction method, to choose between:
    - Benjamini-Hochberg: default value, the most commonly used method when controlling for FDR. For further information about how p-values are adjusted by FDR according to Benjamini-Hochberg procedure please refer to the publication: Benjamini, Y., & Hochberg, Y. (1995). Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological), 57*(1), 289-300.
    - Bonferroni: the most restrictive method, recommended to avoid type I errors at all costs. Controls the family-wise error rate (FWER), or the probability of making one or more false discoveries.
    - Benjamini-Yekutieli: method for controlling the FDR, more conservative than Benjamini-Hochberg and designed to work with dependent conditions.
    - Holm: an updated version of the Bonferroni method, less restrictive and controlling FDR rather than FWER.
    - Hochberg: a method similar to Holm.
- **Reduce to Most Specific** (only for GO annotations): use this option to remove more general GO terms from the results and get only the most specific terms (with the lowest level in the GO DAG).
- **Summary Report.** Generates a report containing basic statistics about the analysis, the configuration parameters and the bibliographic references.

Charts

- **Bar Chart**: this option generates a bar display of the percentages of sequences at both, test and reference set, for each annotation of the table (Figure 3). The configuration wizard allows you to select:
  - Tags: whether to plot GO terms with OVER, UNDER, or both tags.
  - GO Categories: whether to plot GO terms from BP, MF, and/or CC categories.
  - Column to plot: whether to display GO IDs or GO Names.
- **Bubble Plot**: this option generates a dot plot, a chart representing 4 dimensions: the annotation term in the Y axis, the gene ratio (Nr Test / [Nr Test + Not Annot Test]) in the X axis, the Adjustet P-value as the dot color, and the number of test sequences of the set as the dot size (see Figure 4). The configuration wizard includes the same parameters as the Bar Chart:
  - Tags: whether to plot GO terms with OVER, UNDER, or both tags.
  - GO Categories: whether to plot GO terms from BP, MF, and/or CC categories.
  - Column to plot: whether to display GO IDs or GO Names.
- **Enriched GO Graph** (only for GO annotations): use this option to generate a representation on the GO DAG (see Figure 5). Nodes are color-highlighted proportionally to their significance value. The user can choose which type of calculated p-value to use for highlighting and the threshold for filtering out nodes. Additionally, the Filter intermediate the checkbox will hide non-enriched nodes. The configuration wizard allows you to select:
  - Tags: whether to plot GO terms with OVER, UNDER, or both tags.
  - GO Categories: whether to plot GO terms from BP, MF, and/or CC categories. More options are available in the graph viewer's sidebar. Gene Ontology Graphs of this manual gives further information on the graphical functions in OmicsBox.
- **Word Cloud**: Generate a word cloud visualization based on the enriched GO names (Figure 6). The word cloud will display the GO names with sizes proportional to their associated statistical values, creating a visual representation of the data. The configuration wizard allows you to select:
  - Tags: whether to plot GO terms with OVER, UNDER, or both tags.

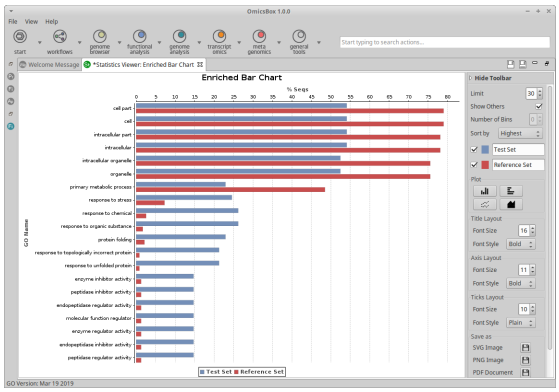


Figure 3: *Enriched Bar Chart*

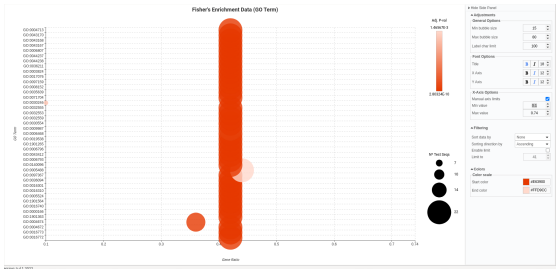


Figure 4: *Bubble Plot Chart*

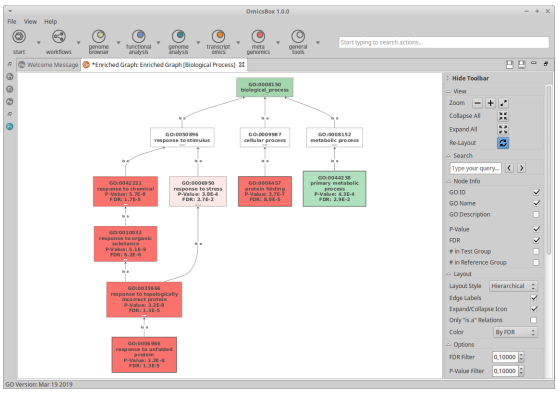


Figure 5: *Enriched Graph*

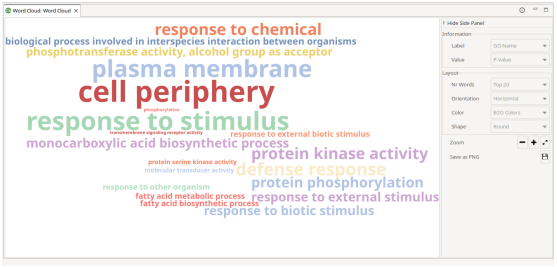


Figure 6: *Word Cloud*

GENE SET ENRICHMENT ANALYSIS (GSEA)

Introduction

OmicsBox includes the GSEA computational method that determines whether an a priori defined set of genes shows statistically significant, concordant differences between two biological states. GSEA considers experiments with genome-wide expression profiles from samples belonging to two classes. Genes are ranked based on the correlation between their expression and the class distinction by using any suitable metric. Given an a priori defined set of genes S (e.g., genes encoding products in a metabolic pathway, located in the same cytogenetic band, or sharing the same GO category), the goal of GSEA is to determine whether the members of S are randomly distributed throughout a ranked list of genes (L) or primarily found at extrems.

If there is no association, genes in S will be uniformly distributed throughout L: that is the null hypothesis of GSEA. If there is association, genes in S will accumulate at the top or at the bottom of L. The magnitude of the association will be measured by the Enrichment Score statistic (ES), see GSEA user guide.

Run GSEA

For this analysis, the completion (but not exclusively) of the involved sequences with their annotations must be loaded in the application. This can either be the result of an OmicsBox annotation or the imported annotation by file (.annot), see Gene Ontology Annotation of this manual.

This functionality can be found as a Side Panel button in the following tables:

- Annotated sequences from Functional Analysis.
- Combined Pathway Analysis results.
- Pairwise Differential Expression results (with and without replicates).

A dialog screen appears (Figure 1). A detailed description of each parameter is available by clicking the help icon next to the parameter. The following explanations refer to the Gene Set Enrichment Analysis run from an Annotated Sequences project in the functional analysis module. To other module's specific implementations, please visit the corresponding user manual section (linked in the bullet points above).

Input Parameters Configuration

- **Rank file**. Ranked list of genes can be selected by uploading text files or ID-Value-List *.box* files containing the lists of sequence IDs and a statistical value for each one.
- **Number of permutations**. Number of gene set permutations to assess the statistical significance of Enrichment Score.
- **Enrichment Statistic**. Each time GSEA encounters a gene in S, a running-sum statistic increases, and decreases if gene is not in S. Enrichment Score (ES) will be 0 if genes in S are randomly distributed throghout L: ES represents the maximum deviation for a random distribution. This option change the way in which ES is calculated (see GSEA paper).
- **Number of Detailed Results**. Set the number of GO terms to get further details.
- **Detailed Results of All GOs**. Check this option to obtain detailed results for all GOs. Be aware that this task is both disk and time consuming.

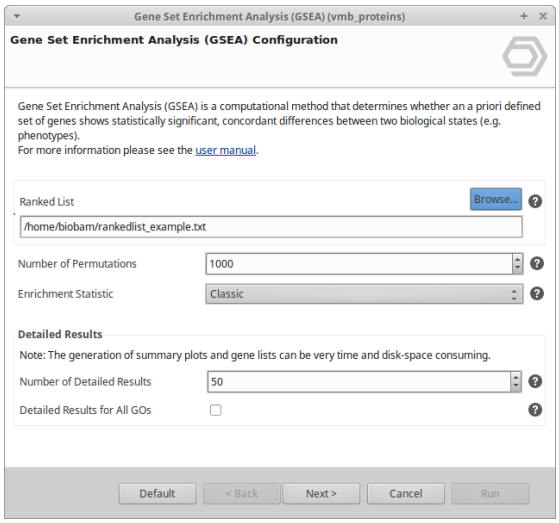


Figure 1: *GSEA Configuration Wizard Page*

Advanced Configuration

- **GO Category**. Select the Gene Ontology Category to run the analysis.
- **Gene Sets Max Size**. Maximum number of genes allowed in a gene set. By default, GSEA ignores gene sets with more than 500 genes because normalization is not very accurate for extremely large gene sets.
- **Gene Sets Min Size**. Minimum number of genes required in a gene set. By default, GSEA ignores gene sets with fewer than 15 genes because normalization is not very accurate for extremely small gene sets. For example, gene sets with fewer than 10 genes can generate significant results with just 2 or 3 genes.
- **Do Not Filter**. By default, only IDs with a higher FDR or p-value than the specified filters will be shown. Check this option to disable the filtering.
- **Filter Mode:** Choose between FDR or P-Value to filter the enriched GOs. Note that FDR is the corrected p-value for multiple testing, so it provides more information about the statistical significance than the raw p-value.
- **Filter Value:** The value of the FDR or P-Value cut-off.

Click on the *Run* button to start the analysis. It may take a while depending on the number of permutations selected.

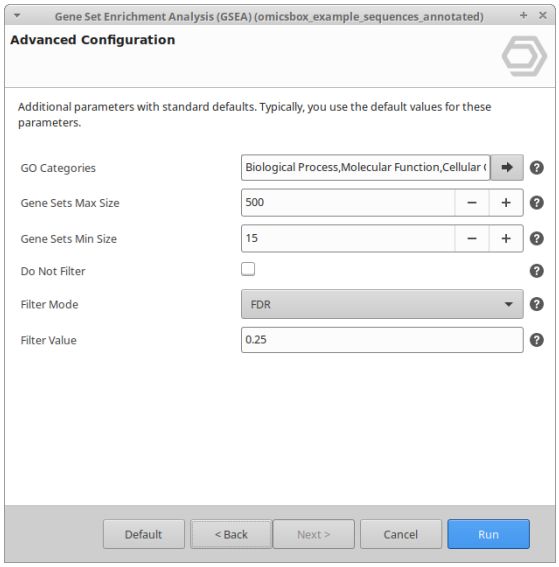


Figure 2: *GSEA Advanced Configuration Wizard Page*

Results

Once completed the results table will be shown in a new tab (Figure 3), where the adjusted p-values of each annotation above a given threshold will be shown. The main columns are:

- **Tags**: Indicates whether a GO term is considered enriched. GOs with the "TOP" tag are over-represented at the top of the ranked list, whereas GO terms with the "BOTTOM" tag are over-represented at the bottom.
- **GO ID**: The Gene Ontology term identifier.
- **GO Name**: The descriptive name of the Gene Ontology term.
- **GO Category**: The GO category (Biological Process, Molecular Function, or Cellular Component).
- **Size**: Number of genes in the gene set after filtering out those genes not in the expression dataset.
- **ES**: Enrichment score for the gene set; that is, the degree to which this gene set is overrepresented at the top or bottom of the ranked list of genes in the expression dataset.
- **NES**: Normalized enrichment score; that is, the enrichment score for the gene set after it has been normalized across analyzed gene sets.
- **Nominal p-val**: Nominal p-value; that is, the statistical significance of the enrichment score. The nominal p-value is not adjusted for gene set size or multiple hypothesis testing; therefore, it is of limited use in comparing gene sets.
- **FDR q-val**: False discovery rate; that is, the estimated probability that the normalized enrichment score represents a false positive finding.
- **FWER p-val**: Familywise-error rate; that is, a more conservatively estimated probability that the normalized enrichment score represents a false positive finding. Because the goal of GSEA is to generate hypotheses, the GSEA team recommends focusing on the FDR statistic.
- **Rank at Max**: The position in the ranked list at which the maximum enrichment score occurred. The more interesting gene sets achieve the maximum enrichment score near the top or bottom of the ranked list; that is, the rank at max is either very small or very large.
- **Leading Edge**: Displays the three statistics used to define the leading edge subset: - Tags: percentage of gene hits before (positive ES) or after (negative ES) the enrichment peak. - List: percentage of genes in the ranked list before (positive ES) or after (negative ES) the enrichment peak. - Signal: enrichment signal strength combining the two previous statistics.
- **Core Enrichment Sequences**: Genes that contribute to the leading-edge subset within the gene set. This is the subset of genes that contributes most to the enrichment result and are the core enriched sequences that account for the enrichment of a certain function.
- **No-Core Enrichment Sequences**: Genes in the gene set that do not contribute to the leading-edge subset.

For further details please refer to the GSEA User Guide.

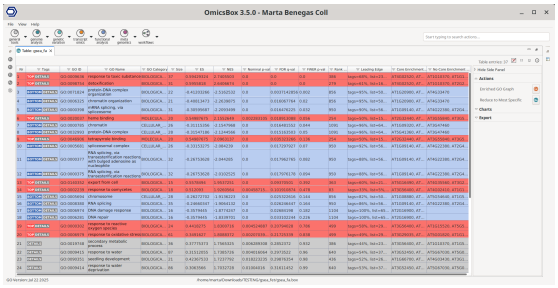


Figure 3: *GSEA result table*

Context Menu

A context menu appears by right-clicking on any row of the results table. The options listed will be applied to the selected rows. The specific options for GSEA results are:

- **Show Details**: it shows more details about the GO term, its enrichment plot, its ES distribution, and the GSEA statistics for each sequence in linked to the GO term.
- **Create ID List of Core Enrichment Sequences**: opens an ID-List with the core enrichment sequences for the given GO term.

Sidebar Options

In the sidebar there are located all possible actions that can be performed for this enrichment result.

Actions

- **Enriched GO Graph**: generate a representation on the GO DAG (Figure 3). Nodes are color-highlighted proportionally to their significance value. The user can choose which type of calculated p-value to use for highlighting and the threshold for filtering out nodes.
- **Reduce to Most Specific**: remove more general GO terms from the results and get only the most specific terms (with the lowest level in the GO DAG).

Charts

- **Show Bar Chart**: compare the core and non-core enriched functions in terms of their abundance of annotated sequences in your dataset (Figure 4). The percentages in the bar chart are calculated as the number of sequences (core or non-core) annotated with each GO. The total is the number of sequences that were provided with the ranked list (all sequences provided for the test).
- **Bubble Plot**: this option generates a dot plot, a chart representing 4 dimensions: the annotation term in the Y axis, the normalized enrichment score (NES) in the X axis, the FDR q-value as the dot color, and the number of test sequences as the dot size (Figure 5). The configuration wizard allows you to select:
    - Tags: whether to plot GO terms with enriched, non-enriched, or both tags.
    - GO Categories: whether to plot GO terms from BP, MF, and/or CC categories.
    - Column to plot: whether to display GO IDs or GO Names.
- **Word Cloud**: Generate a word cloud visualization based on the enriched GOs (Figure 6). The word cloud will display the GO names with sizes proportional to their associated statistical values, creating a visual representation of the data. The configuration wizard allows you to select:
    - Tags: whether to plot GO terms with the OVER, UNDER or both tags.
- **ES Histogram Chart**: this option generates a histogram of enrichment scores across gene sets, which provides a quick, visual way to grasp the number of enriched gene sets (Figure 7).
- **NES vs Significance Chart**: this option generates a plot of p-values versus normalized enrichment scores, which provides a quick, visual way to grasp the number of enriched gene sets that are significant (Figure 8).

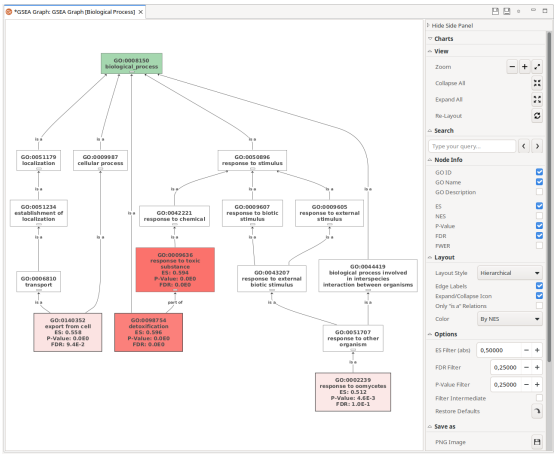


Figure 3: *Enriched Graph*

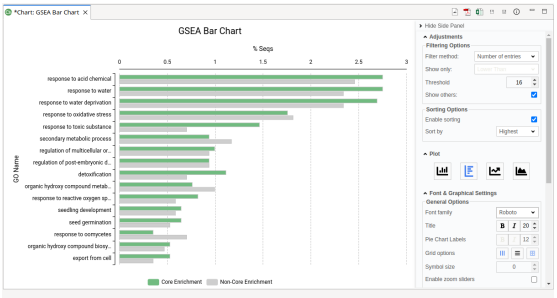


Figure 4: *Bar Chart*

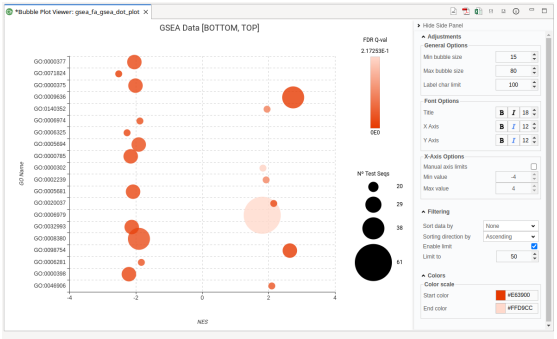


Figure 5: *Bubble Plot*

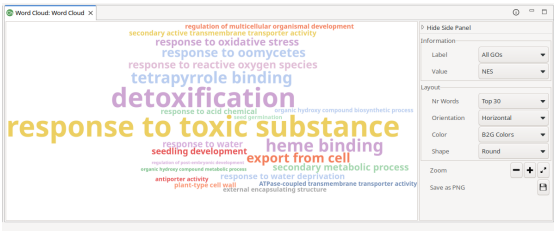


Figure 6: *Word Cloud*

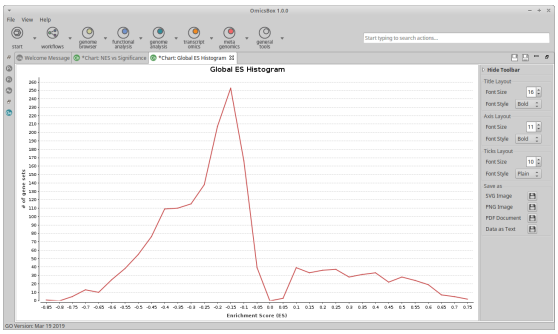


Figure 7: *ES Histogram Chart*

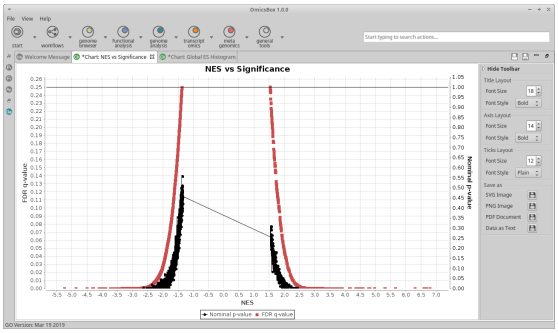


Figure 8: *NES vs Significance Chart*

References

- Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences, 102(43):15545– 15550

Combined Pathway Analysis

INTRODUCTION

Pathway analysis is a useful tool to easily get an overview of the biological mechanisms involved in our data, summarising the information in a way that greatly enhances the capability to interpret the results.

The Combined Pathway Analysis allows two of the most important public pathway databases:

- Reactome: a curated database of pathways and reactions in human biology, but containing inferred orthologous reactions for other 15 non-human species. Reactions can be considered as pathways "steps". Reactome defines a reaction as any event in biology that changes the state of a biological molecule.
- KEGG: a collection of manually drawn pathway maps representing the knowledge of molecular interaction, reaction, and relational networks for metabolism, genetic information processing, environmental information processing, cellular processes, organismal systems, human diseases, and drug development.

The first step of every pathway analysis is to link the sequences to the pathways present in the database.

The structure can change depending on the source but usually, a given pathway has multiple versions, each one associated with a particular species, and so are all the gene products contained within. This means that to link the sequences to a species-specific pathway, traditionally they had to use the same identifiers. An additional way of linking was to use generic annotation data, like GO terms or enzyme codes.

OmicsBox allows to directly link the sequences to pathways by making use of the platform infrastructure, performing an intermediate step to match a gene product (i.e. protein) to the most probable candidate found in the pathway database. More detailed information can be found in the following subsections.

An optional but highly recommended step is the pathway enrichment analysis to provide statistical significance to the previous linking results. The information needed can be automatically retrieved from OmicsBox differential expression objects (pairwise or time-course analysis) or manually provided. See the enrichment section for more information.

Finally, the included viewer offers the possibility to inspect in detail an individual pathway by presenting its topology with a layer of additional info containing matched sequences, products, and expression values heatmaps among other things.

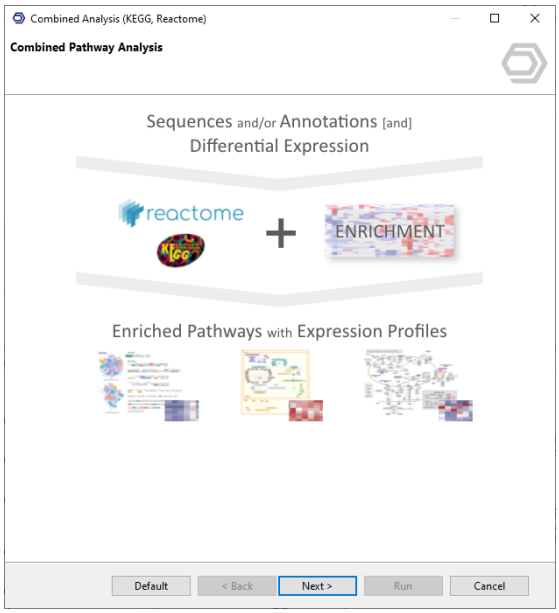


Figure 1: *General Overview*

RUN PATHWAY ANALYSIS

This functionality can be found under Functional Analysis -> Pathway Analysis -> Combined Analysis (KEGG, Reactome). The wizard presents a diagram in the first page, and contains general input and options in the second page, followed by two other pages for Reactome and KEGG settings.

Input options

- **Sequences (.box, .fasta, .annot)**: a file containing sequence info (.fasta, .box), GO / EC annotations (.annot, .box) or both (.box) is required. Depending on the available data some linking options might not be considered in the analysis.
- **Differential Expression Data**: an optional differential expression analysis object. Currently, pairwise and time-course analyses are supported. If it is not provided from the start, it can be added later using the side panel action "Add Differential Expression".

Loading an OmicsBox differential expression analysis object, like pairwise or time course, adds useful information to the pathway object:

- Experimental design information
- Count values
- Differentially expressed features

By providing this data some other features are enabled, like automatically calculating a pre-ranked list for GSEA enrichment analysis or selecting the test-set features in Fisher's choosing by tag. Also, the pathway viewer is enhanced by being able to show expression heatmaps.

To remove current expression data, as well as enrichment information generated with it, the option "Clear current expression data" from the side panel "Add Differential Expression" can be used later.

- **Pathway Enrichment Analysis**: to sort the pathway results table by statistical significance an enrichment analysis needs to be performed. As previously stated, providing a differential expression analysis allows to automatically retrieve differentially expressed features from it and run an enrichment analysis with default options. If enrichment is not enabled in this page, it can be manually done later using the sidebar actions "Fisher's Enrichment Analysis" and "Gene Set Enrichment Analysis".
- **Fisher's Exact Text**: for a detailed description of Fisher's Exact Test, see this page.

  In a Fisher's pathways enrichment, the reference set used are all those sequences associated with a pathway. By default the test-set list is generated by selecting all the sequences with at least one differentially expressed tag (for example, both UP and DOWN tags in a pairwise analysis), with two tailed parameter set to true and the results filtered using FDR < 0.05 .

  To see the enrichment table or customize the settings, the side panel option "Fisher´s Enrichment Analysis" can be used later. + **Gene Set Enrichment Analysis**: for a detailed description of GSEA, see this page.

  The Gene Set Database generated for GSEA pathway enrichment contains every sequence associated with each pathway. However, only those records present also in the ranked list will actually be used.

  The ranked list will be automatically generated based on the statistics found in the differential expression analysis. The formula used to to rank each sequence is: **Rank = sign(logFC) * - log10(p-value)**

  By default the settings are set to 1000 permutations, maximum gene set size of 500 and minimum gene set size of 15. Because both the logFC and p-value are needed for the formula, currently only the pairwise differential expression analysis is compatible with automatic GSEA, as the time course does not provide logFC information.

  To see the enrichment table, access to GSEA detailed plots or customize the settings, the side panel option "Gene Set Enrichment Analysis" can be used later.

Note that when performing pathway analysis on transcripts, rather than genes, enrichment analysis should not be used. This is because both Fisher's Exact Test as well as GSEA expect their input sequences to be independent of each other, which is not the case for transcripts belonging to the same gene.

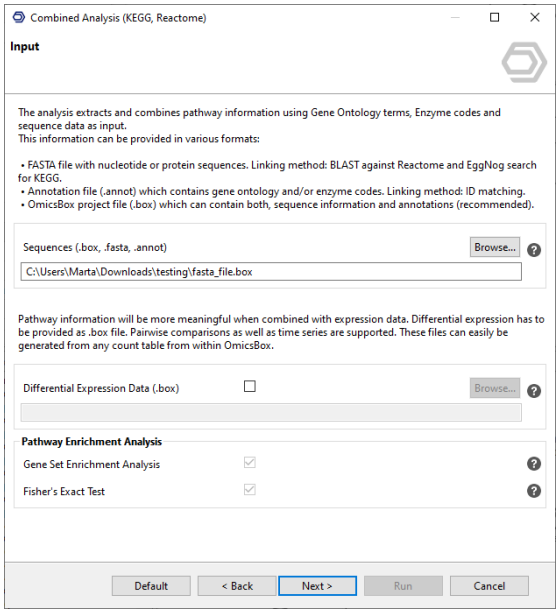


Figure 2: *Input Options*

Configuration Reactome Pathway Analysis

- **Run Reactome Pathway Analysis**: include Reactome database in the analysis or not.

- **Pathway Linking Options:**
  - **Run Blast to link via Protein IDs**: requires having sequence data in our input. This will run a BLAST against a custom database containing the sequences of all available Uniprot proteins associated with a pathway in Reactome. Note: this will consume cloud units.
  - **Link with GeneOntology Terms**: requires having annotated GO terms for each feature. This will link to pathways directly using the GO BP, then a GO MF to associate to the reactions contained in it.
- **Filtering Options**:
  - **Keep Most Specific Pathways**: Reactome's database can contain different species-specific versions of the same pathway (inferred by orthology). At the same time, pathways are organized hierarchically, so the obtained table could contain too many general entries that might not be of interest. This setting will try to discard similar entries, whenever possible, in two ways:
    - If a specific pathway is found, the parents will not be reported.
    - If the pathway is found for multiple organisms and priority has been given to a taxon, only the pathway specific to that taxon will be returned if it has been found, otherwise, all of them will be used.
  - **Give Priority to Taxon**: this setting works in conjunction with the previous one when a pathway has been found for different species. It also affects the BLAST top hit selection; when enabled, the BLAST top hit results will be scanned, choosing the top priority taxon over the other ones or, if not found, it will choose the first one. Note that this selection process takes place over hits meeting the specified e-value criteria.
  - **Top Priority Taxon**: organism to give priority. Reactome is primarily based on human reactions but it contains pathways for other species that might be closer to the dataset used. See the settings "Keep Most Specific Pathways" and "Give Priority to Taxon" for more information.
  - **Blast Expectation Value**: the statistical significance threshold for reporting matches against the database.
  - **Include Categories**: by selecting only the categories of interest the final number of pathways reported is reduced, which could have a positive impact on the multiple testing correction performed on the enrichment analysis.

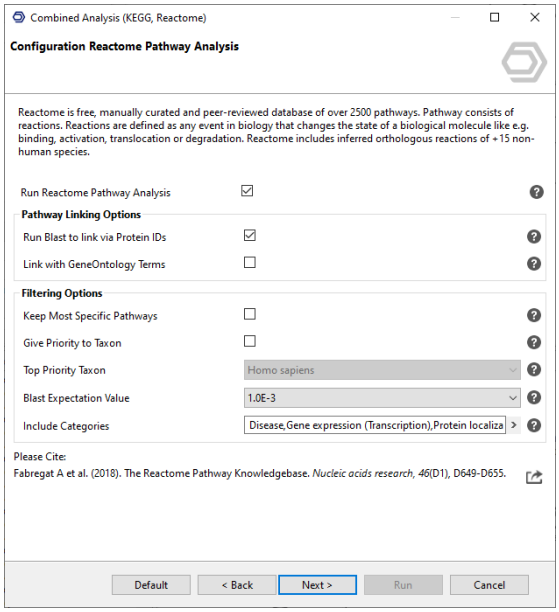


Figure 3: *Reactome Database*

Configuration Gramene Pathway Analysis

- **Run Gramene Pathway Analysis**: include Plant Reactome (Gramene) database in the analysis or not.
- **Pathway Linking Options:**
  - **Run Blast to link via Protein IDs**: requires having sequence data in our input. This will run a BLAST against a custom database containing the sequences of all available Uniprot proteins associated with a pathway in Gramene. Note: this will consume cloud units.
  - **Link with GeneOntology Terms**: requires having annotated GO terms for each feature. This will link to pathways directly using the GO BP, then a GO MF to associate to the reactions contained in it.

- **Filtering Options**:
  - **Keep Most Specific Pathways**: Gramene's database can contain different species-specific versions of the same pathway (inferred by orthology). At the same time, pathways are organized hierarchically, so the obtained table could contain too many general entries that might not be of interest. This setting will try to discard similar entries, whenever possible, in two ways:
    - If a specific pathway is found, the parents will not be reported.
    - If the pathway is found for multiple organisms and priority has been given to a taxon, only the pathway specific to that taxon will be returned if it has been found, otherwise, all of them will be used.
  - **Give Priority to Taxon**: this setting works in conjunction with the previous one when a pathway has been found for different species. It also affects the BLAST top hit selection; when enabled, the BLAST top hit results will be scanned, choosing the top priority taxon over the other ones or, if not found, it will choose the first one. Note that this selection process takes place over hits meeting the specified e-value criteria. Currently the Gramene BLAST database does not contain sequence information for all organisms, so selecting an unavailable species in this parameter will only have effect when selecting a pathway is found for different species, but not for choosing over the top BLAST hits.
  - **Top Priority Taxon**: organism to give priority. See the settings "Keep Most Specific Pathways" and "Give Priority to Taxon" for more information.
  - **Blast Expectation Value**: the statistical significance threshold for reporting matches against the database.
  - **Include Categories**: by selecting only the categories of interest the final number of pathways reported is reduced, which could have a positive impact on the multiple testing correction performed on the enrichment analysis.

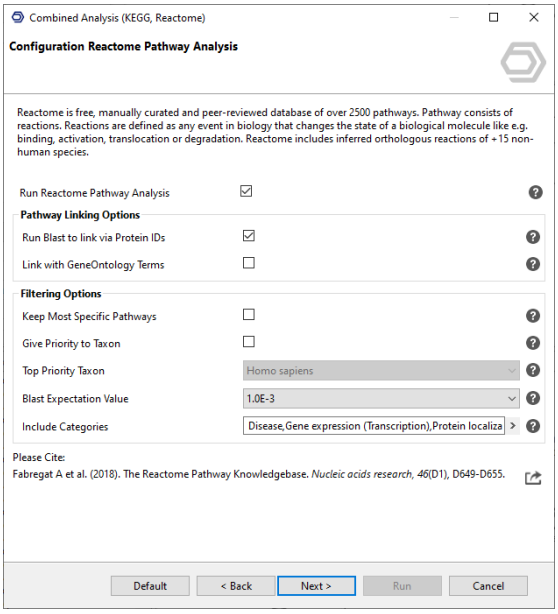


Figure 4: *Gramene Database*

Configuration KEGG Pathway Analysis

- **Run KEGG Pathway Analysis**: include KEGG database in the analysis or not.
- **Pathway Linking Options**:
  - **Link KEGG Orthologs via EggNog**: use the sequence data to retrieve the target orthologs using eggNOG mapper (more info).
  - **Link via Enzyme Codes**: direct link to pathways using the sequence annotated enzymes codes.

- **Filtering Options**:
  - **Include Categories**: by selecting only the categories of interest the final number of pathways reported is reduced, which could have a positive impact on the multiple testing correction performed on the enrichment analysis.

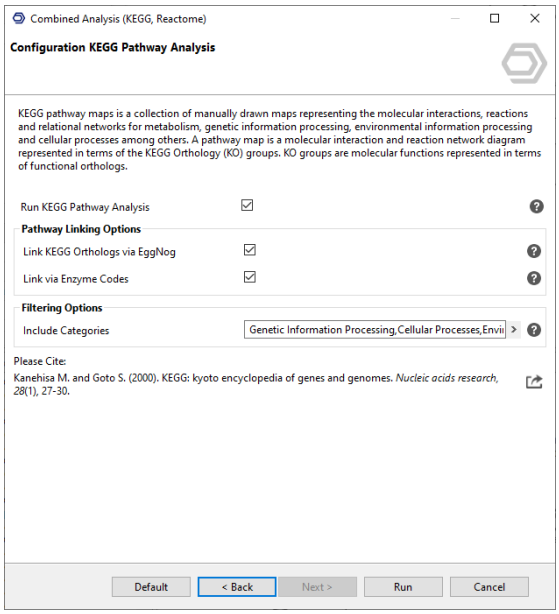


Figure 5: *KEGG Database*

ANALYSIS RESULTS

Results Table

After identifying the pathways associated with the sequences a table will open (figure 5). If it contains enrichment statistical information, it will be automatically sorted using first the absolute GSEA NES or, if not available, Fisher's p-value.

The number of sequences linked to a pathway is usually correlated with the size of the pathway: the bigger it is, the bigger the chance of having sequences linked from the original dataset. That is why sorting by enrichment statistical significance is preferred over the total number of sequences (column #Seqs).

Under some circumstances, on Reactome pathways the total number of sequences (column "#Seqs") associated to a particular BP term (pathway) might be considerably higher than the sequences actually linked to the reactions contained in it (MF term); in those cases the column "#Linked Seqs" might provide a better representation.

For sequences which are not present in the provided differential expression results, the value in the column "#Diff. Expr Seqs" may be "."; this signifies that, although there are sequences associated with the given pathway, none of them have any expression data and are therefore not differentially expressed.

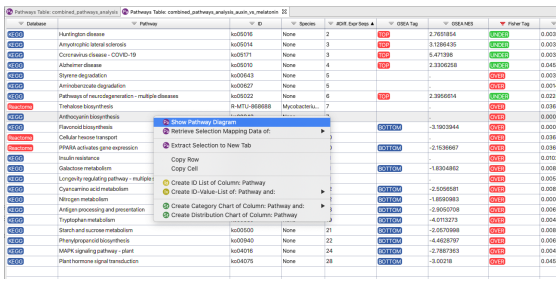


Figure 6: *Pathways Results Table*

Context Menu

- **Show Pathway Diagram**: open the pathway viewer.

- **Retrieve Selection Mapping Data**:
    - **Linked Sequences**: sequences found in the selected pathways.
      Note: for Reactome pathways containing GO terms this will not count the sequences associated to the BP term.
    - **Differentially expressed linked sequences**: differentially expressed sequences found in the selected pathways.
    - **Found GO CC Terms**: only for Reactome pathways, return found GO CC terms. The list might be empty even if GO MF have been found since they are not a requirement in the matching process.
    - **Found MF Terms**: only for Reactome pathways, return found GO MF terms.
    - **Found Entities**: only for Reactome pathways, return found entities. Currently the entities are Uniprot proteins.
    - **Found Enzymes**: only for KEGG pathways, return found enzyme codes.
    - **Found KEGG Orthologs**: only for KEGG pathways, return found KEGG orthologs.

Side Panel Options

- **Summary Report**: report containing the most important findings of the analysis, including linked pathways per database and number of sequences. If enrichment data is available, it will show top 10 enriched pathways. Information about the original input data and parameters is also displayed.
- **Add Differential Expression**: add differential expression data to a pathways analysis project, overriding the previous one if exists. Checking "Remove current expression data" will clear previous expression and enrichment info.
- **Fisher's Enrichment Analysis**: perform a pathway enrichment analysis using Fisher´s statistical method. All sequences linked to at least one pathway will be considered as the reference test. The test-set can be manually provided or automatically generated with the selected tags if a differential expression analysis has been added to the project. Enable the option "Open enrichment projects" to show the enrichment table project for each database.
- **Gene Set Enrichment Analysis**: perform a pathway enrichment analysis using GSEA. The pre-ranked list can be manually provided or automatically generated if a pairwise expression analysis has been added to the project (see the "Input" section for more information). Enable the option "Open enrichment projects" to show the enrichment table project for each database.
- **Generate Charts**: create charts summarizing the results.
    - **Basic stats**: one bar chart with the number of found and enriched pathways grouped by database.
    - **Category distribution**: one bar chart per database with the number of pathways found and enriched for each category.
    - **Fisher's Enrichment stats:** one bubble plot per database, showing the Fisher's enriched pathways with rich ratio (differentially expressed sequences/linked sequencess ratio) as X axis, FDR as color value and number of sequences as point size. Needs expression data loaded into the pathway project.
    - **GSEA Enrichment stats**: one bubble plot per database, showing the GSEA enriched pathways with rich ratio (differentially expressed sequences/linked sequencess ratio) as X axis, NES as color value and number of sequences as point size. Needs expression data loaded into the pathway project.
- **Export data**: export the information contained within each row that is not directly visible on the table (associated terms, sequences…). It provides different configuration options to format the output:
    - **Data to include**: optional columns to append. Note that some of them will be empty for some tables.
    - **Include counters**: include the number of terms found for each column.
    - **Column separator**: character to separate the columns.
    - **Item separator**: character to separate the items inside each column.
    - **Grouping**:
        - One Sequence per Row: one line for each found sequence.
        - One Item per Row: one line for each term found.
        - One Pathway per Row: one line for each pathway.
- **Export pathway diagrams**: export all pathway diagrams from the current results table using the default pathway viewer configuration. Since this action exports diagrams for all pathways in the table, the resulting files may require significant storage space. To export a subset, first select the relevant rows and export the selection to a new project, then export the diagrams from the filtered dataset.

- **Search Function**: an autocomplete search widget to search in the information contained inside pathways. In order to work first a text needs to be entered in the input, then an option must be selected in the suggestion panel before clicking "Search". The clear text button will only remove the input text but not reset the table filtering. Currently wildcards are not supported.

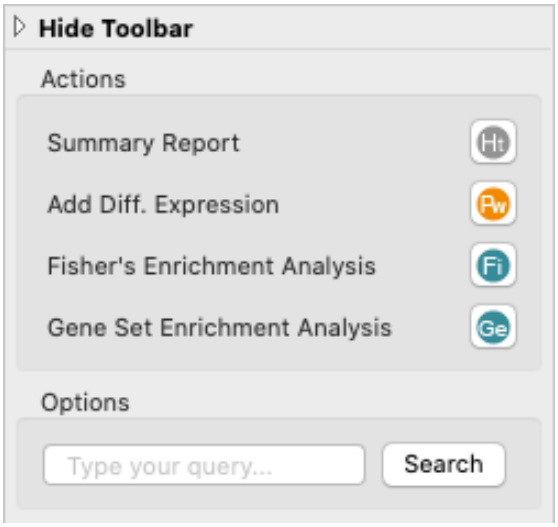


Figure 7: *Side Panel Options*

Pathway Viewer

If the enrichment analysis provides statistical significance about the identified pathways, the pathway viewer helps us to visualize the reactions or interactions between elements in one pathway.

This functionality can be found in the context menu (right-clicking) of each pathway's results table row, under the option "Show Pathway Diagram".

Pathway Map

This is the visual representation of the pathway map (figure 7). In the top-left corner two configuration settings can be found:

- **Center Pathway**: zoom the pathway diagram to fill the available space. Some Reactome pathways might make use of more general diagrams, so using this option in those cases will center around the subset of elements to consider (sub-pathway).
- **Toggle Minimap**: display a minimap view representing the visible diagram portion over the total.

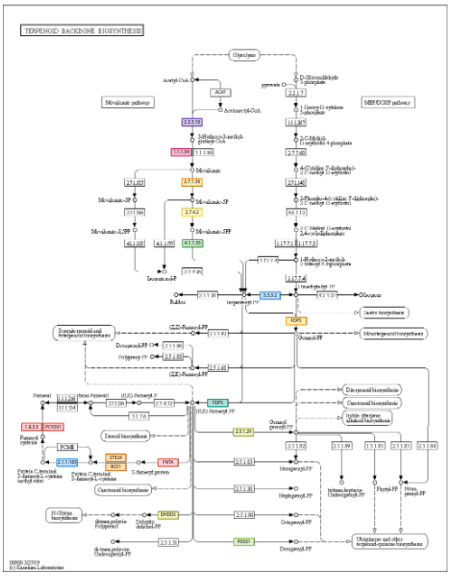


Figure 8: *Pathway Map*

Labels and background coloring

Elements in the pathway are drawn differently depending on the type of map. For Reactome, the entity nodes have a static text that never changes, whereas KEGG displays a dynamic term at the center of each box, dependent on the current settings and the sequences found, so the label change after toggling one or multiple entries is expected behavior for that type of map.

In the default map view mode, the background coloring will use a solid color, assigned to each term or reaction. Because the number of available colors in the palette is limited, repeated colors can be present in medium pathways.

After enabling map expression view mode, a heatmap will be used instead of the color if the element has at least one sequence with expression data. This is the default view mode when expression data is available.

Tool Tip

The tool tip shows details of each element of a pathway (figure 9). The content is different for each pathway database and in every case the information displayed will follow the current filtering settings (i.e. enabled reactions).

On Reactome pathways the tool tip contains:

- Title: type of the element, as described by the diagram file.
- Subtitle: name of the element, usually matching the rendered text.
- Found reactions: reactions associated to the element; one element can be shared with multiple reactions, each with its own color.
- Entities in this node: when BLAST is used as linking option, the sequence is associated to one protein and the participation of said protein in a reaction can be pointed to a particular set of elements (nodes). If the tooltip's element contain this specific information, that will be the one displayed, distinguished by having a green background.
- GO/Entities in this reaction: gene ontology only allows to associate a sequence with a reaction, not to specific elements in it. When no "Entities in this node" is available, the terms associated to the reaction will be shown instead, distinguished by having a blue background.
- Associated Expression in this node: only with differential expression data. Will show a heatmap of the first 5 sequences associated to this node.

On KEGG pathways the tool tip contains:

- Title: type of the element, can be "ortholog", "enzyme" or both, depending if the pathway has KO and EC versions in KEGG.
- Subtitle: names and descriptions of currently enabled terms associated to the element.
- Associated Reactions: in KEGG reactions are only available for metabolic pathways.
- EC/KO in this node: terms associated to the element, only visible for pathways with reactions.
- Associated Terms: if no reactions are available, all the terms associated to the element are shown.
- Associated Expression: only with differential expression data. Will show a heatmap of the first 5 sequences associated to this node.

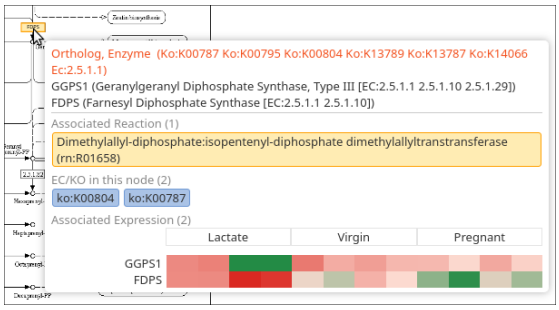


Figure 8: *Tool Tip*

Side Panel Configuration

- **Search**: select one of the elements from the drop-down to show only entries associated with it.
- **View mode**: change the information shown in the sidebar panel (figure 10). Available modes are slightly different between KEGG and Reactome viewers.
  - **Group by sequences**: show one entry per term, these may be KO (Kegg ortholog) and EC (enzyme code) in KEGG, and ET (entity: Uniprot protein) and GO (gene ontology) in Reactome.
  - **Group by reactions**: show one entry per reaction; note that in KEGG, reactions are only available in metabolic pathways.
  - **Group by ortholog groups**: only available on KEGG pathways, show one entry per box (containing related orthologs).
  - **Expression**: show a heatmap containing all the expression values present in the pathway, matching the current filtering settings.
- **Paint expression data in the map**: only with expression data. Activate map expression view mode.
- **Show only results with differential expression data**: only with expression data. Limit entries to those that have at least one differentially expressed sequence.
- **Advanced heatmap options**:
  - **Paint heatmap header using the factor**: only with expression data. Select how the samples should be ordered or grouped in the heatmap; it can be a factor or two-factor combination of the available in the experimental design table.
  - **Paint heatmap values using the attribute**: only with expression data. Select which value (z-score or log CPM) should be used for coloring the heatmap.
  - **Use mean values in heatmaps**: only with expression data. Use the average of the samples included in the selected condition instead of individual values.
  - **Cluster heatmap results**: use hierarchical clustering to sort the rows of the heatmap to group sequences by similar expression patterns.
  - **Re-calculate attributes from group raw counts**: only with expression data. When using mean values in the heatmap, instead of calculating the average of the individual selected attribute (z-score/log CPM) for the corresponding samples, it calculates the average of the counts for those samples and then the real log CPM or z-score .
- **Scale the heatmap color range using individual factors**: only with expression data. To create the heatmap color scale, the range is set considering the sequences found in the whole project, not only the current pathway, and using individual samples excluding outliers; by enabling this setting, the range will be based on the selected factor, which means that instead of taking the individual sample values, it will be based on the average.

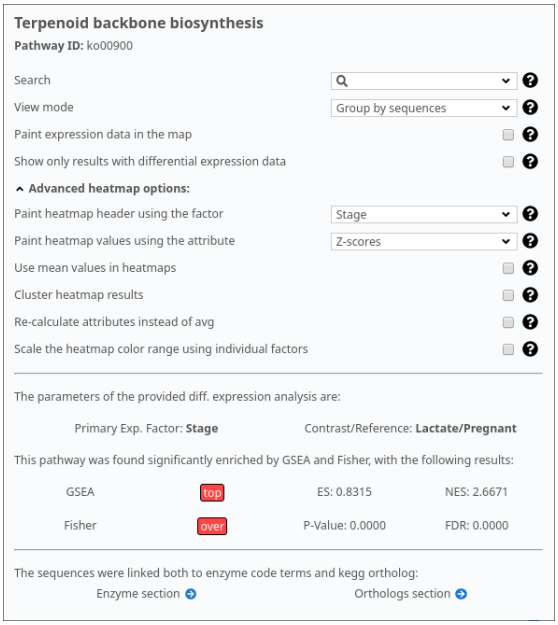


Figure 10: *Side Panel - configuration*

Information Panel

The information panel shows different data depending on the selected "View mode". Common functionalities for panel entries are the "ID" button to create an ID list of the sequences associated to the entry, and a toggle button to paint it or not on the pathway map.

- "Group by sequences" view mode: it can show up to 2 blocks:
  - "Linked Enzyme Sequences" and/or "Linked KO sequences" for KEGG pathways.
  - "Linked MF sequences" and/or "Linked Entity Sequence" for Reactome pathways.
- In the sequences grouping mode, there will be a row for each found term (enzyme code, kegg ortholog, GO molecular function term or entity/protein) with the following information:
  - Sequences associated to the term.
  - Type, ID and name of the term.
  - Reactions (if any) associated to the term. Placing the cursor over the icon will highlight the reaction in the pathway map.
  - Associated Expression: heatmap of the associated sequences having expression data.
- "Group by reactions" view mode: only available for Reactome and KEGG metabolic pathways. In this grouping mode there will be a row for each reaction with the following information:
  - Sequences associated to the reaction.
  - Type, ID and name of the reaction.
  - Associated terms (enzymes, entities...)
  - Associated expression: heatmap of the associated sequences having expression data.
- "Group by ortholog groups" view mode: only available with KEGG pathways. There will be a row for each ortholog group (a box in the pathway map) with the following information:
  - Sequences associated to the ortholog group.
  - Type, ID and name of all the enzymes or orthologs associated to the group.
  - Associated expression: heatmap of the associated sequences having expression data.
- "Expression" view mode: in this mode a general heatmap including all expression data found in the pathway is rendered, including a summary containing the number of found tags for each tag (up, down, none...).

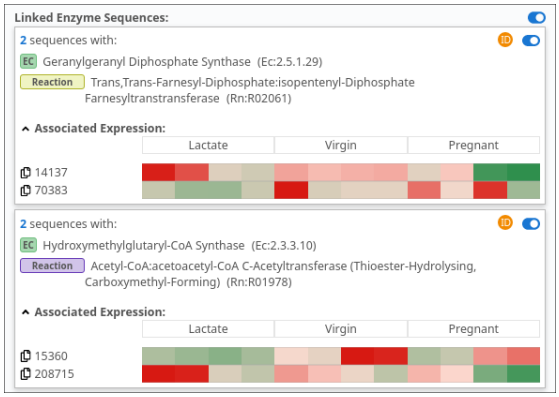


Figure 11: *Side Panel - information panel*

Pathway Hierarchy

Contains the category tree of the current pathway.

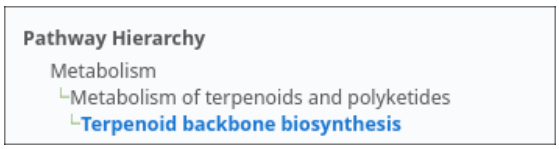


Figure 12: *Side Panel - pathway hierarchy*

ADDITIONAL INFORMATION

Example Datasets can be found within the Functional Analysis Module Example Data: https://resources.biobam.com/omicsbox/example_data/version_2_0_0/FunctionalAnalysis.zip

REFERENCES

- Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990) "Basic local alignment search tool." J. Mol. Biol. 215:403-410.
- BioBam Bioinformatics. (2019). OmicsBox - Bioinformatics made easy (Version 1.4.337). Retrieved March 3, 2019, from https://www.biobam.com/omicsbox.
- Fast genome-wide functional annotation through orthology assignment by eggNOG-mapper. Jaime Huerta-Cepas, Damian Szklarczyk, Lars Juhl Jensen, Christian von Mering and Peer Bork. Submitted (**2016**).
- eggNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. Jaime Huerta-Cepas, Damian Szklarczyk, Kristoffer Forslund, Helen Cook, Davide Heller, Mathias C. Walter, Thomas Rattei, Daniel R. Mende, Shinichi Sunagawa, Michael Kuhn, Lars Juhl Jensen, Christian von Mering, and Peer Bork. Nucl. Acids Res. (04 January **2016**) 44 (D1): D286-D293. doi: 10.1093/nar/gkv1248
- Fabregat A et al. (2018). The Reactome Pathway Knowledgebase. Nucleic acids research, 46(D1), D649-D655.
- Kanehisa M. and Goto S. (2000). KEGG: kyoto encyclopedia of genes and genomes. Nucleic acids research, 28(1), 27-30.
- Naithani S et al. (2020). Plant Reactome: a knowledgebase and resource for comparative pathway analysis. Nucleic acids research, 48(D1), D1093-D1103.

Comined Graph

INTRODUCTION

OmicsBox generates combined graphs where the annotation of a group of sequences is visualized together. This can be used to study the joined biological meaning of a set of sequences. It can be used to visualize results at different stages of the application.
Combined graphs are a good alternative to enrichment analysis where there is no reference set to be considered or the number of involved sequences is low.

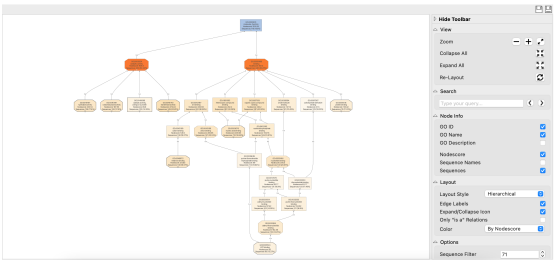


Figure 1: *Combined graph visualization*

Graph Drawing Configuration

The following parameters are available:

- Graph Title
- GO Categories

For each Gene Ontology category, a graph will be displayed. OmicsBox allows extracting information from the graph nodes such as tooltip (figure 4), create a subgraph from that specific GO, create an Id list of the sequences that have been annotated with that particular GO (figure 5). The generated Id list can then be used within OmicsBox in the select by sequences feature (see Selection Section).

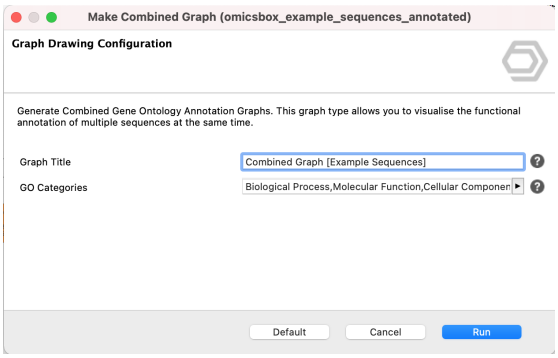


Figure 2: *Combined Graph Drawing Configuration Dialog allows to provide a graph title header and to choose between the different GO categories*

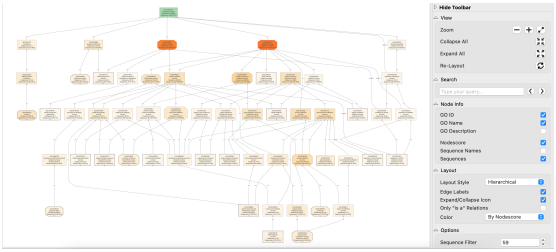


Figure 3: *Biological Process Combined Graph*

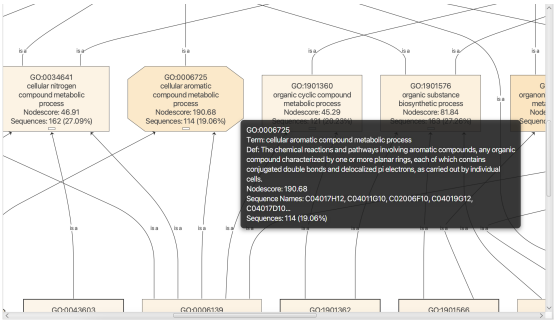


Figure 4: *Graph Node Tooltip*

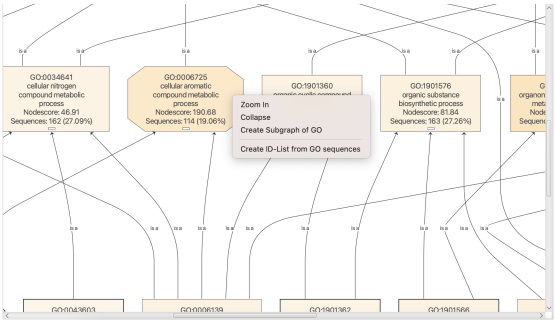


Figure 5: *Extract Node Information*

RESULTS

Graph element legend

Gene Ontology term obtained by mapping which can directly be associated with one or more BLAST hits. (GO-Accession, maximum hit e-value assigned, max. hit similarity assigned, number of hits belonging to this)

Non-annotated GO term node (GO term name, mean e-value of all hits contributing to this node, max. e-value, max. Similarity, number of Hits contributing to this node, Annotation Algorithm Score)

Annotated GO term node (GO term name, mean e-value of all hits contributing to this node, max. e-value, max. Similarity, number of Hits contributing to this node, Annotation Algorithm Score)

The GO Graphs are displayed in different shapes:

- **octagon:** Annotated GO Terms
- **square:** Intermediate GO Terms
- **ellipsis:** GO Terms linked to a Blast Hit

Graph Side Panel

The generated combined graph is interactive and its parameters can be modified from the side panel.

- View. This section controls the graph visualization within its area.
- Zoom: Zoom in/out is supported on the mouse wheel or from the icons.
- Collapse All: The nodes will collapse and only the root will be visualized.
- Expand All: The nodes will expand to the original graph visualization.
- Re-Layout: The whole graph will be re-scaled to adjust to the visualization area.
- Search. Allows to search for GO IDs/ Terms/ Description in the Combined Graph.
- Node Info. This parameter controls the information shown at a node. Possible values are:
- GO ID: If checked the GO ID will be included in the node.
- GO Name: The GO Names are shown in the node.
- GO Description: When checked the GO Description will be included in the node.
- Nodescore: The node score will be shown in the node.
- Sequence Names: The names of the sequences annotated at each GO are included in the node. The limit number of names to be displayed is 15.
- Sequences: The number of sequences annotated with that particular GO will be displayed in the node.
- Layout.
- Edge Labels: When checked the labels on the edges will be shown.
- Expand/Collapse Icon: If checked the ions that represent expand/collapse on the node are displayed.
- Only "is a" Relations: Only the is a relation between nodes will be displayed if the box is checked.
- Color: OmicsBox highlights nodes proportionally to some parameter of the analysis which result is visualized on the DAG.
    - Ontology: All nodes will be colored according to the ontology category, Biological Process - green; Molecular Function - blue; Cellular Component - yellow.
    - White: The nodes will turn white.
    - By Nodescore: A Score is computed at each node according to the formula:

$$score = \sum_{GOs} seq \times \alpha^{dist}$$

  where seq is the number of different sequences annotated at a child GO term and dist the distance to the node of the child. GO term Coloring by Score will highlight areas of high annotation density. - By Sequence Count: Node color intensity will be proportional to the number of contributing sequences at the node. * Options.
- Sequence Filter: The minimal number of sequences a GO node must have assigned, to be displayed. This filter is used to control the number of nodes present in the graph. It is recommended to start the analysis with a high number that, depending on the number of total sequences, is expected to overload the graph. Depending on the result adjust this value until you obtain a satisfactory graph. Start with 10% of your total number of sequences.
- Nodescore Filter: OmicsBox allows modulation of graph size by introducing node filters that depend on the type of graph considered.
- Score alpha. The value for parameter alpha in the Score formula Node Score Filter. Only nodes with a Score value higher than the Filter will be shown. Use this parameter to thin out the GO-DAG for low informative nodes.
- Restore Defaults: All filters will be set to the default values.
- Charts. (see next section)
- Save as. The information present in a Combined Graph can be saved as an image (.png) or in table format. This will generate a .txt file where all information related to each node of the plotted Graph is provided in different columns.
- Overview. Provides a radar-like view of the graph, which allows adjusting the visible window.
- Open With. Open the graph information as WordCloud (see following section).

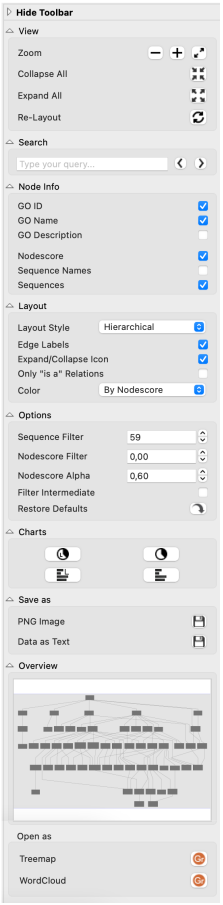


Figure 6: *Combined Graph Side Panel*

Graph Charts

Analysis of GO Term associations in a set of sequences can also be done by Pie/Bar Charts. For this analysis, a Combined Graph must have been generated first. Once the graph is visible in the GO Graph panel you can find several icons to visualize the 4 different types of charts.

Four possibilities are available:

1. Sequence distribution by GO level (Pie-Chart): This pie chart represents the number of sequences for each Gene Ontology term for a given level. See figure 8.
2. Sequences per GO terms (Multilevel Pie): This function generates a Pie with the lowest node per branch of the DAG that fulfills the filter condition., e.g. will find all the lowest nodes with the given number of sequences or Score value and will plot them jointly in a Pie representation. See figure 9.
3. Gene Level (Bar-Chart): A bar chart representing the GO terms according to the number of annotated sequences. See figure 10.
4. Sequence distribution by GO level (Bar-Chart): This bar chart represents the number of sequences for each Gene Ontology term for a given level. See figure 11.

When any of these functions are called, a table of node counts is generated and displayed in the statistics tab.

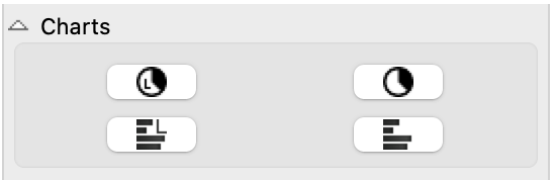


Figure 7: *Combined Graph Pie and Bar-Charts*

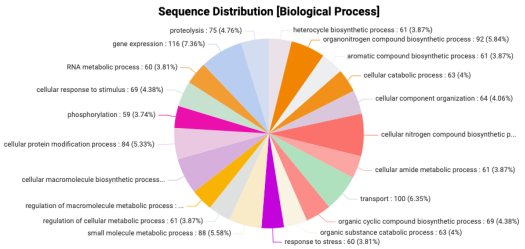


Figure 8: *Sequence distribution by GO level: Pie Chart*

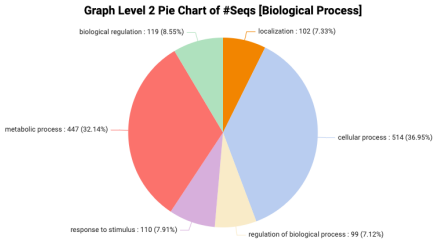


Figure 9: *Sequence Distribution/GO as Multilevel-Pie (#score or #seq cutoff)*

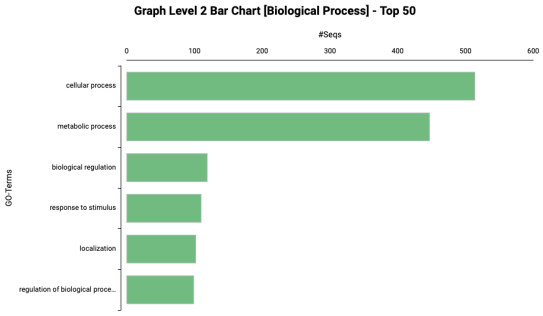


Figure 10: *Biological Process Level 2*

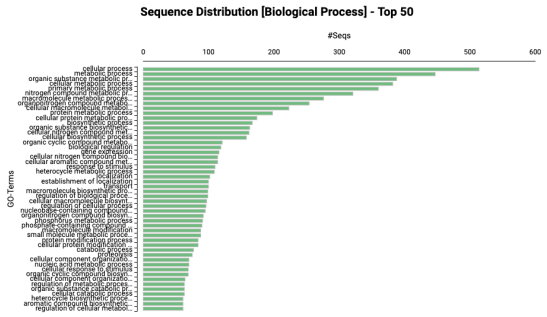


Figure 11: *Sequence distribution by GO level: Bar Chart*

WordCloud

A WordCloud is a visual representation of a list of labels. The importance of words, here GO terms, is represented by its font size. The font size depends on either the sequence count or the NodeScore of each GO term. The list of words can be limited to a specific Gene Ontology category (BP, CC or MF). The coloring is random. Several options to change the graphical appearance are available like the number of words, the orientation and shape of the cloud as well as the color scheme.

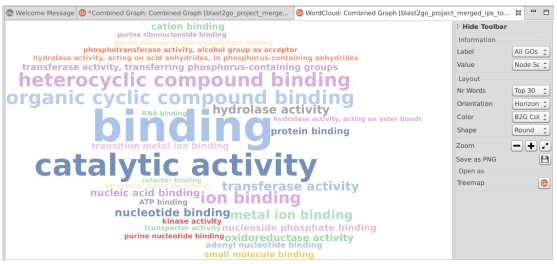


Figure 12: *Convert Graph to Word Cloud*

4.5.9 Gene Ontology Graphs

Introduction

The Gene Ontology structure can be described in terms of a graph, where each GO term is a node, and the relationships between the terms are edges between the nodes. GO is loosely hierarchical, with child terms being more specific than their parent terms. A child term may have more than one parent term. There exist different types of relationships between child and parent terms: *is a* (is a subtype of); *part of*; *has part*; *regulates*, *negatively regulates,* and *positively regulates*. Children that represent a more specific instance of a parent term have *is a* relationship with the parent. Children that are a constituent of the parent term have a *part of* a relationship. The three GO categories (cellular component, biological process, and molecular function) are each represented by a separate root ontology term.

OmicsBox offers the possibility of visualizing the hierarchical structure of the gene ontology by directed acyclic graphs (DAG). OmicsBox integrates a viewer for graph visualization. It allows fast navigation and zooms on the GO DAG. OmicsBox provides various ways for the joint analysis of groups of annotated sequences.

It is possible to generate these graphs in OmicsBox:

- **Simple GO Graph:** Generates the GO graph of the provided GOs.
- **Colored Graph:** Generates the GO graph from a text file.
- **Combined Graph:** Generates the GO graph to visualize the annotation results.

These functionalities are available under the **functional analysis → Gene Ontology Graph** and the **Combined Graph** on the side panel once a sequencing project has been loaded.

Simple GO Graph

The "Make GO Graph" function allows visualizing any set of GO terms/Ids and these have to be provided by the user (figure 2).

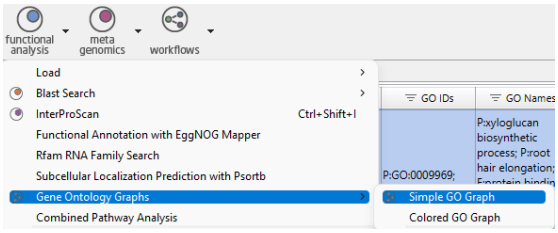


Figure 1: *Simple GO Graph*

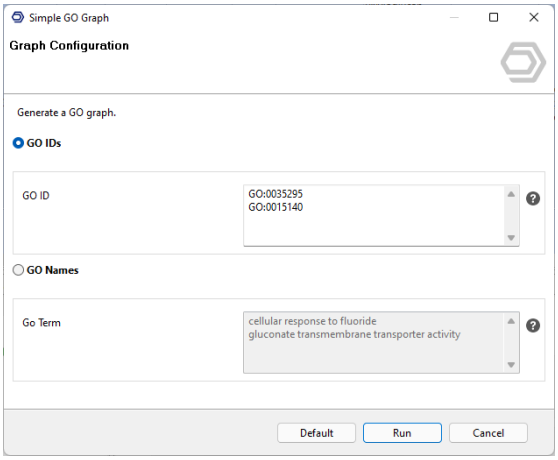


Figure 2: *Simple GO ID Graph*

Colored Graphs

We can generate a GO graph from a text (.txt) file which contains a list of GOs and the desired color for each of them. It is also possible to label groups of GOs with the same name. Figure 4 shows an example that was created introducing the following text file:

```
GO:0000003    6    Group A
GO:0040007    8    Group B
GO:0050896    1    Group B
```

The text file has to follow a simple structure, to be processed correctly. It may contain from 2 to 3 columns in each line. The first column has to contain a GO, the second a number (0.0 to

) and the optional third column contains a text that will be written into the octagon of the corresponding GO. The columns must be separated with a tabulator character. According to the example above Group B has two GO IDs that contain different values. It is also possible to differentiate these GO IDs by coloring according to their

values. In order to color the octagon according to the value, you should select the gradient color on the next page on the color graph configuration window (see figure 6).

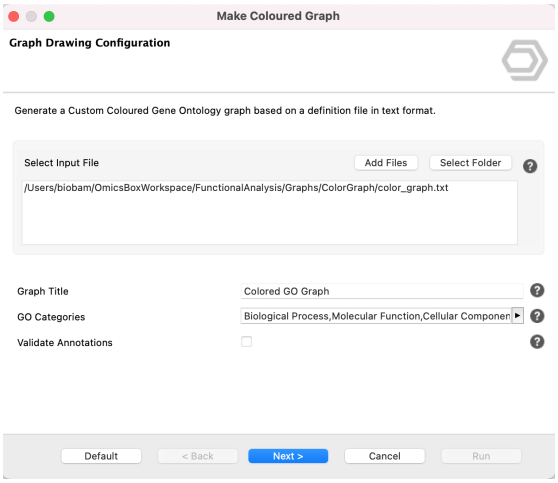


Figure 3 *Colour Configuration Window*

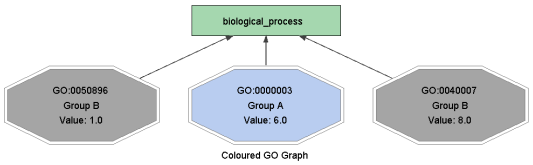


Figure 4: *Coloured GO Graph by Group*

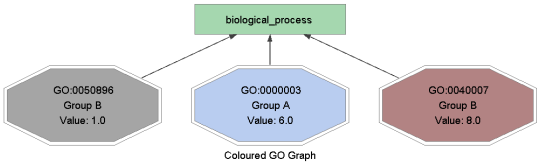


Figure 5:*Coloured GO Graph by Group value*

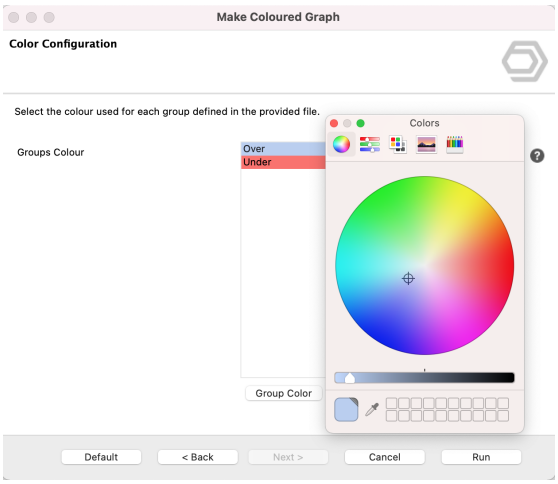


Figure 6:*Select Colour to differentiate values within the same group.*

4.5.10 Pathway Browser

Introduction

The Pathway Browser is an interactive web-based tool designed for exploring biological pathways across multiple databases. It defines search logic, queries specific terms, and visualizes detailed pathway diagrams. This tool provides access to three major pathway databases: Reactome, Gramene, and KEGG, enabling cross-database comparisons and insights.

This functionality is located under *Functional Analysis → Pathway Browser*.

Interface Overview

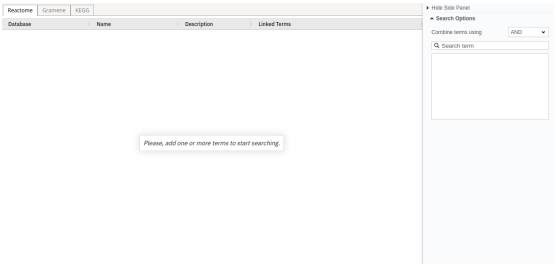


Figure 1: *Pathway Browser Overview*

SIDE PANEL WIDGETS

1. **Combine terms using:** Select the desired search logic:
    - AND: Only pathways containing all specified terms will match.
    - OR: Pathways matching at least one term will be included.
2. **Autocomplete Input:** Enter search terms (minimum 3 characters). The autocomplete feature provides suggestions based on the entered text. Wildcards (e.g., `*`) can be used at the end of the string for broader searches.
    - Press `Enter` or select a suggested term to add it to the query.

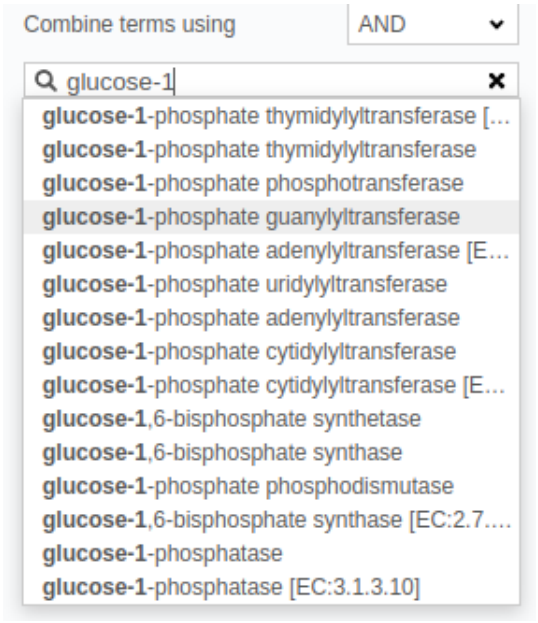


Figure 2: *Autocomplete Input*

3. **Query Term List:** Displays all selected terms in a text area-like format. Each term can be removed by clicking the “X” icon next to it. Adjusting the terms dynamically updates the search results.

RESULTS TABLE

After submitting a query, the search runs independently across Reactome, Gramene, and KEGG. The results for each database appear in separate tabs within the results table. Pathways matching the query are listed with their respective details.

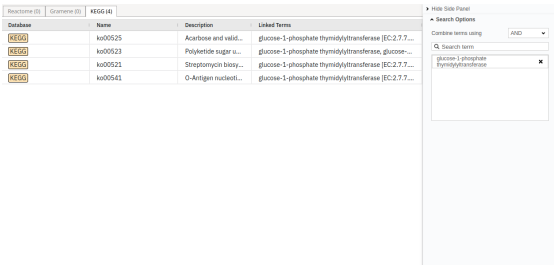


Figure 2: *Results table. The total number of matches for each database is displayed in the tab title.*

CONTEXT MENU AND DIAGRAM VIEWER

In the results table, right-clicking a row opens a context menu with the option to view the pathway diagram ("Show pathway diagram"). Selecting this option loads a detailed diagram of the pathway, enriched with the terms included in the search query. These terms act as a "dictionary" to facilitate contextual analysis.

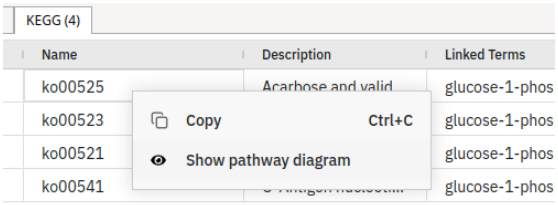


Figure 3: *Context Menu.*

Diagram Viewer

The Diagram Viewer in the Pathway Browser offers functionality similar to the Pathway Viewer found in the Combined Pathway Analysis tool). However, instead of displaying only terms associated with sequences in a project, it shows all the terms found within the pathway. This comprehensive view allows for a broader analysis of pathways, highlighting the presence of all relevant terms identified in the search query.

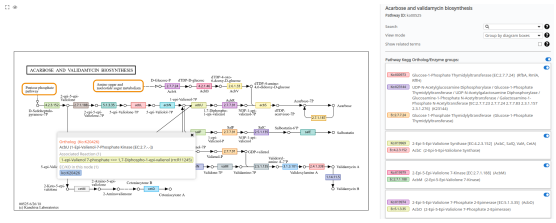
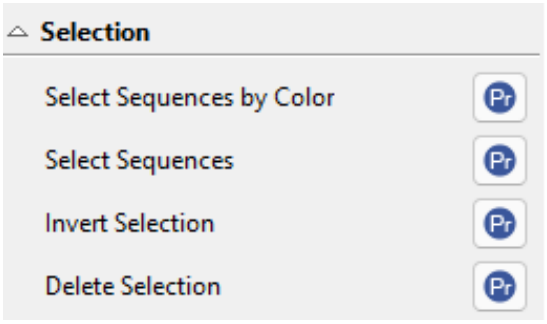


Figure 4: *Diagram Viewer.*

4.5.11 Selection



Selection options

There are different functions for selecting and deselecting sequences. Most functions in OmicsBox are only applied to selected elements. Selections allow to create subset or apply certain functions to parts of a given dataset.

SELECT SEQUENCE BY COLOR

This function allows (de)selection of sequences on the basis of their color code i.e. the processing stage they have.

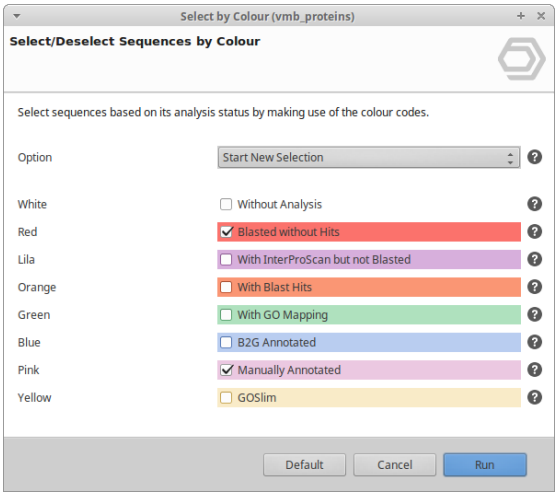


Figure 1: *Selection by Color*

SELECT SEQUENCES

The Select Sequences feature can be applied to OmicsBox Projects only and allows to select sequences for many different criteria. Selections can by added to existing ones, subtracted or created from scratch.

- Sequence Name. This is a general function for (de)selecting sequences by loading a file containing a list sequence IDs.
- Sequence Description. OmicsBox allows to (de)select sequences according to Blast result description.
- Species.
- Function (GO-Terms or GO-IDs). This is a general function for (de)selecting sequences by loading a file containing a list of GO-Terms or GO-IDs.
- InterProScan IDs.
- Enzyme IDs.

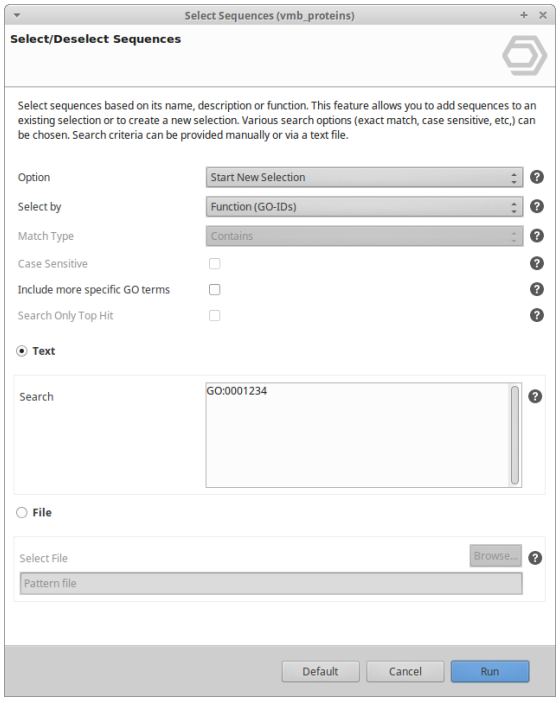


Figure 2: *Start New Selection*

INVERT SELECTION

This function will invert the current selection. Those sequences that are not selected will now be selected and vice versa.

DELETE SELECTED SEQUENCES

This function will delete selected sequences from the Main Sequence Table

OTHER SELECT OPTIONS

Extract Selection to New Tab

One can extract a subset of the selected sequences to a new project.

Once one has the desired sequences selected it is possible to hide/filter out the deselected ones by clicking on the icon next to the selection check box on the table.

With Ctrl+A in Windows/Linux or appleKey+A on Mac OS, all selected sequences will be marked. Now right click on one of the sequences on the table and choose the 2nd option Extract Selection to New Tab.

A new project will be created of the selected sequences.

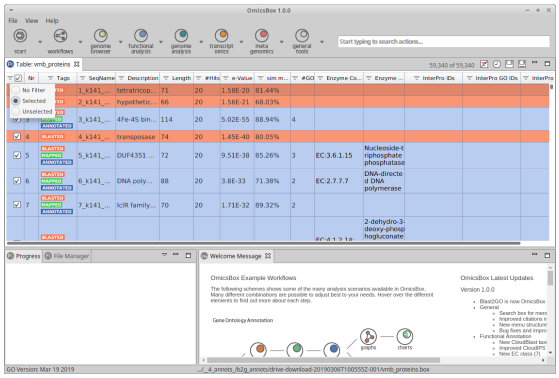


Figure 3: *Show only selected sequences*

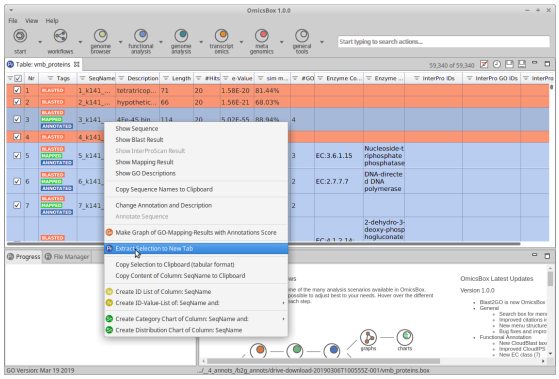


Figure 4: *Extract Selection to New Tab*

4.5.12 Tools

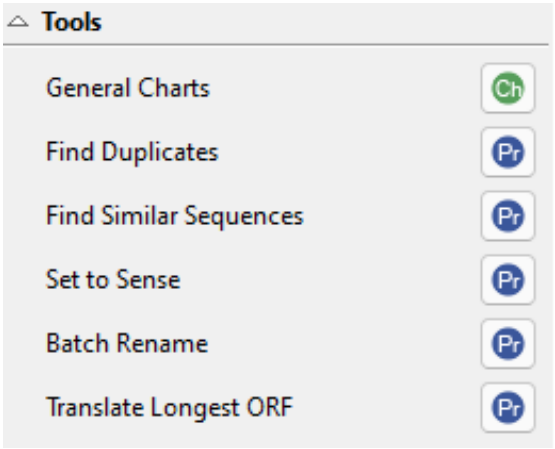


Figure 1: *Tools options*

GENERAL CHARTS

It is possible to generate different statistic charts related to the sequence project and also to understand the progress of the analysis (figure 3, figure 4 and figure 5).

- **Data distribution bar chart:** Bar chart showing the number of sequences with Blast (with or without hits), GO Mapping and GO Annotation results.
- **Data distribution pie chart:** Pie chart showing the number of sequences with Blast (with or without hits), GO Mapping and GO Annotation results.
- **Analysis Progress:** Bar chart showing the cumulative number of sequences with Blast hits, InterProScan, GO Mapping and GO Annotation results.
- **Sequence Length Distribution:** Area chart showing the number of sequences for each sequence length.

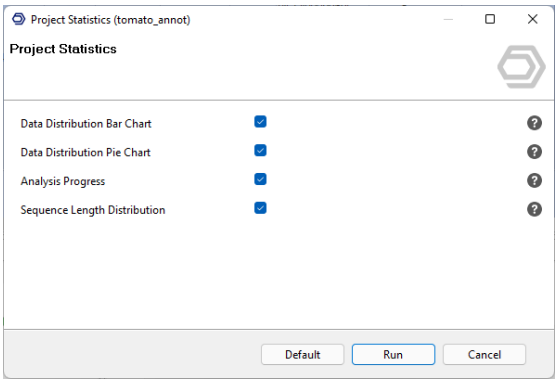


Figure 2: *Project Statistics*

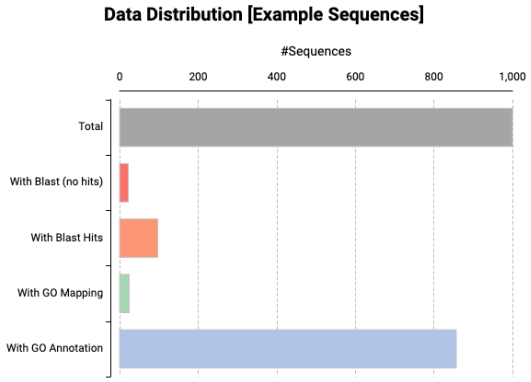


Figure 3: *Data Distribution Bar Chart*

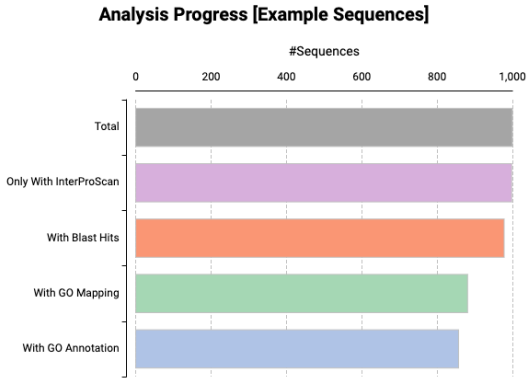


Figure 4: *Analysis Progress*

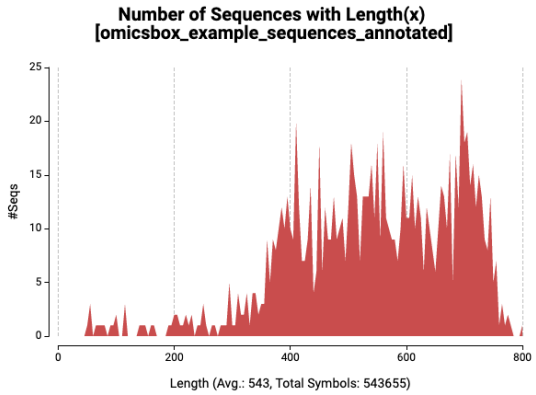


Figure 5: *Sequence Length*

FIND DUPLICATED SEQUENCES

This function allows to quickly identify and remove redundant sequences (exactly the same sequences) within a dataset.

It is possible to select mark as selected or directly remove or Create an ID-List of all sequences in the dataset which have the exact same sequence string.

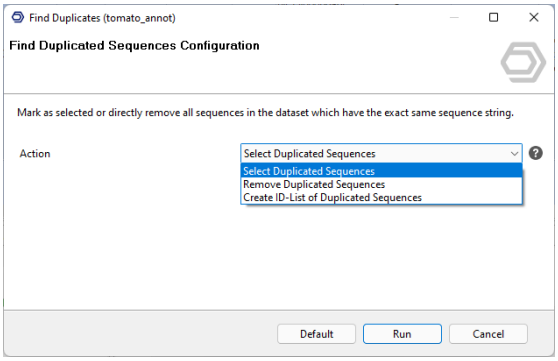


Figure 6: *Find Duplicated Sequences wizard*

FIND SIMILAR SEQUENCES

This function allows searching for similar sequences within a dataset. The search for similar sequences is done via BLAT alignments. The function searches a list of sequences against itself and reports all alignments above a certain similarity percentage. It is possible to remove similar sequences from the project or remove or to extract a less redundant result dataset into a new project.

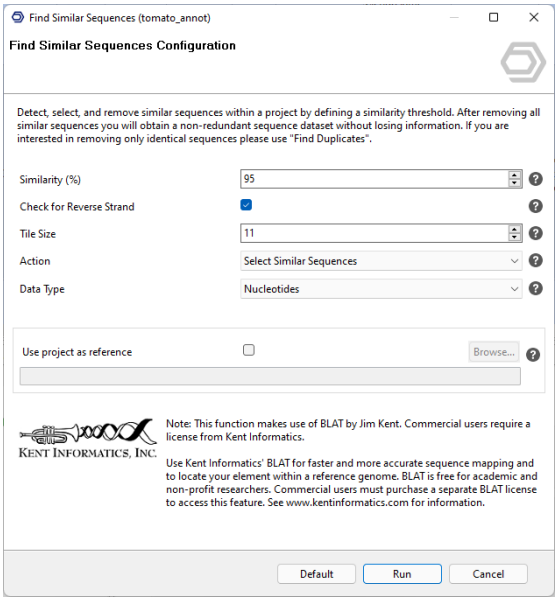


Figure 7: *Find Similar Sequences wizard*

SET TO SENSE (BASED ON BEST-BLAST-HIT)

Convert all selected sequences with a negative reading frame Best-Blast-Hit to anti-sense i.e. query-sequences will be translated to its reverse complement (e.g.: ATTG ->CAAT). The tag "_antisense" will be added to the end of the sequence names. Use the batch rename function to undo the name change.

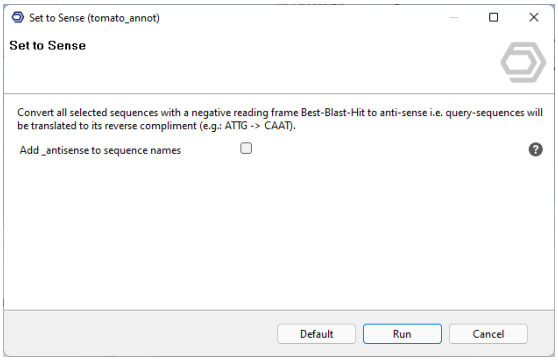


Figure 8: *Set to Sense wizard*

BATCH RENAME

Perform a batch rename of all selected sequences by converting, replacing or adding text to the actual sequence name. Link here for a detailed explanation on how to use this tool.

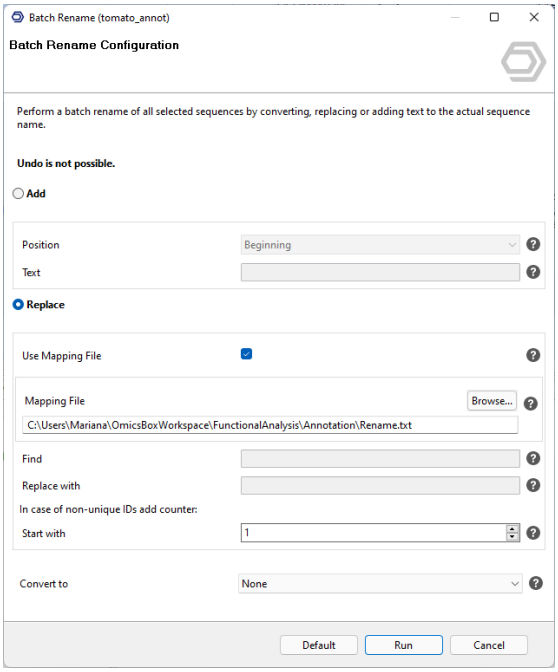


Figure 9: *Bach Rename wizard*

TRANSLATE LONGEST ORF

Convert all selected sequences to its longest ORF protein sequence. The tag "_ORF" will be added to the sequence names. Use the batch rename function to undo the name change. The user may select the reading frame, the genetic code depending to the species that will be considered to the prediction.

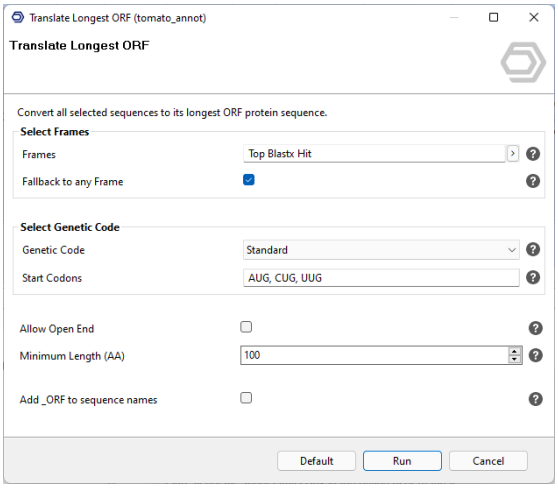


Figure 10: *Bach Rename wizard*

4.5.13 Subcellular Localization Prediction with PSORTb

Introduction

The PSORT principle uses the amino acid sequence information to generate an overall prediction of the protein localization sites. These rules are derived from experimental observations. For example, when analysing a gram-negative organism, possible localization sites are cytoplasm, cytoplasmic membrane, periplasm, outer membrane, and extracellular space.

OmicsBox allows assigning sub-cellular localization sites to proteins based on their amino acid sequence via PSORTb. PSORTb is an algorithm that can be applied to bacteria or archaea protein sequences and uses a probabilistic system to predict the most probable localization. Once sites are predicted, their corresponding cellular component GO terms can be merged with the already existing annotations.

Run

Starting with a previously loaded .box/.b2g project with PROTEIN sequences, the PSORTb tool can be found under **Functional Analysis → Subcellular Localization Prediction with PSORTb.**

If the loaded project contains nucleotide sequences, the "*Translate Longest ORF*" tool can help to obtain the predicted protein sequences and be able to run PSORTb.

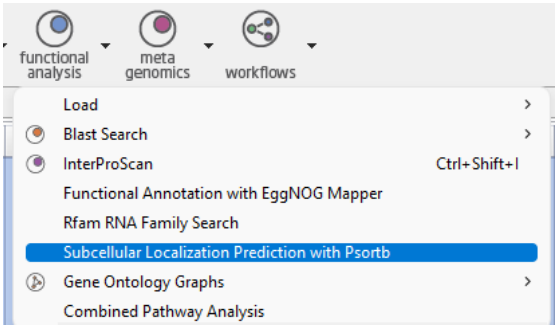


Figure 1. *Run PSORTb in the Functional Analysis menu.*

Wizard and parameters

The wizard allows adjusting the algorithm parameters (Figure 2).

It performs different analyses depending on the **Organism Type** and the **Gram Stain**. It can be used with bacteria positive and negative gram strains or archaea organism sequences. For more details of the core algorithm, visit psortb.org.

The algorithm returns score values between 0 and 10 for each localization site, the **Cutoff** parameter allows setting a minimum value of each localization above which the value can be considered as possible localization.

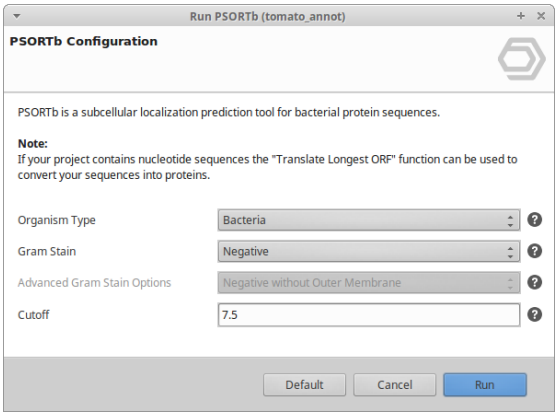


Figure 2. *PSORTb wizard where the user can adjust the parameters.*

Results

The tool will iterate over the input sequences and analyze each of them with the PSORTb 3. The process will open a new tab and as the results come back, they are shown in a table format.

The table contains one row for each sequence. The table columns are:

- **Sequence name:** shows each sequence identifier.
- **Final localization:** contains the predicted localization name.
- **Final score:** represents the prediction score for the localization.
- **GO ID:** the Gene Ontology ID associated to the location.
- **Secondary Localization:** a possible secondary localization when there is more than one score above the cutoff.
- The next 6 columns, hidden by default, show the score for all possible localizations.

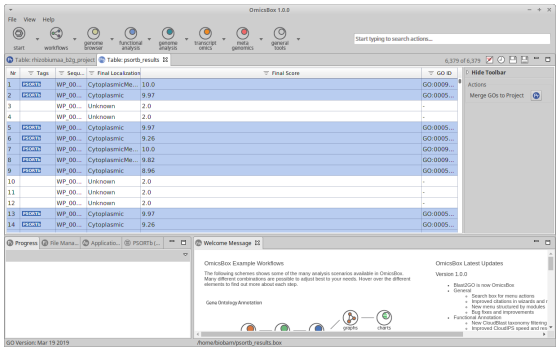


Figure 3. *PSORTb results table.*

Merge GO information

The GO IDs from the prediction can be merged into the original Blast2GO project as cellular component characterization of the sequences.

The merge option is available in the right-side panel of the PSORTb results (Figure 3).

The merge wizard asks for the OmicsBox project file where to merge the GO results and will add the GO information to the project, matching the Sequence Name. Note: The initial OmicsBox project must be saved as a file before running the Merge GOs option.

For more information regarding PSORTb, visit the psortb.org documentation page.

4.5.14 Rfam RNA Family Search

Introduction

The Rfam database is a collection of RNA families, each represented by multiple sequence alignments, consensus secondary structures and covariance models (CMs). The families in Rfam break down into three broad functional classes: non-coding RNA genes, structured cis-regulatory elements and self-splicing RNAs. Typically these functional RNAs often have a conserved secondary structure which may be better preserved than the RNA sequence. The CMs used to describe each family are a slightly more complicated relative of the profile hidden Markov models (HMMs) used by Pfam. CMs can simultaneously model RNA sequence and the structure in an elegant and accurate fashion (Rfam description from: http://rfam.xfam.org/).

Please cite: Nawrocki, E. P., Burge, S. W., Bateman, A., Daub, J., Eberhardt, R. Y., Eddy, S. R., Floden, E. W., Gardner, P. P., Jones, T. A., Tate, J., et al. (2014). Rfam 12.0: updates to the rna families database. Nucleic acids research, page gku1063.

This functionality can be found under *Functional Analysis → Coding Potential → Run Rfam*. A dialog screen appears (see image below). Sequences longer than a given length can be skipped during the analysis.
Click on the *Run* button to start the analysis. It may take a while depending on the number of sequences and the EMBL-EBI servers.

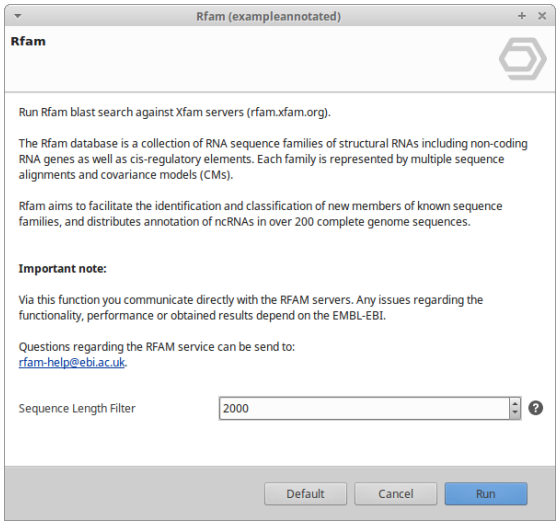


Figure 1: *Rfam Dialog*

Results Table

Once Rfam analysis has begun a table with the corresponding results will be displayed in a new tab. Sequences will turn red/orange depending if Rfam found hits for them (red if no hits were found, orange otherwise). White rows are sequences that have not been analysed yet. For each sequence It is possible to consult details about each one of their hits using the context menu (similar to consult Blast results).

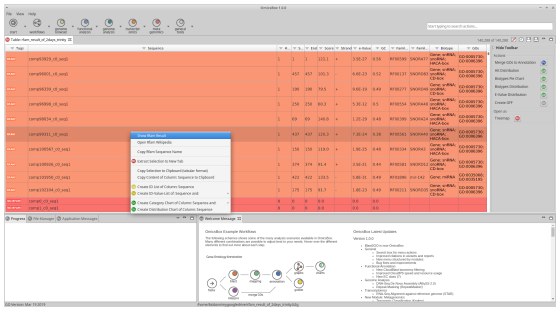


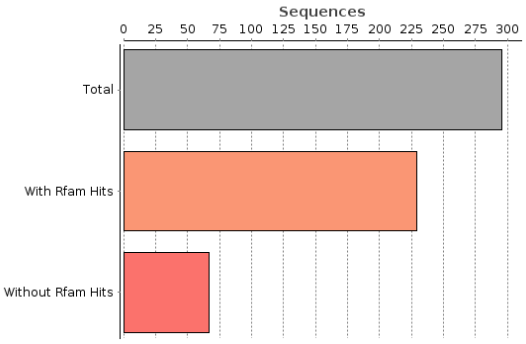
Figure 2: *Rfam Table Results*

SIDEBAR OPTIONS

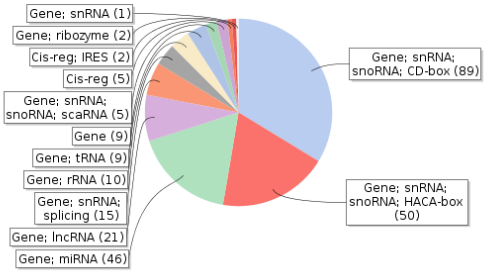
In the sidebar there are located all possible action that can be performed for the Rfam result, including one option for the visual display of the results:

1. **Hit Distribution:** This chart shows a distribution chart of the number sequences with hits in the Rfam analysis.
2. **Biotypes Pie Chart:** This pie chart shows the distribution of the Rfam families of the sequences.
3. **Biotypes Distribution:** The same as the former but in a bar-style.
4. **E-Value Distribution:** This chart plots the distribution of E-values for the Rfam hits.
5. **Create GFF:** This will create a GFF file for the Rfam results.

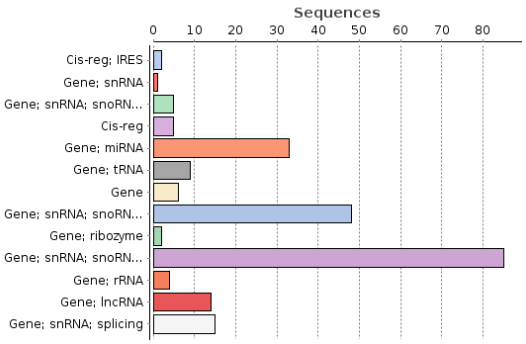
Rfam Hit Distribution



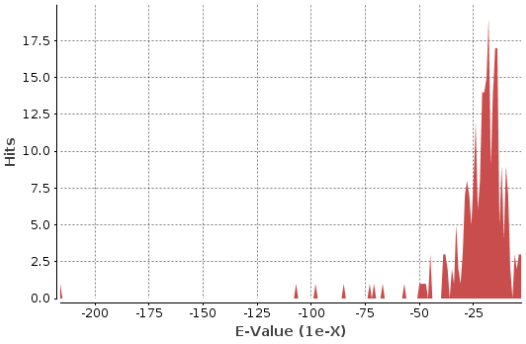
Rfam Biotypes Distribution



Rfam Biotypes Sequence Distribution



Rfam E-Value Distribution



4.5.15 Export

The following export options are available in the side panel.

	Description
Export Table	Export the current Main Sequence Table for the selected sequences.
Generic Export	This option allows you to export all the desired information to a text file.
Export as Fasta File	Export sequences of this project in fasta format.
Export GFF	Export the annotations of this project as GFF file.
Export Blast Top-Hits	It will export the best-blast-hit for each sequence, this is the hit with the lowest e-value.
Export Mapping Results	Allows to export all the information obtained and used during the Gene Ontology mapping process as GFF formatted text file.
Export Annotations	Allows to export all the information obtained and used during the Gene Ontology annotation process in different file formats, such as .annot, GeneSpring, GOstats, WEGO, GAF, GO Propagation and Gene Sets.

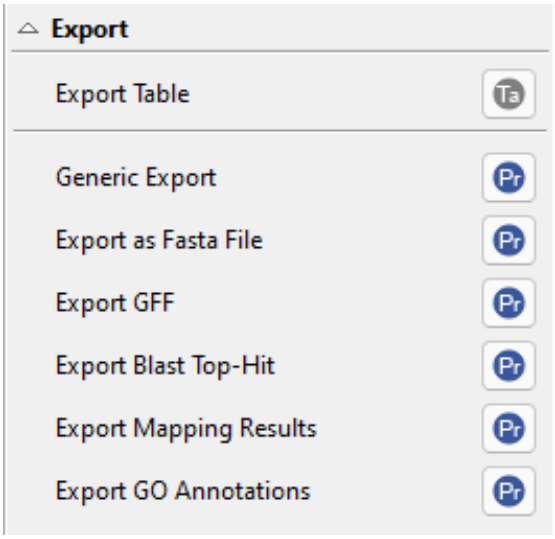


Figure 1: *Export side panel*

4.5.16 Quick Start

Functional Annotation Analysis Steps

This section provides a quick run-through of a basic functional annotation process done within OmicsBox. More detailed descriptions of the different analysis steps and more advanced features are described in the remaining sections of this documentation.

LOAD DATA

Go to File **Load > Load Sequences > Load Fasta File** and select your *.fasta* file containing the set of sequences in FASTA format. Alternatively, you can load the example sequences into OmicsBox by choosing**Load > Load > Load Example Sequences**. Please download example files to try and test OmicsBox: b2g_example_files.zip

BLAST

Click on **Blast** from the **Side Panel.** In the Blast Configuration Dialog (BLAST) select the way in which Blast will be executed (CloudBlast, NCBI Blast or Local Blast), the type of Blast mode which is appropriate for your sequence type (Blastx for nucleotide and Blastp for protein data) and the taxonomies you want to blast against. Click Next for the advanced settings and to choose where to save the Blast results, and click Run to start the Blast search.

1. Once your BLAST analysis is finished visualize your results from the **Side Panel > Charts**.
2. On the Main Sequence Table, right-click on a sequence to open the Single Sequence Menu (BLAST). Select Show BLAST Result to the BLAST Browser for that sequence.

If you are running blast using CloudBlast we recommend to run blastx-fast or blastp-fast as it is faster and fewer computation units will be consumed.

INTERPROSCAN

By clicking on the **InterPro** icon on the **Side Panel** the corresponding Wizard will be shown. If InterProScan is executed via the EMBL-EBI web service, please provide a valid email address. This is not needed if InterProScan is run via CloudIPS. It is highly recommended to run IPS in order to improve the quality of the annotations. Once InterProScan results are retrieved using **Merge GOs** to add GO terms obtained through motifs/domains to the current annotations. InterProScan can be run in parallel with BLAST.

MAPPING

Click on **Run GO Mapping** from the **Side Panel** to open to start mapping GO terms. Mapped sequences will turn **green**. Once Mapping is completed visualize your results at the **Side Panel > Charts.**

ANNOTATION

Click on **Run GO Annotation**on the **Side Panel** to open the Annotation Configuration Window. Click Next to change the evidence codes and finally click Run to start the annotation. Annotated sequences will turn **blue**.

1. Once the annotation is completed you are able to visualize your results with **Charts**.
2. On the Main Sequence Table, right-click on a sequence to open the Single Sequence Menu. Select **GO-Mapping Graph with Annotation Score** to visualize the annotation on the GO DAG for that sequence.
3. If desired, modify the annotation by clicking the right mouse button and selecting **Change Annotation and Description**or reducing it to a GO-Slim representation **Run GO- Slim** from the **Side Panel** under the **Functional Analysis** menu.
4. During the annotation process, Enzyme Codes (EC) will be also given when a GO-term/EC number equivalence is available.

ENRICHMENT ANALYSIS

OmicsBox provides tools for the statistical analysis of GO term frequency differences between two sets of sequences. On the **Side Panel** go to **Functional Analysis > Enrichment Analysis** and a new Dialog window will open to choose between the Fisher's Exact Test or GSEA. Select a *.txt* file or an ID list containing the sequence IDs for a subset of sequences. A test-set example file can be downloaded from the OmicsBox website. Select the second set of sequences as a reference set if desired. If no reference set is provided all annotations of the corresponding project will be used as the reference. Click **Run** to start the analysis. A table containing the results of this analysis will be displayed in a new tab.

1. Click on **Make Enriched Graph** icon to visualize the results of the Fisher's Test on the GO DAG.
2. Click on **Show Bar Chart** to obtain a bar chart representation of GO frequencies.
3. The results can be reduced to more specific GO terms in the corresponding icon and saved as text format (**Export Table**).

COMBINED GRAPH

OmicsBox can visualize the combined annotation for a group of sequences on the GO DAG. Select a group of sequences to generate their combined graph atthe **Side Panel** under **Selection > Select Sequences**. Now **Select by Features** and **Select by Name or ID**. You can use the Demo Test Set used previously for this. Alternatively, you can select sequences using the sequence checkboxes of the Main Sequence Table. Now on the **Side Panel** under **Functional Analysis > Combined Graph**. Now click Run.

SAVE RESULTS

File > Save saves the current OmicsBox project as .box file.

EXPORT RESULTS

- **Export** allows exporting the generated data in many different formats.
- **Export GO Annotations** exports the actual annotation results as a .annot file or generate your own formatted annotation file as .txt file.

- The enrichment analysis results can be exported in various formats from the Fisher's Exact Test Result Viewer. "Export Table" exports the results as a tabulator separated text file.
- To export GO graphs use the sidebar of the corresponding graph viewer. Graphs can be saved/exported in *.png and .txt*.

PROJECT STATISTICS

Once finished with any step or at the beginning, we can obtain a general chart that shows the state of the analysis of the entire data.
We will be able to know the number of sequences that belong to a concrete state (**Tools > General Charts**).

The data distribution can be visualized in two different charts, one as a bar chart and the other as a pie chart (Figures 1 and 2).

These are the different states we are going to find in the charts:

1. **Total:** The total amount of sequences in the project (only in the bar chart).
2. **Without Analysis:** Sequences without processing or have been reset in the BLAST menu (functional analysis > Blast > Remove Blast Results).
3. **With Only InterProScan:** Sequences that only have InterProScan and nothing else.
4. **Without Blast Hits:** Sequences that have been sent to BLAST but no hits have been found.
5. **With Blast:** Successful sequences after BLAST step or have been reset in the Mapping menu (functional analysis > Blast2GO Mapping > Remove Mapping).
6. **With Mapping:** Successful sequences after Mapping step or they have been reset in the Annotation menu (functional analysis > Blast2GO Annotation > Remove Annotation).
7. **With GO Annotation:** Successful sequences after Annotation step.
8. **With Manual Annotation:** Manually annotated sequences before or after executing the annotation step.
9. **With GO-Slim Annotation:** Sequences with GO-Slim Annotation.

Each state will have assigned a specific colour.

It is also possible to see the progress of the analysis (Figure 3).

From the 1000 sequences, 700 have blast results.
From the chart, it could suggest there are still some analyses to be completed, such as mapping, annotation and specially InterProScan.

Once all the analysis steps have been executed the Analysis Progress chart should be similar to the one in Figure 4.

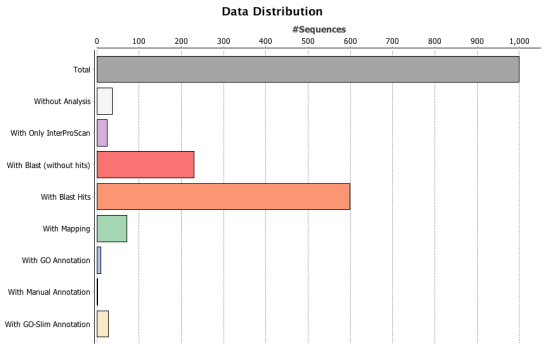


Figure 1: *Data Distribution Bar chart*

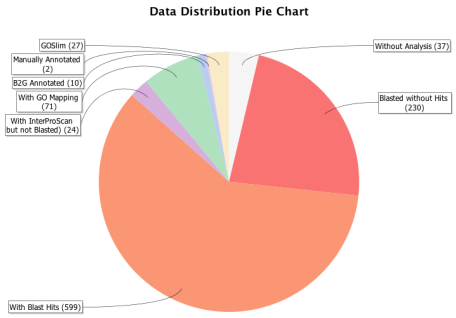


Figure 2:*Data Distribution Pie chart*

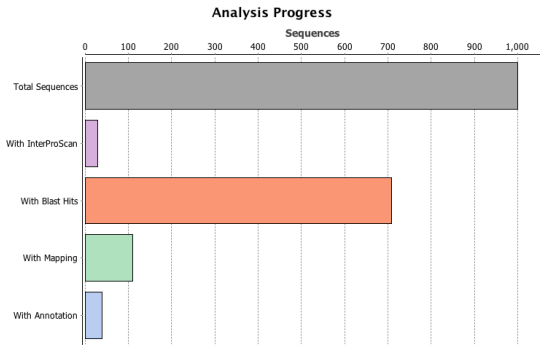


Figure 3:*Analysis Progress Chart*

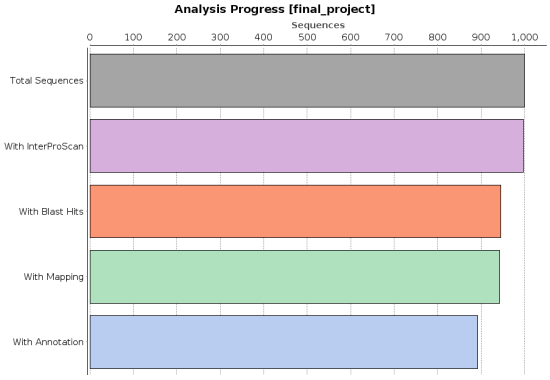
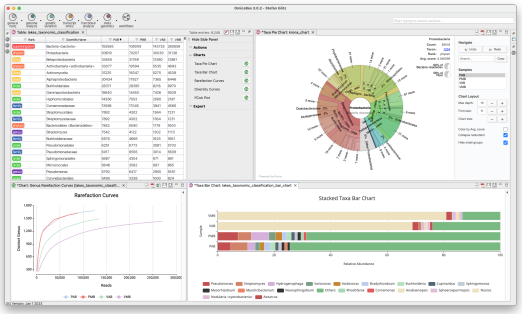


Figure 4:*Analysis Progress Chart after running all analysis*

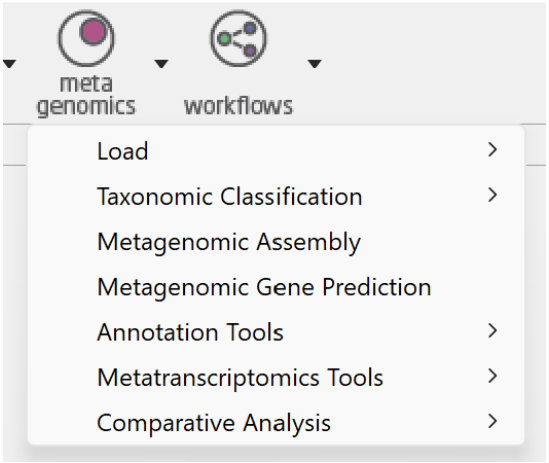
4.6 Module Metagenomics

4.6.1 Module Metagenomics



The Metagenomics Module of OmicsBox enables the seamless combination and integration of all essential steps for a comprehensive microbiome data analysis in a flexible and user-friendly manner. Users can create custom pipelines tailored to their individual analysis strategies.

- **Quality Control And Assessment**: Use FastQC and Trimmomatic for performing quality control on your samples, filtering reads, and removing low-quality bases.
- **Taxonomic Classification:** Identify Bacteria, Archaea, Fungi, Protozoa, and Viruses down to strain level with Kraken 2. Gain insights from rich visualizations and test for the differential abundance of taxa.
- **Metagenomic Assembly:** Choose between MetaSPAdes and MEGAHIT to assemble large datasets easily and fast in the cloud.
- **Gene Prediction:** Use FragGeneScan for plain reads and Prodigal for assembled data to identify and extract possible genes and proteins.
- **Functional Analysis:** Obtain high-throughput functional annotations with EggNOG-Mapper and PfamScan. Visual comparison of results is supported, along with differential abundance testing of functions.



Additional Resources

Metagenomic Analysis use case: https://www.biobam.com/metagenomic-analysis-of-two-soda-lakes-with-and-without-cyanobacterial-bloom-with-omicsbox/ .

Metagenomics Example Dataset: Download

4.6.2 Taxonomic Classification

Introduction

Traditional microbiology procedures allow us to study only about 1% of bacteria observed in natural environments, as these are the ones that can be cultured under standard laboratory conditions. This leaves a vast majority of microorganisms unexplored. Metagenomics, however, opens a new window into this unseen world. By applying sequencing techniques to DNA extracted from microbial communities in their natural habitats, metagenomics enables us to uncover the full spectrum of microorganisms and their genes present in these samples.

The primary objective of metagenomics experiments is often to identify and quantify the microorganisms present, a process known as taxonomic classification or profiling. To assess the taxonomic composition of a sample, two main strategies are employed: amplicon sequencing (16S/18S/ITS) and whole-genome sequencing (WGS). Each offers a unique perspective and set of insights into the microbial community under investigation.

AMPLICON SEQUENCING (16S/18S/ITS)

The 16S rRNA gene serves as a key reference for taxonomic identification within bacterial communities. This gene encompasses both conserved regions, which are utilized for primer design, and hypervariable regions (V1 to V9), which aid in distinguishing different taxa.

Amplification of these variable regions allows for the observation of these specific areas and the identification or quantification of a microorganism by examining this gene. There exists a variety of strategies for designing amplification primers. Some studies propose that sequencing should encompass one or more of the V2, V3, V4, V6, or V3/V4 regions, but a consensus on the most suitable hypervariable regions for analysis remains elusive. The resolution at which taxa can be detected is directly contingent on the sequencing depth and the regions selected for amplification.

The amplicon-based approach offers the primary advantage of necessitating minimal sequencing effort, thereby rendering the analysis cost-effective. However, this strategy is not without its limitations. For some bacterial species, their rRNA genes do not exhibit sufficient differences to enable clear differentiation. Furthermore, the presence of multiple rRNA gene copies in many bacterial genomes can confound species quantification results. Other factors, such as amplification biases or chimera formation, further complicate the 16S classification.

Several publicly available databases, such as GreenGenes (containing Archaeal and Bacterial 16S sequences) and Silva (comprising Archaeal, Bacterial, and Eukaryotic sequences), provide information about the DNA sequence of the 16S rRNA genes for numerous known organisms. These databases include information about both long subunits (LSU) and short subunits (SSU) of ribosomal genes. The strategy for taxonomic classification with amplicon data involves aligning the sequences to these databases. This approach, while not without its challenges, continues to be a valuable tool in the field of bioinformatics.

WHOLE GENOME SEQUENCING (WGS)

High-throughput sequencing technologies enable the sequencing of the entire genomic content of a sample's microbial community. This approach, known as whole metagenome shotgun sequencing (WGS or WMGS), generates metagenomes that encompass comprehensive genomic information.

Taxonomic classification tools for WMGS compare sequences - typically reads or assembled contigs - against a microbial genome database to determine the taxon of each sequence. In the initial stages of metagenomics, sequence alignments (e.g., BLAST) were commonly used to query reads against extensive databases (RefSeq or GenBank). However, as both the reference databases and the volume of sequencing data expanded, alignment using BLAST became computationally prohibitive. This necessitated the development of metagenomic classifiers that deliver results more rapidly while maintaining comparable sensitivity. Several strategies are available for the matching step, including:

- Aligning reads to a database of reference genomes (e.g., MEDUSA, GOTTCHA).
- Mapping k-mers (e.g., Kraken, MetaCV).
- Aligning only marker genes (e.g., MetaPhlAn).
- Translating metagenomic DNA and aligning it to protein sequences (e.g., Kaiju).

OmicsBox incorporates Kraken 2 as the preferred tool for taxonomic classification due to its advantageous features, such as compatibility with both amplicon and WGS data, and commendable benchmark performance.

Taxonomic Classification with Kraken

Kraken is a taxonomic sequence classifier that assigns taxonomic labels to short DNA reads. It accomplishes this by examining the k-mers within a read and querying a database with those k-mers. This database comprises a mapping of every k-mer in Kraken's genomic library to the lowest common ancestor (LCA) in a taxonomic tree of all genomes containing that k-mer. The set of LCA taxa corresponding to the k-mers in a read are then analyzed to assign a single taxonomic label to the read. This label can correspond to any node in the taxonomic tree. Kraken is designed for speed, sensitivity, and high precision, making it suitable for both metagenomics WGS and 16S/ITS amplicon read input data.

The current version of Kraken, Kraken 2, offers significant enhancements over Kraken 1, including faster classification speeds and reduced database sizes, enabling the inclusion of more data. To execute Kraken2, navigate to Metagenomics > Taxonomic Classification > Kraken2 (refer to Figure 1 and Figure 2). The taxa contained in each database can be visualized and explored via Taxonomic Classification > Database Info.

We currently provide access to various databases:

- NCBI RefSeq Genomes
- Silva 138.1 SSU and LSU
- The SILVA SSU database contains small subunit (16S/18S) rRNA sequences, while the SILVA LSU database contains large subunit (23S/28S) rRNA sequences.
- The SILVA SSU database is beneficial for broad taxonomic classification across all three domains of life (Bacteria, Archaea, and Eukarya), whereas the SILVA LSU database provides higher resolution for phylogenetic analysis and is particularly useful for eukaryotic sequences.
- Greengenes 13.5
- GTDB

The choice of databases depends on the nature of the input data and the specific research question at hand.

- **Database:** Choose from the target databases.
- **Sequencing Data:** Choose the type of input data: single-end, paired-end or interleaved paired-end reads If paired-end is selected, two files per sample are required and the file pattern has to be provided.
- **Reads, Contigs, or Genes:**Select files that contain the desired input data. Kraken was designed to work with short reads but works reliably with long reads, assembled sequences, or genes.
- **Paired-end configuration:** When working with paired-end libraries, a so-called pattern has to be established to help the software distinguish between upstream and downstream read files. Per default, we assume the following pattern:
- upstream: SampleA_1.fastq
- downstream: SampleA_2.fastq

Note:

For example, if the upstream file is named SRR037717_1.fastq and the downstream one SRR037717_2.fastq, you should establish "_1" as the upstream pattern and "_2" as the downstream pattern.

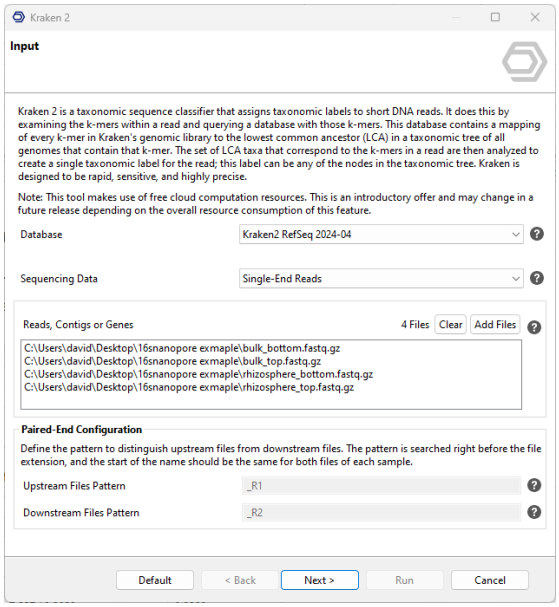


Figure 1.*Taxonomic Classification Wizard: input page*

- **Kraken Confidence Filter:** Each classified read is also assigned a confidence score between 0-1, where 1.0 is best. Reads that are classified with a lower confidence score are not taken into account and considered unclassified. Use the following table to set confidence score filtering to approximately adjust sensitivity and precision. Please find more information here http://ccb.jhu.edu/software/kraken/MANUAL.html#confidence-scoring .
- **Minimum Hit Groups:** Minimum number of hit groups (overlapping k-mers sharing the same minimizer) needed to make a call.

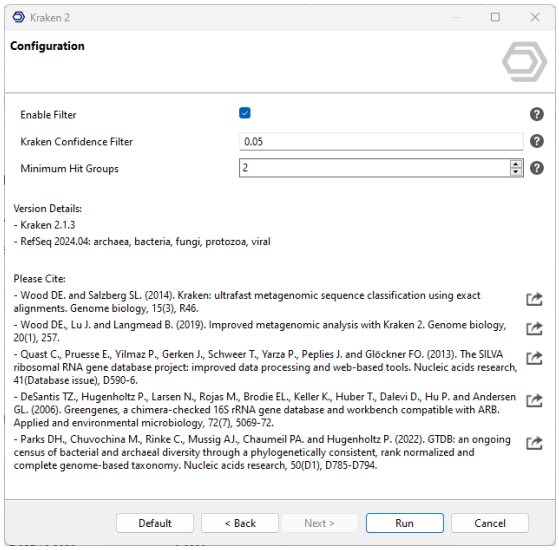


Figure 2. *Taxonomic Classification Wizard: configuration page.*

USING A CUSTOM KRAKEN2 DATABASE

This section describes how to upload and use a custom Kraken2 database for taxonomic classification in OmicsBox.

Custom Database Requirements

- The custom database is in Kraken2 format consisting of three files: hash.k2d, taxo.k2d and opts.k2d
- The custom database has been created from nucleotide sequence data. Kraken in OmicsBox does not accept protein-based databases.
- The custom database is built based on the NCBI taxonomy. Other taxonomies e.g. Silva, Green Genes, or GTDB are not supported in OmicsBox.

Please have a look at this page if you want to download alternative pre-formatted Kraken2 databases:

- https://benlangmead.github.io/aws-indexes/k2

If you want to create your own Kraken2 Database from a fasta file please follow the instructions provided by the official Kraken documentation below. This is only recommended if you know some shell scripting, know how to handle big amounts of data, and have access to a powerful Linux machine (50+GB of RAM).

- https://github.com/DerrickWood/kraken2/blob/master/docs/MANUAL.markdown#custom-databases
- https://software.cqls.oregonstate.edu/updates/docs/kraken2/MANUAL.html

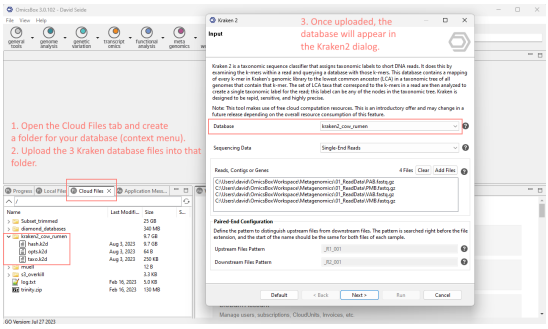
How to upload a custom database

1. **Access the Cloud Files Tab within OmicsBox:** On the left-hand side of the OmicsBox interface, you will find a panel with different tabs. Click on the "Cloud Files" tab to access your personal cloud storage.
2. **Create a New Folder:** Right-click in any location of the "Cloud Files" tab to create a new folder with a significant name at the desired location. The folder name will determine the database name in the OmicsBox user interface.
3. **Upload the Kraken2 Database Files:** Drag and drop all three database files (hash.k2d, taxo.k2d, opts.k2d) from your local computer into the newly created folder within the "Cloud Files" tab.

How to use a Custom Database

To classify taxa using your custom Kraken2 database in OmicsBox, follow these steps:

1. Open the Taxonomic Classification dialog under the Metagenomics menu.
2. In the analysis settings, choose your custom Kraken2 database from the "Database" dropdown.
3. Adjust any required parameters and settings for your classification analysis.
4. Continue with your taxonomic classification analysis in OmicsBox as usual.



Results

The results of the taxonomic classification with Kraken are:

- The main result table shows all identified taxa for each provided sample.
- PDF Report that shows overall input and carried-out analysis information.
- Stacked bar chart to compare samples at specific taxonomic levels.
- Radial cladogram in as a Krona chart to study taxa abundances in a sample.
- Rarefaction curves to assess the sequencing depth.
- Chao1 (species) diversity curve to evaluate the (species) diversity in the whole data set.
- Principal Coordinates Analysis plot to get an overview and to identify outliers.

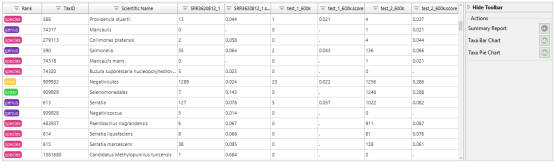


Figure 4. *Taxonomic Classification results table.*

MAIN RESULT TABLE

The primary result table (refer to Figure 4) presents the number of taxa identified and their corresponding confidence scores for each analyzed sample. A count of zero indicates the absence of the taxon in the given sample. Confidence scores range from 0 to 1, with 1 indicating the highest level of confidence. The displayed counts are cumulative, signifying that they aggregate not only the direct hits for a specific taxon but also the taxa that are hierarchically linked to it in the taxonomic tree. We utilize

a simplified form of the NCBI taxonomic hierarchy, which comprises eight main levels as opposed to the original 33, thereby summarizing numerous levels (as depicted in Figure 5).

The result table offers a filtering function via the column header, enabling users to display taxa at specific taxonomic levels, such as species or phylum.

Right-clicking on a taxon reveals a context menu for the table, providing options to generate statistics and ID lists, among other functionalities. The **Extract Sequences** option facilitates the export of read names and actual reads of the currently selected taxa. This feature enables the extraction of all reads classified under a particular category, such as bacteria, thereby reducing the dataset size for subsequent gene finding and functional annotation tasks. This functionality can be applied in various scenarios and use cases.

Figure 5. *NCBI taxonomic hierarchy.*

ADD, REMOVE, AND RENAME SAMPLES

The OmicsBox software provides the functionality to integrate different taxonomic classification results. This can be achieved by selecting the **Add Samples** option from the side panel. Upon selection, a dialog box will appear, allowing the user to browse through taxonomic classification results and choose the samples to be added. Consequently, a new object combining all the selected samples will be generated.

The software also allows for the removal or renaming of samples. This can be accomplished by right-clicking on the desired column or sample.

Please note that all actions performed via the side panel, such as generating reports or bar charts, must be executed anew to incorporate these modifications.

Critical Note: It is imperative to ensure that only samples obtained from the same target database or those utilizing the same taxa ID - scientific name relationship are combined. Failure to adhere to this guideline may result in data inconsistency, thereby compromising the accuracy of subsequent analyses or visualizations.

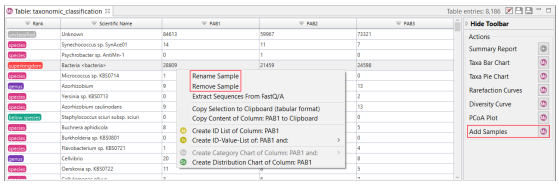


Figure 6a. *Add, Remove, and Rename Samples.*

STACKED BAR CHART

The Stacked bar chart (Figure 6b) is a combined view for inter-sample comparison, separated into the 7 main taxonomic levels. Average taxa are ordered by abundance from high to low. Only the 500 biggest taxa are shown for each sample, the remaining are gathered into an extra group called Others. These low frequent taxa can be analyzed in detail with the Krona Pie Chart. The button **Hide Unclassified** in the top-right corner shows how the percentages change when only taking into account the data that could be classified by Kraken.

The graphic can be exported as a PNG image by clicking the corresponding icon in the top-right corner.

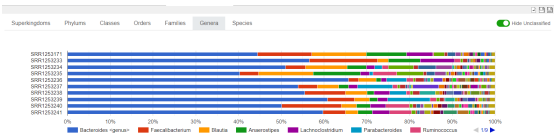


Figure 6b. *Stacked bar chart.*

KRONA PIE CHART

This graphic (Figure 7) shows a slightly modified Krona chart with various options in the side panel. Again, the counts are cumulative and grouped into roughly 8 main levels. However, all direct counts are shown as well, which is helpful when looking at the "below species" level, which includes subspecies and strains.

The currently visualized sample is selected from a list in the side panel, the **All Combined** entry shows all samples together in one chart. Furthermore, text sizes can be adjusted and OTUs can be searched. Coloring by average Kraken evidence scores is also possible.

The graphic can be exported as a PNG image and PDF by clicking the corresponding icons in the top-right corner.

SUMMARY REPORT

A summary report which shows basic statistics and alpha-diversity indices for each analyzed sample. It also gives information about the percentages of reads that were classified. In addition, for each of the 7 main taxonomic levels (Superkingdom, Phylum, Class, Order, Family, Genus, and Species), the top 10 OTUs per sample are listed.

The graphic can be exported as PDF by clicking the corresponding icon in the top-right corner.

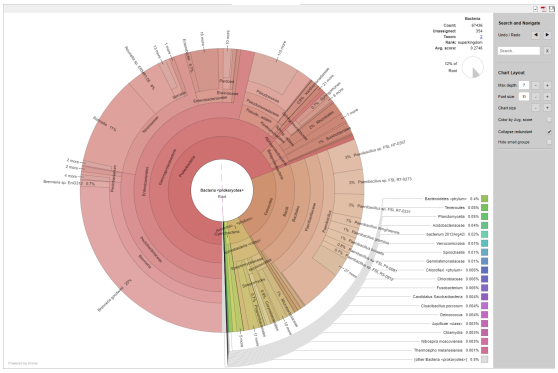


Figure 7. *Krona pie chart.*

In addition, more **charts and statistics** can be generated to offer a global visualization of the taxonomic classification results. These charts can be found in the side panel of the taxonomic classification results.

RAREFACTION CURVES

A rarefaction graph, as depicted in Figure 8, illustrates the expected number of taxa (represented on the Y-axis) discovered in n Next-Generation Sequencing (NGS) reads (represented on the X-axis).

The primary objective of rarefaction is to ascertain if the sequencing coverage is sufficiently comprehensive to provide a reliable estimate of the total taxa present within a specific sample.

If the rarefaction curve continues to exhibit an upward trend towards its end, it indicates that the sequencing coverage is insufficient to accurately represent the true microbial diversity of the sample. Conversely, if the curve approaches a horizontal asymptote, it suggests that a satisfactory estimation of diversity has been achieved.

It is important to note that the outcomes of the rarefaction technique provide an indication of the sequencing coverage but are not definitive. In other words, even if the curve approaches an asymptotic trend, there may still be rare taxa present in the sample that have not yet been observed.

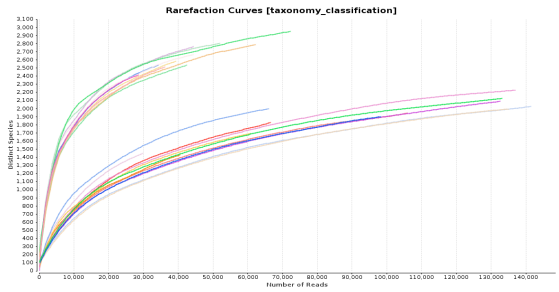


Figure 8. *Rarefaction curves.*

DIVERSITY CURVE

An accumulation or diversity curve, as illustrated in Figure 9, is a graphical representation that plots the cumulative count of unique taxa identified as a function of the number of samples examined. In other words, it displays the minimum, average, and maximum number of taxa observed when examining 1, 2, … N samples from the current dataset.

The curve provides a visual understanding of the richness and diversity of taxa within the dataset. As you move along the X-axis (number of samples), the Y-axis (number of distinct taxa) increases, indicating the accumulation of unique taxa with each additional sample.

If the curve is steep and continues to rise with the addition of more samples, it suggests that the dataset is rich in microbial diversity and that there are likely more unique taxa to be discovered with further sampling. On the other hand, if the curve begins to flatten, it indicates that most of the microbial diversity has been captured, and adding more samples may not significantly increase the number of unique taxa identified.

This curve is a valuable tool for assessing the benefits of including additional samples in the dataset. It can help determine whether the current sampling effort has sufficiently captured the microbial diversity or if more samples are needed.

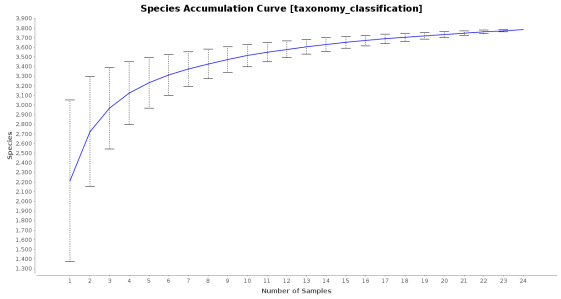


Figure 9. *Diversity curve.*

PRINCIPAL COORDINATE ANALYSIS (PCOA PLOT)

Principal Coordinate Analysis (PCoA), depicted in Figure 10, is a two-dimensional graphical representation that visualizes the Bray-Curtis distances between samples.

In the PCoA plot, each point corresponds to a sample, and the spatial proximity between points reflects the Bray-Curtis distances between samples. In other words, samples that are similar in their taxonomic composition are located close to each other, while dissimilar samples are positioned further apart.

The PCoA plot allows for the incorporation of experimental conditions and taxonomy levels. Users can select a specific experimental condition to color the points, providing a visual means to distinguish samples based on the selected condition. This feature can be particularly useful in studies where samples are collected under different conditions, as it allows for an immediate visual assessment of the impact of these conditions on the microbial composition.

Additionally, users can choose the taxonomy level at which the distances will be calculated. This flexibility allows users to explore patterns of similarity and dissimilarity at various levels of taxonomic resolution, from broad taxonomic groups to specific species.

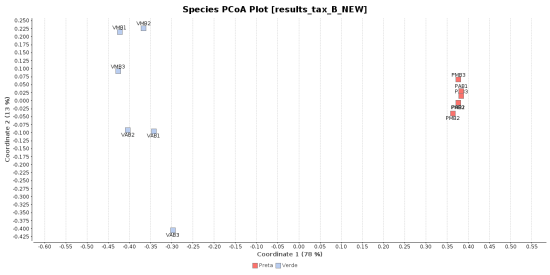


Figure 10. *PCoA plot.*

References

- Wood DE, Salzberg SL: Kraken: ultrafast metagenomic sequence classification using exact alignments.*Genome Biology* 2014, 15:R46.
- Wood DE., Lu J. and Langmead B. (2019). Improved metagenomic analysis with Kraken 2. Genome biology, 20(1), 257.
- Langmead B. and Salzberg SL. (2012). Fast gapped-read alignment with Bowtie 2. Nature methods, 9(4), 357-9.
- Ondov BD., Bergman NH. and Phillippy AM. (2011). Interactive metagenomic visualization in a Web browser. BMC bioinformatics, 12, 385.
- Quast C., Pruesse E., Yilmaz P., Gerken J., Schweer T., Yarza P., Peplies J. and Glöckner FO. (2013). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. Nucleic acids research, 41(Database issue), D590-6.
- DeSantis TZ., Hugenholtz P., Larsen N., Rojas M., Brodie EL., Keller K., Huber T., Dalevi D., Hu P. and Andersen GL. (2006). Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. Applied and environmental microbiology, 72(7), 5069-72.
- Parks DH., Chuvochina M., Rinke C., Mussig AJ., Chaumeil PA. and Hugenholtz P. (2022). GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. Nucleic acids research, 50(D1), D785-D794.

4.6.3 Contaminant Removal

Introduction

When working with host-associated studies, it is often necessary to isolate the host-related DNA from the sequencing data.

It is very likely that reads that contain host DNA can not be classified with Kraken 2 and simply increase the noise of the dataset. Mapping the read data to the host genome can help to reduce the number of unclassified reads. This process can be repeated to refine the data every time a bit more (e.g. with different phylogenetically close target genomes).

- **Sequencing Data:** Choose the type of input data: fasta, single-end, or paired-end. If paired-end is selected, two files per sample are required and the file pattern has to be provided.
- **Reads:**Select files that contain the desired input data.
- **Paired-end configuration:** When working with paired-end libraries, a so-called pattern has to be established to help the software distinguish between upstream and downstream read files. Per default, we assume the following pattern:
- upstream: SampleA_1.fastq
- downstream: SampleA_2.fastq

For SRR037717_1.fastq and SRR037717_2.fastq as up and downstream files, please select "_1" and "_2" respectively for the patterns.

- **Database Index:** Choose from one of the included target genomes (Homo sapiens, Mus musculus, PhiX, etc.), or select **Create database from genome** and provide your own genome.
- **Target Genome:** Select the target genome in Fasta format to map the selected read data against.

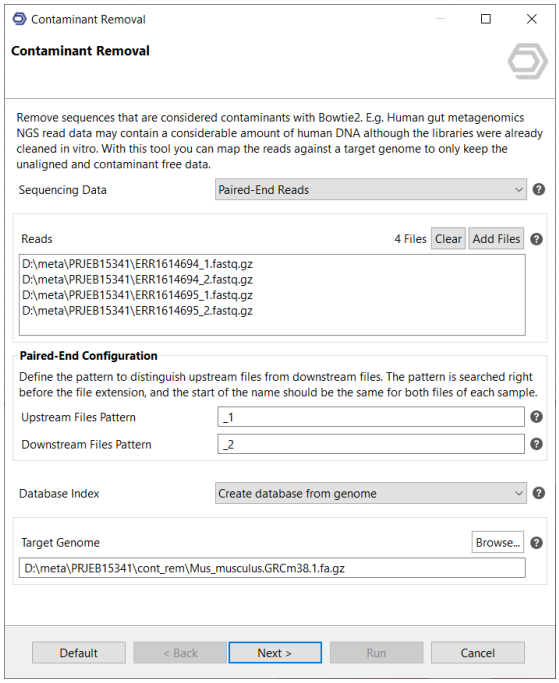


Figure 1

- **Save Results:** Choose whether to save only contaminant, non-contaminant, or both types of sequences.

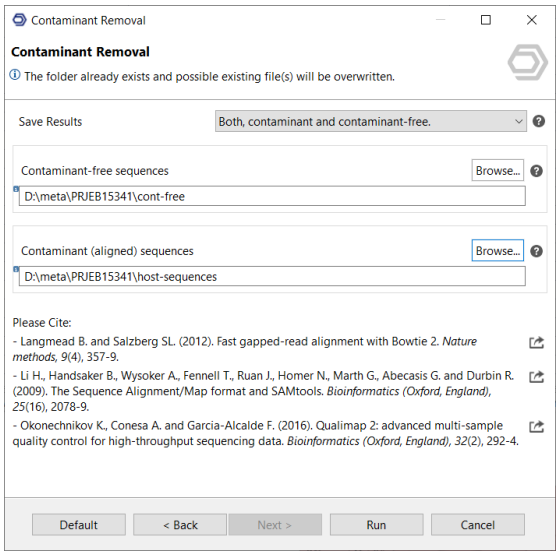


Figure 2

References

- Langmead B. and Salzberg SL. (2012). Fast gapped-read alignment with Bowtie 2. Nature methods, 9(4), 357-9.
- Li H., Handsaker B., Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G. and Durbin R. (2009). The Sequence Alignment/Map format and SAMtools. Bioinformatics (Oxford, England), 25(16), 2078-9.

• Okonechnikov K., Conesa A. and Garcia-Alcalde F. (2016). Qualimap 2: advanced multi-sample quality control for high-throughput sequencing data. Bioinformatics (Oxford, England), 32(2), 292-4.

4.6.4 Metagenome Assembly

Metagenome Assembly

In metagenomics, reads as such (typically Illumina 2 x 150 bp) are usually too short for direct functional characterization. Therefore, we offer metagenome assembly tools as a previous step to gene prediction and functional annotation.

metaSPAdes

SPAdes – St. Petersburg genome assembler – is an assembly toolkit containing various assembly pipelines. In OmicsBox, SPAdes is run with the `--meta` option, this flag is recommended when assembling metagenomic data sets (see paper for more details).

metaSPAdes (figures 1, 2, and 3) addresses various challenges of metagenomic assembly by capitalizing on computational ideas that proved to be useful in assemblies of single cells and highly polymorphic diploid genomes. Note, that SPAdes was initially designed for small genomes. It was tested on bacterial (both single-cell MDA and standard isolates), fungal, and other small genomes. Currently, metaSPAdes supports only paired-end libraries. Note that metaSPAdes might be very sensitive to the presence of the technical sequences remaining in the data (most notably adapter readthroughs), please run quality control and pre-process your data accordingly.

SPAdes is a de Bruijn graph-based assembler. Input reads are split into k-mers to create the graph and to find its Eulerian path, i.e. the shortest path that visits every edge exactly once. metaSPAdes employs a few modifications to avoid misassemblies, creating shorter high-quality contigs instead of a few long contigs.

metaSPAdes, in comparison to MEGAHIT, needs more resources and takes more time, but also creates better results, i.e. higher Nx values.

INPUT

- **Up / Downstream Reads:** Choose the files containing the paired-end reads respectively. SPAdes is not able to continue if the number of upstream reads doesn't exactly match the number of downstream reads, or if the read names differ.
- **Read Orientation:**For forward-reverse orientation, the forward reads correspond to the left reads, and the reverse reads, to the right. Similarly, in reverse-forward orientation left and right reads correspond to reverse and forward reads, respectively.

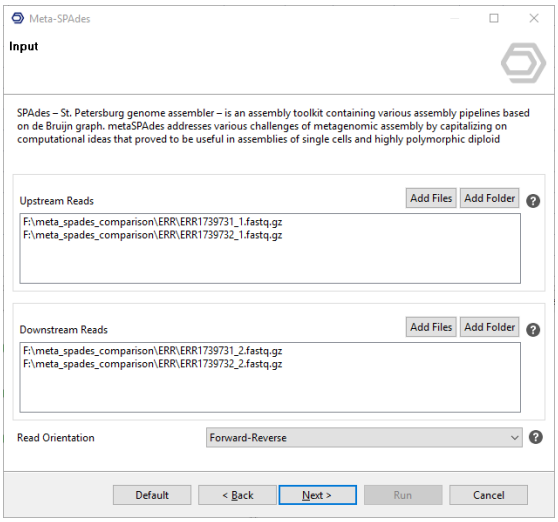


Figure 1.*MetaSPAdes assembly wizard: input page.*

CONFIGURATION

K-mer sizes:SPAdes will automatically select the k-mer sizes for graph construction. If desired otherwise, please provide a comma-separated list of odd k-mer sizes (1-128).

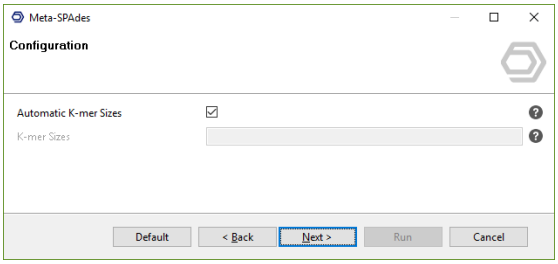


Figure 2.*MetaSPAdes assembly wizard: configuration page.*

OUTPUT

- **Contigs Fasta:** Choose where to save the resulting multi-fasta file.
- **Scaffolds Fasta:** Choose where to save the resulting file containing the scaffolds.

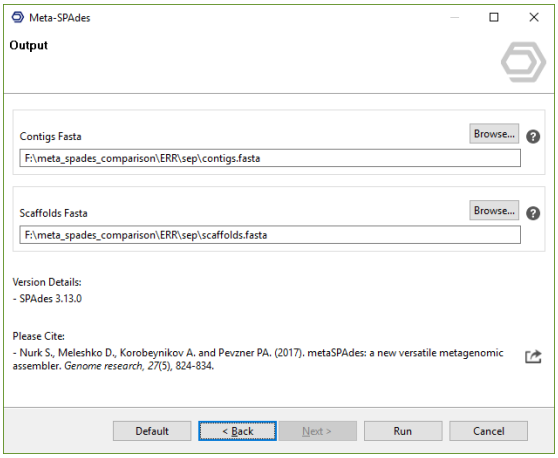


Figure 3.*MetaSPAdes assembly wizard: output page.*

The results of SPAdes are the assembled contigs and scaffolds in two separate multi Fasta files. Additionally, Quast is used to generate some basic statistics to asses the quality of the assembly, the PDF is accompanied by an Nx distribution chart.

REFERENCES

Bankevich A et al. (2012). SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. Journal of computational biology : a journal of computational molecular cell biology, 19(5), 455-77.
Li D, Liu C-M, Luo R, Sadakane K, Lam T-W. (2015). MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics. 31(10), 1674–1676,

Nurk S., Meleshko D., Korobeynikov A. and Pevzner PA. (2017). metaSPAdes: a new versatile metagenomic assembler. Genome research, 27(5), 824-834.
van der Walt AJ., van Goethem MW., Ramond JB., Makhalanyane TP., Reva O. and Cowan DA. (2017). Assembling metagenomes, one community at a time. BMC genomics, 18(1), 521.

Vollmers J., Wiegand S. and Kaster AK. (2017). Comparing and Evaluating Metagenome Assembly Tools from a Microbiologist's Perspective - Not Only Size Matters! PloS one, 12(1), e0169662.

MEGAHIT

MEGAHIT is an NGS de novo assembler for assembling large and complex metagenomics data in a time- and cost-efficient manner. MEGAHIT assembles the data as a whole, i.e. no pre-processing like partitioning and normalization is needed (figures 4, 5, and 6).

Megahit was created in the same research group that was involved in the development of SOAPdenovo and SOAPdenovo2 and may be seen as the successor of these tools. It uses a range of k-mer values for iteratively improving assemblies in a strategy adopted from the IDBA assemblers. It employs a new data structure, the "succinct de Bruijn graph", which has been designed to significantly reduce memory requirements. As an additional step to further reduce memory consumption, only k-mers occurring at a frequency above a specified cutoff are retained as "solid-k-mers", while the rest is removed as potential sequencing errors. By default, the cutoff value is 2, so k-mers occurring at least twice are kept while singleton k-mers are discarded. Because this eliminates not only sequencing errors, but also removes

information from genuinely low abundant genome fragments, a "mercy-k-mer" strategy was introduced which recovers discarded k-mers if they provide new and useful information within a trustworthy context: Discarded singleton k-mers that occur on the same read as "solid k-mers" and are needed to connect these "solid k-mers" within the de Bruin graph are recovered and added to the graph. This minimizes the loss of sequencing information while still keeping the influence of sequencing errors low.

INPUT

- **Sequencing Data:** Choose the type of input data: single-end, paired-end or interleaved paired-end reads If paired-end is selected, two files per sample are required and the file pattern has to be provided.
- **Input Reads:**Provide the files containing sequencing reads. These files are assumed to be in FASTQ / GZ format.
- **Paired-end configuration:** When working with paired-end libraries, a so-called pattern has to be established to help the software distinguish between upstream and downstream read files. Per default, we assume the following pattern:
- upstream: SampleA_1.fastq
- downstream: SampleA_2.fastq

Note:

For example, if the upstream file is named SRR037717_1.fastq and the downstream one SRR037717_2.fastq, you should establish "_1" as the upstream pattern and "_2" as the downstream pattern.

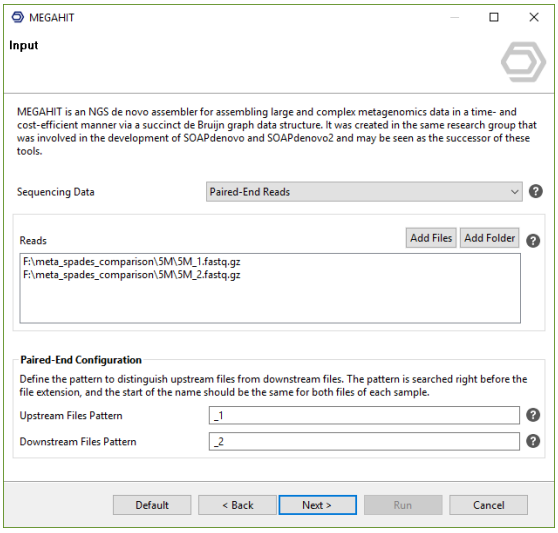


Figure 4.*MEGAHIT assembly wizard: input page.*

CONFIGURATION

- **Minimum Multiplicity:**K-mers that appear less times are filtered out. (*kmin*+1)-mer with multiplicity lower than *d* will be discarded. You should be cautious to set *d* less than 2, which will lead to a much larger and noisy graph. We recommend using the default value 2 for metagenomics assembly.
- **K-mer Sizes:** Provide a list of k-mer sizes for iterative graph creation. Values have to be odd and in the range 15-255.
- for ultra-complex metagenomics data such as soil, a larger *kmin*, say 27, is recommended to reduce the complexity of the *de Bruijn* graph. Quality trimming is also recommended.
- for high-depth generic data, large `--k-min` (25 to 31) is recommended.
- smaller `--k-step`, say 10, is more friendly to low-coverage datasets.
- **No Mercy K-mers:**Do not add mercy k-mers. Mercy k-mers are specially designed for metagenomics assembly to recover low coverage sequences. For generic dataset >= 30x, MEGAHIT may generate better results with `--no-mercy` option.
- **Bubble Level:**Intensity of bubble merging. Bubbles occur in the de Bruijn graph when several paths start in the same vertex and end in another vertex together.
- **Bubble Merge Level L:**in complex bubbles with length <= L * k-mer size are merged.
- **Bubble Merge Level S:**Complex bubbles with similarity >= S are merged.
- **Prune Level:**Strength of low depth pruning.
- **Prune Depth:**Remove unitigs with average k-mer depths less than this value.
- **Low Local Ratio:** Ratio threshold to define low local coverage contigs.
- **Max Tip Length:**Remove tips shorter than this value.

- **Disable Local Assembly:** The local assembly module was introduced in version 1.0 and creates local contigs between iterations with high confidence k-mers.

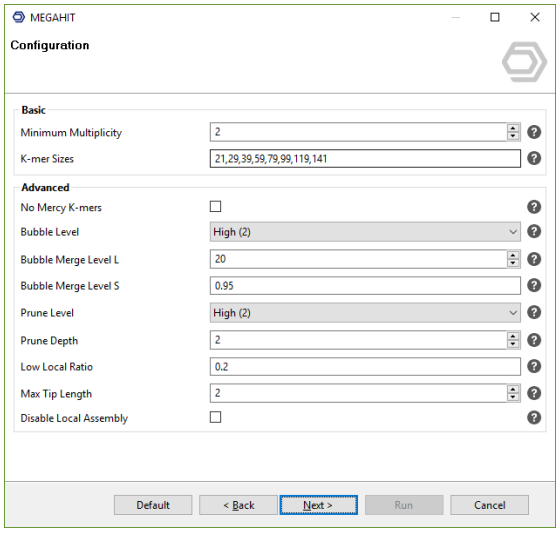


Figure 5.*MEGAHIT assembly wizard: configuration page.*

OUTPUT

- **Contig Fasta:**The final fasta file containing the assembled contigs, will be saved in this file location.

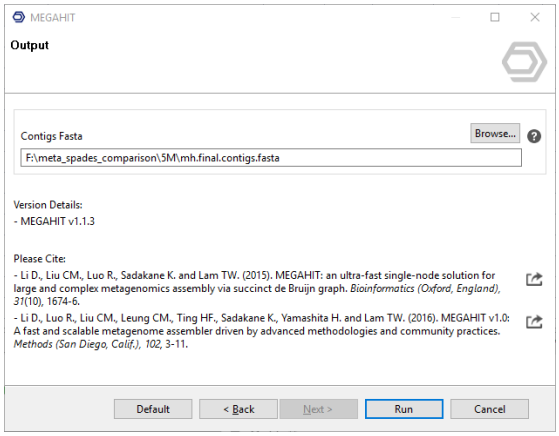


Figure 6.*MEGAHIT assembly wizard: output page.*

The results of Megahit are the assembled contigs in a multi Fasta file. Additionally, Quast is used for generating some basic statistics to assess the quality of the assembly, the PDF is accompanied by an Nx chart.

REFERENCES

Li D., Liu CM., Luo R., Sadakane K. and Lam TW. (2015). MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics (Oxford, England), 31(10), 1674-6.

van der Walt AJ., van Goethem MW., Ramond JB., Makhalanyane TP., Reva O. and Cowan DA. (2017). Assembling metagenomes, one community at a time. BMC genomics, 18(1), 521.

Vollmers J., Wiegand S. and Kaster AK. (2017). Comparing and Evaluating Metagenome Assembly Tools from a Microbiologist's Perspective - Not Only Size Matters! PloS one, 12(1), e0169662.

4.6.5 MetaGenome Gene Prediction

FragGeneScan

FragGeneScan is an application for finding (fragmented) genes in short reads. It can also be applied to predict prokaryotic genes in incomplete assemblies or complete genomes. A fundamental step in the analysis of environmental sequence information is the prediction of potential genes or open reading frames (ORFs) encoding the metabolic potential of individual cells and entire microbial communities. FragGeneScan was designed to predict intact and incomplete ORFs on short sequencing reads by combining codon usage bias, sequencing error models, and start/stop codon patterns in a hidden Markov model (HMM), to find the most likely path of hidden states from a given input sequence. It provides a promising route for gene recovery in environmental datasets with incomplete assemblies. (Figures 1, 2, and 3)

Features

- Hidden Markov Model supported approach.
- FragGeneScan can be used for gene prediction in complete genomes, assemblies, and short reads.
- Plug and use -- no need to train specific models for different datasets.
- FragGeneScan handles sequencing errors.

INPUT DATA

- **Reads, Contigs, or Scaffolds:**Select files that contain reads or assembled sequences. This tool can work with plain reads instead of contigs

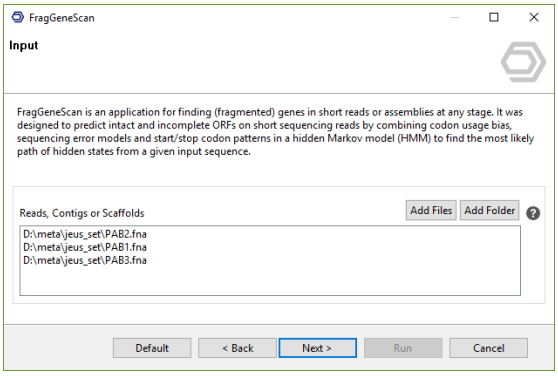


Figure 1. *FragGeneScan wizard: input page.*

CONFIGURATION

- **Type of Data:** Decide between short sequence reads or assembled sequences as input.
- **Model for Input Data:**
  [complete] for complete genomic sequences or short sequence reads without sequencing error
  [sanger_5] for Sanger sequencing reads with about 0.5% error rate
  [sanger_10] for Sanger sequencing reads with about 1% error rate
  [454_5] for 454 pyrosequencing reads with about 0.5% error rate
  [454_10] for 454 pyrosequencing reads with about a 1% error rate
  [454_30] for 454 pyrosequencing reads with about a 3% error rate
  [illumina_5] for Illumina sequencing reads with about 0.5% error rate
  [illumina_10] for Illumina sequencing reads with about a 1% error rate

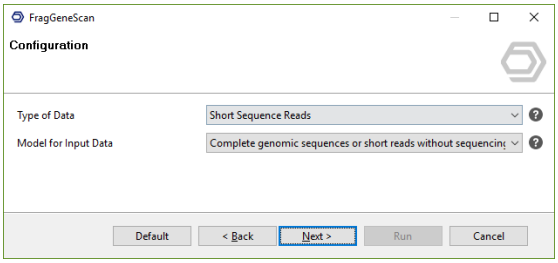


Figure 2. *FragGeneScan wizard: configuration page.*

OUTPUT

- **Nucleotide Fasta:**Select a file location for the genes multi fasta output.

- **Amino-Acid Fasta:**Select a file location for the protein sequences multi fasta output.
- **GFF:** Select a file location to save the gene feature format file.

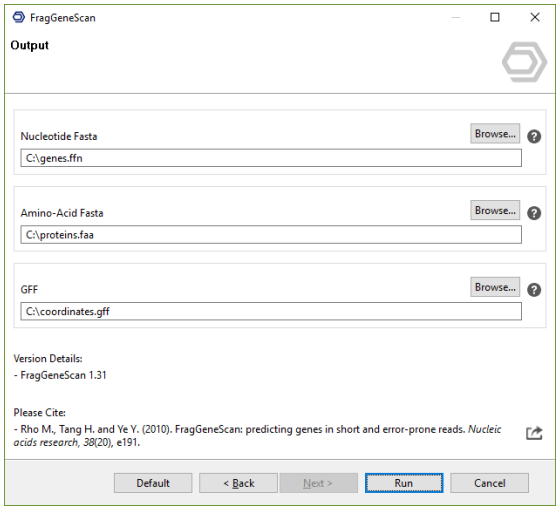


Figure 3. *FragGeneScan wizard: output page.*

REFERENCES

- Rho M., Tang H. and Ye Y. (2010). FragGeneScan: predicting genes in short and error-prone reads. Nucleic acids research, 38(20), e191.
- Trimble WL., Keegan KP., D'Souza M., Wilke A., Wilkening J., Gilbert J. and Meyer F. (2012). Short-read reading-frame predictors are not created equal: sequence error causes loss of signal. BMC bioinformatics, 13, 183

Prodigal

Fast, reliable protein-coding gene prediction for prokaryotic genomes. Prodigal's algorithm for gene prediction follows the basic principle of KISS (Keep It Simple, Stupid). Compared to other methods, Prodigal's naive log-likelihood functions seem deceptively simple. Despite its lack of complexity (no Hidden Markov Model, no Interpolated Markov Model, etc.), Prodigal nonetheless achieves good results. (Figures 4, 5, and 6)

Features

- Predicts protein-coding genes: Prodigal provides fast, accurate protein-coding gene predictions.
- Handles draft genomes and metagenomes: Prodigal runs smoothly on finished genomes, draft genomes, and metagenomes.
- Runs unsupervised: Prodigal is an unsupervised machine learning algorithm. It does not need to be provided with any training data, and instead automatically learns the properties of the genome from the sequence itself, including RBS motif usage, start codon usage, and coding statistics.
- Handles gaps and partial genes: The user can specify if Prodigal should build genes across runs of N's as well as how to handle genes at the edges of contigs.
- Identifies translation initiation sites: Prodigal predicts the correct translation initiation site for most genes and can output information about every potential start site in the genome, including confidence score, RBS motif, and much more.

INPUT

Contigs or Scaffolds: Select files that contain reads or assembled sequences.

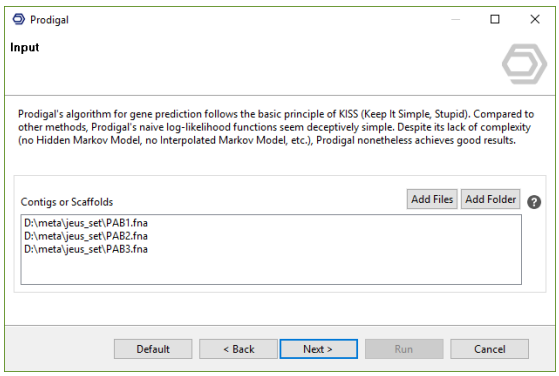


Figure 4. *Prodigal wizard: input page.*

CONFIGURATION

- **Closed Ends:**Force genes to have start and stop codon, partial genes are not reported.
- **Genetic Code:**Specify a translation table to use. "auto" will try 11 and then 4 automatically, otherwise the selected genetic code (1-25) will be used.
- **Treat Runs of N as Masked Sequence:**Tells Prodigal not to build genes around sequences of Ns.
- **Bypass Shine-Dalgarno Trainer:**Bypass Shine-Dalgarno trainer and force a full motif scan.

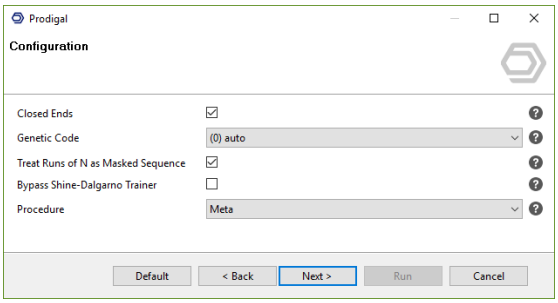


Figure 5. *Prodigal wizard: configuration page.*

OUTPUT

- **Nucleotide Fasta:**Select a file location for the genes multi fasta output.
- **Amino-Acid Fasta:**Select a file location for the protein sequences multi fasta output.
- **GFF:** Select a file location to save the gene feature format file.

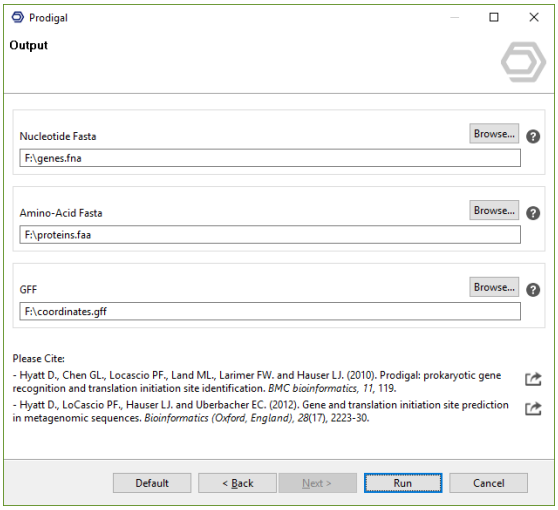


Figure 6. *Prodigal wizard: output page.*

REFERENCES

- Hyatt D., Chen GL., Locascio PF., Land ML., Larimer FW. and Hauser LJ. (2010). Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC bioinformatics, 11, 119.
- Hyatt D., LoCascio PF., Hauser LJ. and Uberbacher EC. (2012). Gene and translation initiation site prediction in metagenomic sequences. Bioinformatics (Oxford, England), 28(17), 2223-30.
- Trimble WL., Keegan KP., D'Souza M., Wilke A., Wilkening J., Gilbert J. and Meyer F. (2012). Short-read reading-frame predictors are not created equal: sequence error causes loss of signal. BMC bioinformatics, 13, 183.

4.6.6 Functional Annotation with EggNOG

EggNOG-Mapper

Eggnog-mapper is a tool for fast functional annotation of novel sequences (genes or proteins) using precomputed eggNOG-based orthology assignments. Obvious examples include the annotation of novel genomes, transcriptomes or even metagenomic gene catalogs. The use of orthology predictions for functional annotation is considered more precise than traditional homology searches, as it avoids transferring annotations from paralogs (duplicate genes with a higher chance of being involved in functional divergence). (Figures 1 and 2)

Details and methodology about the tool and its database are best explained on their website: http://eggnogdb.embl.de/#/app/methods

INPUT

- **Genes or Proteins:** A multi-fasta file containing genes or proteins.

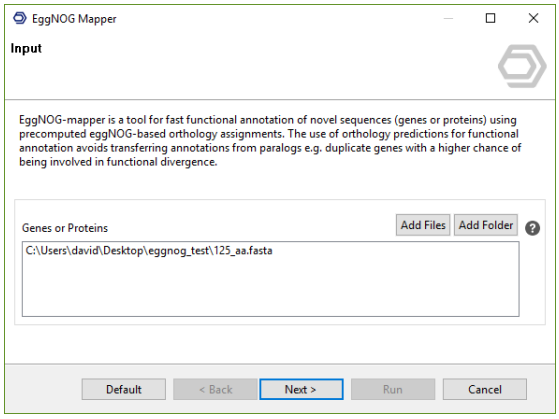


Figure 1.*EggNOG Mapper wizard: input page.*

CONFIGURATION

- **Taxonomic Scope:**Fix the taxonomic scope used for annotation, so only orthologs from a particular clade are used for functional transfer. By default, this is automatically adjusted for every query sequence.
- **Target Orthologs:** Define what type of orthologs should be used for functional transfer.
- **GO Evidence:** Defines what type of GO terms should be used for annotation:
- experimental = Use only terms inferred from experimental evidence
- non-electronic = Use only non-electronically curated terms

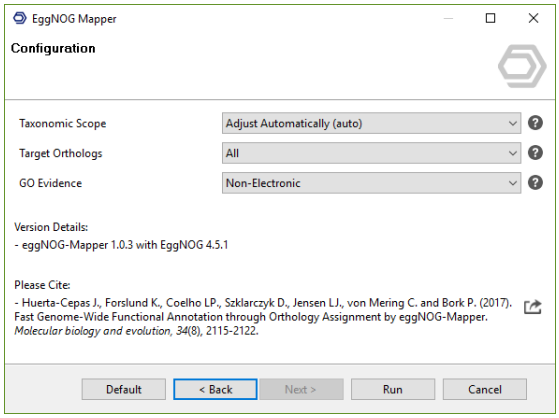


Figure 2.*EggNOG Mapper wizard: configuration page.*

RESULTS

The result table (figure 3) summarizes all annotations that could be transferred with EggNOG Mapper. Besides ordering and filtering, the context menu allows to take a closer look at certain results.

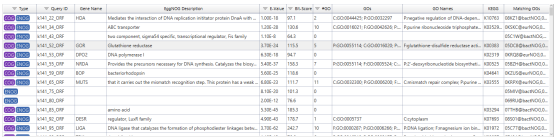


Figure 3.*EggNOG Mapper results table.*

The annotation details (figure 4) provides link-outs, where possible, and give detailed information about annotated GOs.

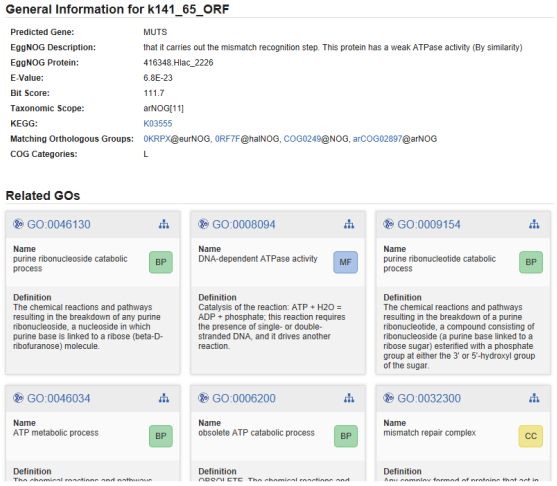


Figure 4.*EggNOG Mapper annotation details.*

REFERENCES

Fast genome-wide functional annotation through orthology assignment by eggNOG-mapper. Jaime Huerta-Cepas, Damian Szklarczyk, Lars Juhl Jensen, Christian von Mering and Peer Bork. Submitted (**2016**).

Huerta-Cepas J et al. (2019). eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. Nucleic acids research, 47(D1), D309-D314.

PfamScan

Pfam is a database of protein families. Briefly, each Pfam database entry is comprised of a seed alignment, which forms the basis to build a profile hidden Markov model (HMM) using the HMMER software (http://hmmer.org/). The profile HMM is then queried against a sequence database called *pfamseq*, and all matches scoring above the curated threshold (carefully chosen to avoid the inclusion of any known false positives), are aligned back to the profile HMM to generate the full alignment. Where possible, each entry is annotated with functional information derived from literature. To improve sustainability, especially with regard to scaling of the resource, *pfamseq* is derived only from the UniProt Knowledgebase (UniProtKB) sequences that belong to Reference Proteomes, rather than the entirety of UniProtKB. (Figure 5)

INPUT

- **Genes or Proteins:** A multi-fasta file containing genes or proteins.

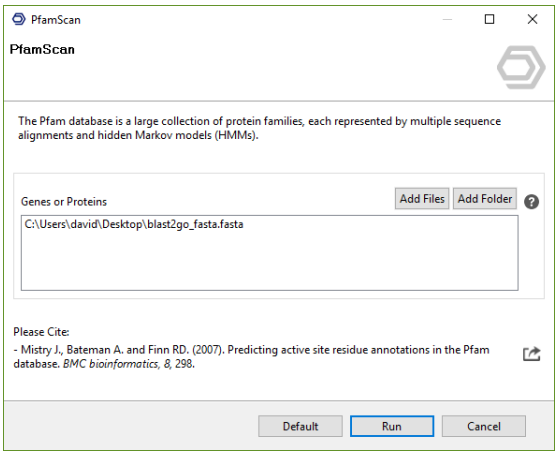


Figure 5.*PfamScan wizard.*

RESULTS

The result table (figure 6) summarizes all PfamScan annotations. Besides ordering and filtering, the context menu allows to take a closer look at certain results.

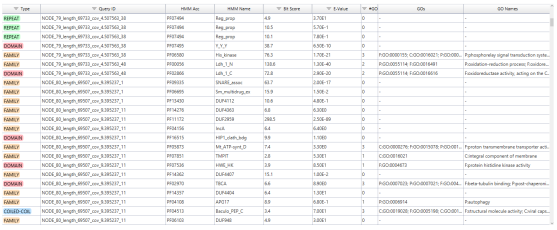


Figure 6.*PfamScan results table.*

The annotation details (figure 7) provide link-outs, where possible, and give detailed information about annotated GOs.

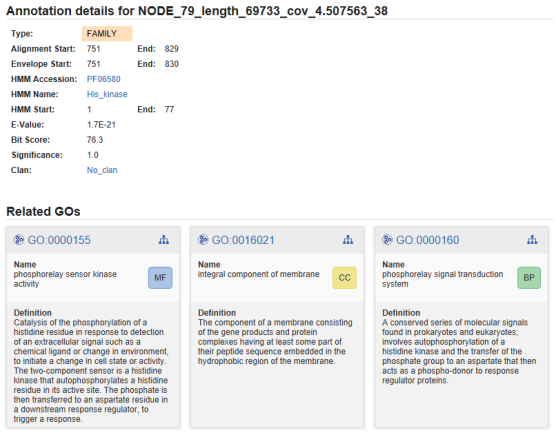


Figure 7.*PfamScan annotation details.*

REFERENCES

The Pfam protein families database in 2019: S. El-Gebali, J. Mistry, A. Bateman, S.R. Eddy, A. Luciani, S.C. Potter, M. Qureshi, L.J. Richardson, G.A. Salazar, A. Smart, E.L.L. Sonnhammer, L. Hirsh, L. Paladin, D. Piovesan, S.C.E. Tosatto, R.D. FinnNucleic Acids Research (2019) doi: 10.1093/nar/gky995

4.6.7 Comparative Analysis

Introduction

This section explains the tools for the comparison of identified OTUs and functional annotation compositions between samples.

The 2 first tools, sample comparison chart, and graph are visual and allow to compare function abundances between samples. GO Slim generalizes GO annotations to make them comparable.

OTU Differential Abundance Testing identifies over and underrepresented OTUs between samples and conditions with the help of edgeR, a Bioconductor package in R.

Sample Comparison

SAMPLE COMPARISON CHART

This feature helps to compare annotations between different samples with distribution charts. It also helps to compare GO annotations from EggNOG and PfamScan (or other tools) for the same sample. First, the different samples have to be selected. It is also possible to load external annotations through File > Load > Load Metagenomic GO Annotations and to load them here (figure 1).

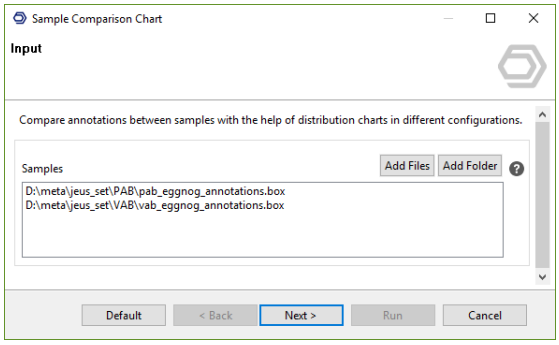


Figure 1. *Sample comparison charts: input data page.*

The second wizard page allows configuring the distribution chart (figure 2).

- **Columns to Compare:** Only annotations that exist in all selected data-sets, can be selected here.
- **Normalize Counts:** In most cases normalizing the counts between 0 and 1 gives better results, because sample sizes are seldom equal.
- **GO Categories:** Create charts for each of the 3 main GO categories.
- **Propagation of GO Terms:** The GO hierarchy is reflected in the resulting chart and helps to compare less and more specific GO annotations at higher levels.
- **GO Level Filter:** Obviously, GOs at higher levels are represented in higher numbers (if propagation is enabled). This option makes it possible to focus on specific levels.

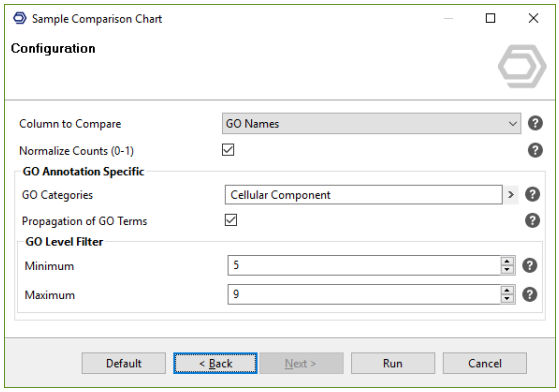


Figure 2. *Configuration page.*

On the right we can see the comparison of two different samples, both annotated with EggNOG Mapper. Cellular Component GO levels from 5 and lower are shown, ordered by maximum difference. The graphic visualizes that the red sample has major activity in intracellular parts and external encapsulating structures, while the blue sample works in different parts of the cell.

The graphic can be plotted as vertical or horizontal bars, lines, or area charts. Samples can be included or excluded, their colors can be changed, as well as their labels. The remaining options are self-explaining (figure 3).

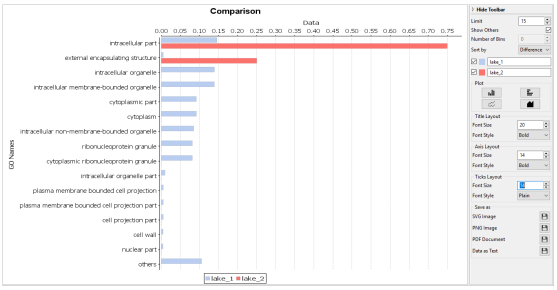


Figure 3. *Sample comparison GO chart.*

SAMPLE COMPARISON GO GRAPH

The colored GO graph on the right side visualizes the same data as above. Only GOs that appear in both samples are shown (Sample Filter = 2). The graph nodes are colored with different areas for each sample. The area's sizes depend on the relative counts (figure 4).

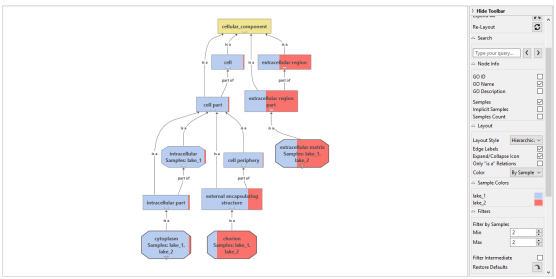


Figure 4. *Sample comparison GO graph.*

GO SLIM

GO Slim is a reduced version of the Gene Ontology that contains a selected number of relevant GOs. More specifically, GO annotations are generalized and lifted up in the hierarchy. This can be seen as a way to normalize GO annotations to simplify comparison between samples.

Differential Abundance Analysis of Taxa

The Differential Abundance Analysis of Taxa is a tool to identify Operational Taxonomic Units (OTUs) that significantly differ between two microbial communities. This feature is based on edgeR, which belongs to the Bioconductor project, and implements statistical tests to evaluate the significance of OTU abundances between contrast and a reference group.

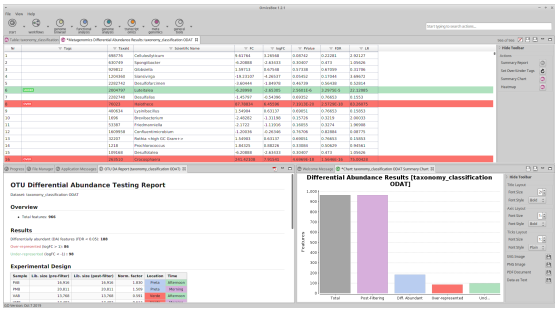


Figure 5. *Differential Abundance Testing: presentation of results.*

With a Taxonomic Classification result opened, go to **Metagenomics → Comparative Analysis → Differential Abundance Analysis of Taxa**. In the wizard, you can select the parameters to run the test. It is divided into three different sections: filtering and normalization (figure 6), experimental design (figure 8), and statistical test (figure 9).

First Wizard Page - Filtering and Normalization

OTUs with low counts will not be considered for the test as they provide little evidence of differential abundance. There are two different filtering steps:

- **Counts per Million Filter.**Set a filter to exclude OTUs with low counts across all samples. Filtering is performed on a count-per-million (CPM) basis to account for differences in library sizes between samples (e.g. a CPM of 1 corresponds to a count of 6 in a sample with 6 million total counts). Set this value to 0 if no filtering is desired.
- **Minimum Samples Filter.** Set a minimum number of samples in which the CPM has to be above the previous filter. If this value is set to e.g. 5, at least 5 of the samples have to show a count above the given CPM. The number of samples of the smallest group is usually used (e.g. in an experiment that has 2 replicates for each condition or group, an OTU should be counted in at least 2 samples). Set this value to 0 if no filtering is desired.

In this test, the normalization takes the form of scaling factors for library sizes that enter into the statistical model. These correctional factors are used to compute the effective library sizes. 5 different options are available for the normalization step:

- **TMM (Trimmed Mean of M-values).** The M-values are weighted according to inverse variances and computed by the delta method for logarithms of binomial random models.
- **TMMwsp (TMM with singleton pairing).**This is a variant of TMM that is intended to perform better for data with a high proportion of zeros (default).
- **RLE (Relative Log Expression).** Scale factors are the median ratio of each sample to the median library (geometric mean of all samples).
- **Upper-quartile.**75% quantiles for the counts of each library are used to calculate the scale factors.
- **None.** All normalization factors are set to 1.

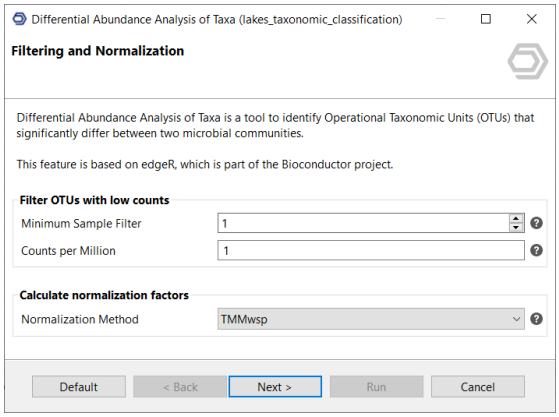


Figure 6. *Differential Abundance Testing wizard: filtering and normalization page.*

Second Wizard Page - Experimental Design

Here, the two groups for the test, reference, and contrast, have to be specified. You can select the groups by choosing which samples from the taxonomic classification project you want to include in each one, or by loading an experimental design file and selecting the conditions you want to test.

Select samples (no experimental design file loaded)

Select the samples to be considered for the test and divide them into two groups or conditions. The **Contrast Group** will be the samples that will be tested against the **Reference Group**.

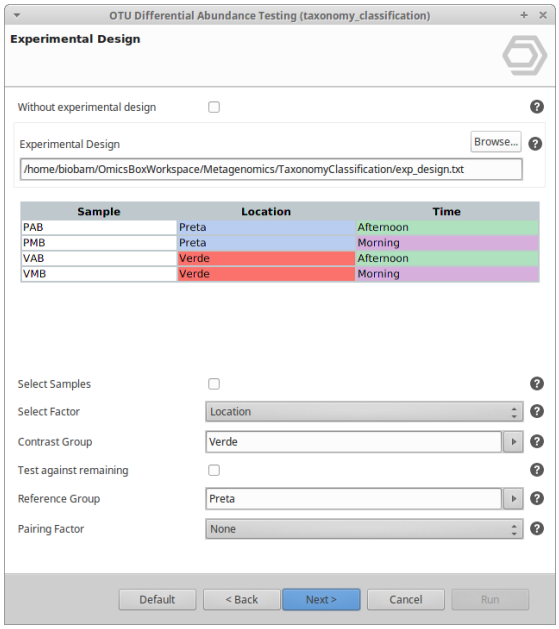


Figure 8. *Differential Abundance Testing wizard: experimental design page.*

EXPERIMENTAL DESIGN FILE

You can load your **experimental design file**. This file must contain the sample names in the first column and the experimental conditions of each sample in the following ones, as can be seen in figure 7. Please make sure the sample names in the first column of this experimental design file match exactly with the samples in the taxonomic classification result.

This experimental design file must be in **tsv format** (tab-separated values file). In this kind of files, each field is separated with a tab character. Please do not use spaces and avoid strange characters when writing your experimental design file to be sure that it will be correctly read and processed.

Once the file is properly loaded, you can select an **experimental factor** from the experimental design and the conditions to test in both, Contrast and Reference group. You can also select samples separately as described in the previous section if the **Select Samples** option is checked.

If a paired design is desired, a **Pairing Factor** from the experimental design can be optionally selected to adjust for the baseline difference of this factor. Note that this option is only available if you have provided an experimental design file.

Experimental Design

```
 Sample  Lake    Time
PAB Preta   Afternoon
 PMB Preta   Morning
VAB Verde   Afternoon
 VMB Verde   Morning
```

Figure 7. *Experimental Design file.*

Third Wizard Page - Statistical Test

You can **Test at Specific Taxonomic Levels**to only consider results for a specific taxon (species, genus, family, ...).

Here, you can select the statistical test to be used to detect the differentially abundant OTUs. The test will suppose that the OTU counts across groups are distributed as negative binomial random variables. Two different kinds of tests are available:

- **Exact Test.** Run an Exact Test to detect a difference in mean between two groups of OTU abundance libraries, reference and contrast groups. This test is performed for each OTU and can only be used if no pairing factor is selected.
- **Generalized Linear Model.** Fit a negative binomial generalized log-linear model (GLM) to the counts for each OTU. Two different GLM tests are allowed:
- **GLM Likelihood Ratio Test.** This mode conducts likelihood ratio tests for the coefficients in the linear model using the Cox-Reid dispersion estimates.
- **GLM Quasi Likelihood F-Test.** It is similar to the LRT test, except that it replaces likelihood ratio tests with empirical Bayes quasi-likelihood F tests. This test provides a more robust and reliable error rate control when the number of replicates is small.

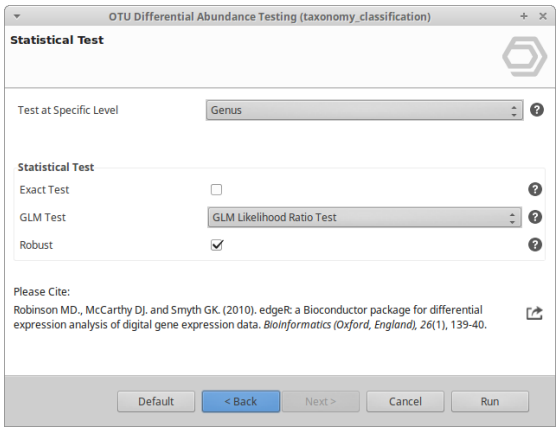


Figure 9. *Differential Abundance Testing wizard: statistical test page.*

RESULTS

Once the taxonomic abundance analysis has finished, a new **table with the results** will open (figure 10). Each row of this table corresponds to a different tested OTU. Each column contains:

- **Tags.**Indicate if a specific OTU is overrepresented -OVER- (FDR < 0.05 and logFC > 1) or underrepresented -UNDER- (FDR < 0.05 and logFC < -1) in the contrast sample.
- **FC (Fold Change).** The ratio between the mean abundance value of a specific OTU in the contrast condition and this value in the reference condition, if the mean abundance value in the contrast group is bigger than in the reference group. If this value is bigger in the reference group, then the FC is calculated as the ratio between the mean abundance value in the reference condition and the value in the contrast condition with a negative sign. By default, an OTU is defined as overrepresented if FC > 2, and it is underrepresented if FC < -2.
- **LogFC.** The log2 FC. By default, an OTU is defined as overrepresented if logFC > 1, and it is underrepresented if logFC < -1 if it is statistically significant (FDR < 0.05 by default).
- **LR (Likelihood Ratio).**Likelihood Ratio statistic for the GLM (only if GLM LR test is selected).
- **F.**Quasi-likelihood F-statistic for the GLM (only if GLM QL test is selected).
- **P-value.** The p-value for the null hypothesis of non-differential abundance.
- **FDR.**A corrected p-value for multiple testing comparisons (Benjamini Y., Hochberg Y., 1995). If meeting the logFC criterion (logFC > 1 or logFC < -1 by default), an OTU must have an FDR < 0.05 to be considered as differentially abundant.

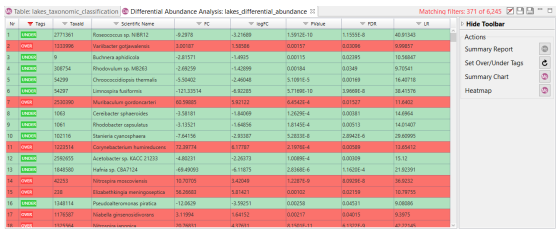
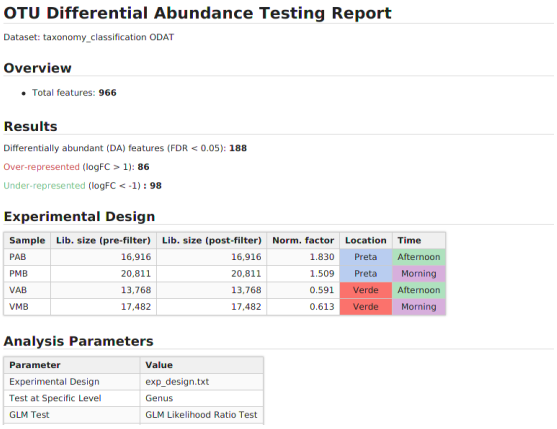


Figure 10. *OTU Differential Abundance Testing results.*

SIDE PANEL

Summary Report

Creates an HTML report which can be saved in PDF with the main results of the Differential Abundance Testing: parameters used for the test, number of differentially abundant OTUs, experimental design, ... (figure 11).



OTU Differential Abundance Testing Report

Dataset: taxonomy_classification ODAT

Overview

- Total features: **966**

Results

Differentially abundant (DA) features (FDR < 0.05): **188**

Over-represented (logFC > 1): **86**

Under-represented (logFC < -1) : **98**

Experimental Design

Sample	Lib. size (pre-filter)	Lib. size (post-filter)	Norm. factor	Location	Time
PAB	16,916	16,916	1.830	Preta	Afternoon
PMB	20,811	20,811	1.509	Preta	Morning
VAB	13,768	13,768	0.591	Verde	Afternoon
VMB	17,482	17,482	0.613	Verde	Morning

Analysis Parameters

Parameter	Value
Experimental Design	exp_design.txt
Test at Specific Level	Genus
GLM Test	GLM Likelihood Ratio Test

Figure 11. *OTU Differential Abundance Testing summary report.*

Summary Chart

Shows a bar chart with the main results: OTUs pre and post-filtering steps, OTUs which are considered as differentially abundant, and the over-/underrepresented ones (figure 12).

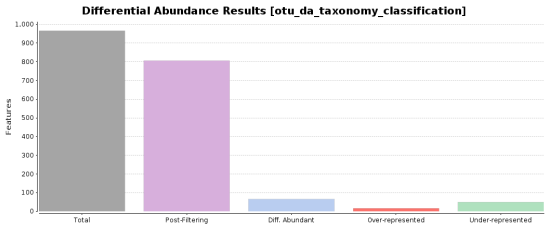


Figure 12. *OTU Differential Abundance Testing summary chart.*

Set Over/Under Tags

Establish a new FDR and Fold Change cutoff to consider OTUs as differentially abundant. FDR < 0.05 and logFC < -1 or logFC > 1 are set as default (figure 13).

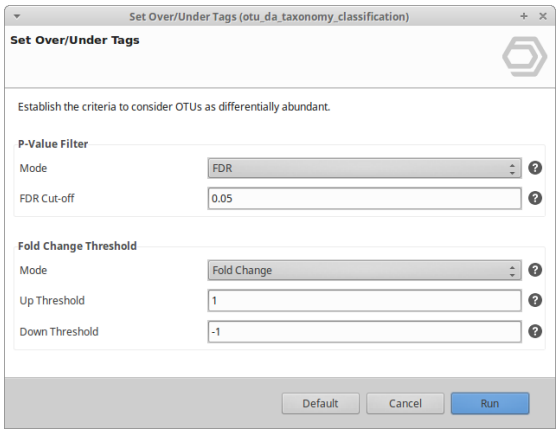


Figure 13. *Set Over/Under Tags.*

Heatmap

Shows a two-dimensional heatmap in which the abundance values are represented by ranges of colors (figure 14). The dendrograms added to the left and top side are produced by a hierarchical clustering method that takes as input the Euclidean distance computed between OTUs (left) and samples (top).

The upper bars show the experimental conditions of the study (columns) and the OTUs names are shown at the right of each row.

You can select if you want to draw the heatmap with the raw counts or with the CPM values, and if any transformation is necessary (logarithm in base 2, Z-score or both).

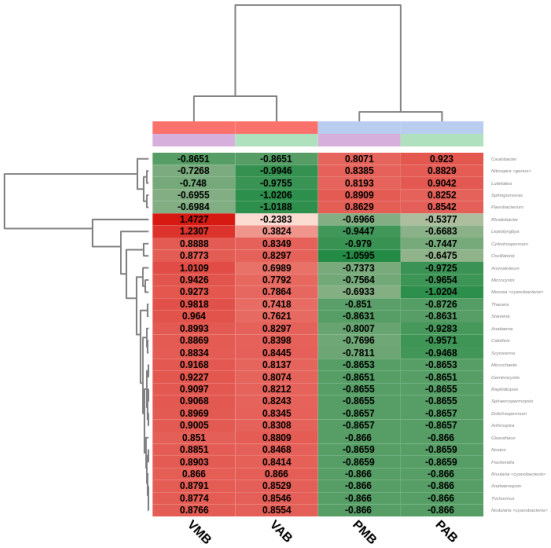


Figure 14. *Heatmap.*

REFERENCES

Robinson MD, McCarthy DJ and Smyth GK (2010). "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data." Bioinformatics, 26, pp. -1.

Differential Abundance Analysis of Functions (Pfam / EggNOG)

The **Metagenomics Functional Differential Abundance Analysis** tool is designed to detect which functional annotations are enriched between two different environmental conditions. The statistical test of this tool is based on an over-dispersed Poisson generalized linear model, specifically designed to detect the differentially abundant annotations between metagenomes.

The analysis needs one metagenomic annotation, Pfam or EggNOG, per sample and the condition to which each sample belongs as inputs. To generate the annotations, some previous steps have to be performed in OmicsBox. In starting from having raw reads for each sample, there is the need to perform a Quality Control, a metagenomic assembly, a gene-finding step, and finally the annotation. All these steps need to be running **for each sample** to finally have one annotation file per sample as shown in Figure 15. The annotation files will be used as input for the metagenomics functional differential abundance test.

This tool is based on the **ShotgunFunctionalize**library and on **HirBin**.

The Functional Differential Abundance Test only allows selecting **one type of annotation (Pfam or EggNOG).**

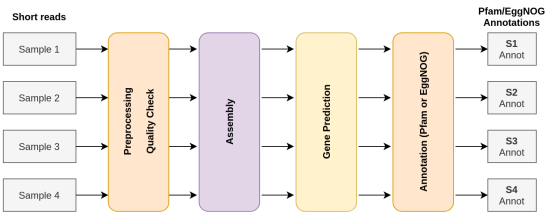


Figure 15. *Steps previous to the Functional Differential Abundance Analysis.*

GENERAL WORKFLOW

The general workflow of this tool is drawn below (figure 16).

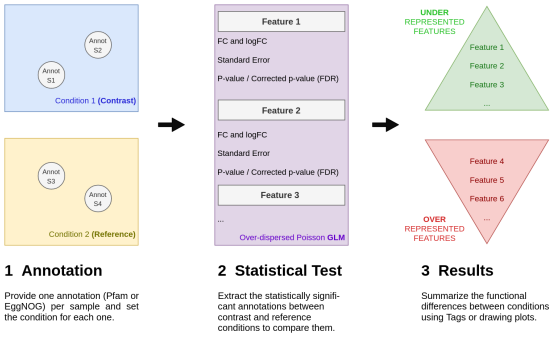


Figure 16. *Functional Differential Abundance Analysis: general workflow.*

To run the analysis, go to **Metagenomics → Comparative Analysis → Differential Abundance Analysis of Pfam / EggNOG**. The wizard allows the selection of the annotation files, the experimental conditions, and other parameters for the test (figure 17 and figure 18).

First Wizard Page - Input

On this page, the annotations files of the different samples have to be uploaded as input (figure 17).

In this section, you can also select which **items do you want to compare**. It depends on the type of annotations you provided:

- **Pfam Annotations.** You can compare at Pfam **Domain** level or at Pfam **Family** level.
- **EggNOG Annotations**. You can compare at **Cluster of Orthologous Groups** level (COGs, arCOGs, and KOGs), or at **KEGG Pathways** level.

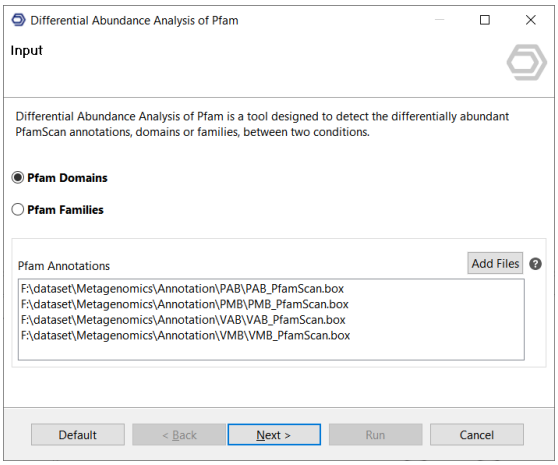


Figure 17. *Functional Differential Abundance Analysis wizard: input page.*

Second Wizard Page - Configuration

Here, the two groups for the test, reference and contrast, have to be specified (figure 18). You can select the groups by choosing which samples from the annotation files previously loaded you want to include in each condition.

Filtering

Some annotations are poorly present in the dataset and can cause false positives or alter the statistical test including non-testable features. In **Counts Filter**, you can set the minimum number of times a feature has to be annotated to be included in the statistical test. We highly recommend setting this filter to a number at least equal to the number of samples of the dataset (default = 1).

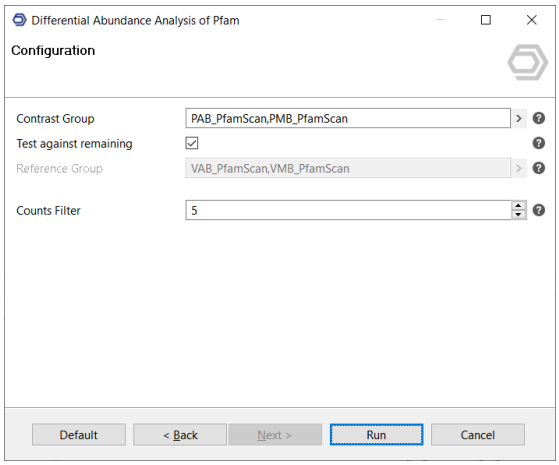


Figure 18. *Functional Differential Abundance Analysis wizard: configuration page.*

RESULTS

Once the test has finished, a **table with the results**will open (figure 19). Each row of this table corresponds to a specific annotation at the selected level (*Items to compare* option, described above). Each column represents:

- **Tags.** Indicate if the annotation is overrepresented -OVER- or underrepresented -UNDER- in the contrast condition. Thresholds are set by default in 0.05 for FDR and 2 and -2 for FC.
- **Feature and Description.** The feature ID and its description.
- **FC (Fold Change).** The ratio between the mean abundance value of a specific annotation in the contrast condition and this value in the reference condition, if the mean abundance value in the contrast group is bigger than in the reference group. If this value is bigger in the reference group, then the FC is calculated as the ratio between the mean annotation abundance value in the reference condition and the value in the contrast condition with a negative sign. By default, an annotation is defined as overrepresented if FC > 2, and it is underrepresented if FC < -2.
- **LogFC.**The log2 FC.
- **Std.Error**. The standard deviation of the coefficient point estimate in the GLM.
- **P-value.**The p-value for the null hypothesis of an equal number of annotations between conditions.
- **FDR.**A corrected p-value for multiple testing comparisons (Benjamini Y., Hochberg Y., 1995). If meeting the logFC criterion (logFC > 1 or logFC < -1 by default), an annotation must have an FDR < 0.05 to be considered as over or underrepresented in the contrast group.

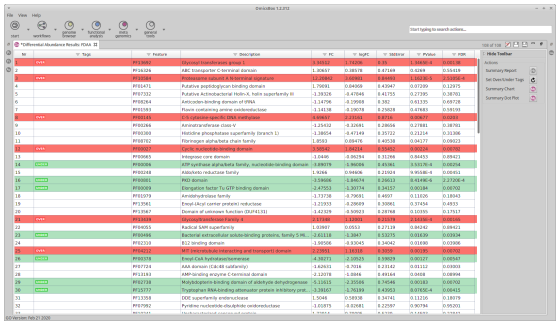
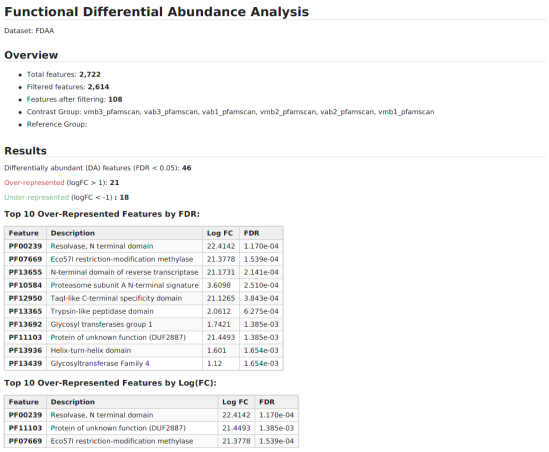


Figure 19. *Functional Differential Abundance Analysis main results.*

SIDE PANEL

Summary Report

Creates an HTML report which can be saved in PDF with the main results of the differential abundance test: parameters used for the test, number of enriched annotations, experimental design, and top 10 over and underrepresented annotations ordered by logFC and FDR (figure 20).



Functional Differential Abundance Analysis

Dataset: FDAA

Overview

- Total features: **2,722**
- Filtered features: **2,614**
- Features after filtering: **108**
- Contrast Group: vmb3_pfamscan, vab3_pfamscan, vab1_pfamscan, vmb2_pfamscan, vab2_pfamscan, vmb1_pfamscan
- Reference Group:

Results

Differentially abundant (DA) features (FDR < 0.05): **46**

Over-represented (logFC > 1): **21**

Under-represented (logFC < -1) : **18**

Top 10 Over-Represented Features by FDR:

Feature	Description	Log FC	FDR
PF00239	Resolvase, N terminal domain	22.4142	1.170e-04
PF07669	Eco57I restriction-modification methylase	21.3778	1.539e-04
PF13655	N-terminal domain of reverse transcriptase	21.1731	2.141e-04
PF10584	Proteasome subunit A N-terminal signature	3.6098	2.510e-04
PF12950	TaqI-like C-terminal specificity domain	21.1265	3.843e-04
PF13365	Trypsin-like peptidase domain	2.0612	6.275e-04
PF13692	Glycosyl transferases group 1	1.7421	1.385e-03
PF11103	Protein of unknown function (DUF2887)	21.4493	1.385e-03
PF13936	Helix-turn-helix domain	1.601	1.654e-03
PF13439	Glycosyltransferase Family 4	1.12	1.654e-03

Top 10 Over-Represented Features by Log(FC):

Feature	Description	Log FC	FDR
PF00239	Resolvase, N terminal domain	22.4142	1.170e-04
PF11103	Protein of unknown function (DUF2887)	21.4493	1.385e-03
PF07669	Eco57I restriction-modification methylase	21.3778	1.539e-04

Figure 20. *Functional Differential Abundance Analysis summary report.*

Summary Chart

Shows a bar chart with the main results: annotations pre and post-filtering steps, annotations that are considered as enriched, and the over-/underrepresented ones (figure 21).

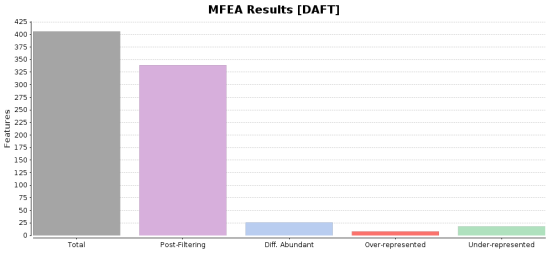


Figure 21

Set Over/Under Tags

Establish a new FDR and Fold Change cutoff to consider an annotation as significant. FDR < 0.05 and logFC < -1 or logFC > 1 are set as default (figure 22).

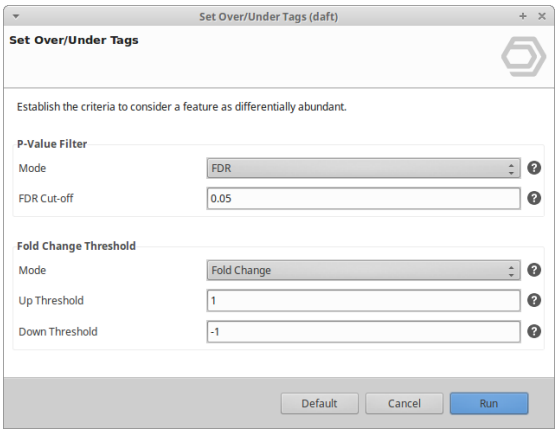


Figure 22.*Set Over/Under Tags.*

Summary Dot Plot

Shows a dot plot with the main results of the test (figure 23). You can select the date which will be included in this chart on the wizard page: represent the **over-or the under-represented features**, order them by **FC or by FDR,** and **how many annotations** the graph will contain (top 10, top 20, etc.).

Once displayed, each row of the graph contains an enriched feature. The **X-axis** represents the effect size (logFC), the **dot color** represents the significance (FDR), and the **dot size** represents the number of genes in the global dataset annotated with this feature.

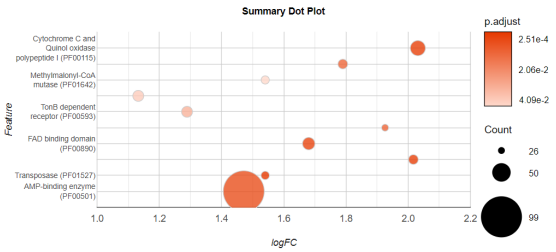


Figure 23. *Functional Differential Abundance dot plot.*

REFERENCES

- Erik Kristiansson, Philip Hugenholtz, Daniel Dalevi, ShotgunFunctionalizeR: an R-package for functional comparison of metagenomes, Bioinformatics, Volume 25, Issue 20, 15 October 2009, Pages 2737–2738, https://doi.org/10.1093/bioinformatics/btp508
- Österlund, T., Jonsson, V. & Kristiansson, E. HirBin: high-resolution identification of differentially abundant functions in metagenomes. *BMC Genomics 18*,316 (2017). doi:10.1186/s12864-017-3686-6

4.6.8 rRNA Removal with SortMeRNA

Introduction

Applying NGS technologies for metatranscriptomics profiling is common practice. It allows for full extraction of coding and non-coding RNA in a community of organisms and has become particularly important for samples that cannot be cultivated outside their native environment. The extracted RNA can be roughly divided into mRNA (messenger) and rRNA (ribosomal). It is necessary to separate both types because mRNA helps to understand the sample's gene expression patterns, while rRNA reveals information on the community's structure and biodiversity (phylogenetic analysis and taxonomic classification). rRNA can comprise up to 90% of total RNA but does not contribute to the gene expression pattern analysis. Even with pre-sequencing procedures to isolate mRNA, up to 15% rRNA may still remain in silico and can possibly be further diminished with tools like SortMeRNA. OmicsBox offers SortMeRNA to separate both types of RNA.

- **Sequencing Data:** Choose the type of input data: fasta, single-end, or paired-end. If paired-end is selected, two files per sample are required and the file pattern has to be provided.
- **Reads:**Select files that contain the desired input data.
- **Paired-end configuration:** When working with paired-end libraries, a so-called pattern has to be established to help the software distinguish between upstream and downstream read files. Per default, we assume the following pattern:
- upstream: SampleA_1.fastq
- downstream: SampleA_2.fastq

For SRR037717_1.fastq and SRR037717_2.fastq as up and downstream files, please select "_1" and "_2" respectively for the patterns.

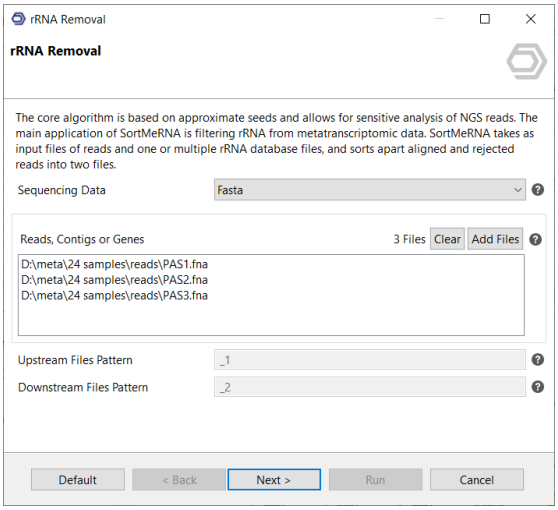


Figure 1

Several rRNA databases are available, and user compiled databases can also be provided by selecting **Additional Database** in **Target Databases**. This allows uploading own Fasta files with the **Additional Database** file selection widget.

Paired Mode configures how SortMeRNA handles read pairs with ambiguous alignments:

- Paired In: With one aligned read, both are considered aligned.
- Pared-out: If one read can not be aligned, both are considered not-aligned.

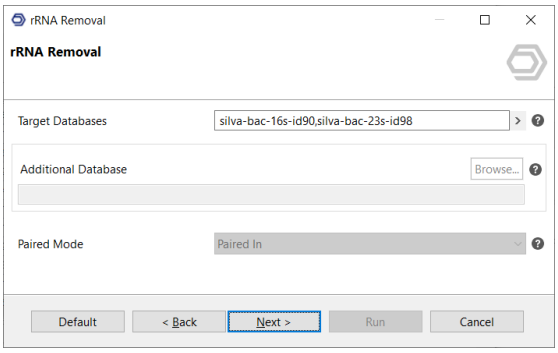


Figure 2

Save rRNA and mRNA separately and discard results if not desired.

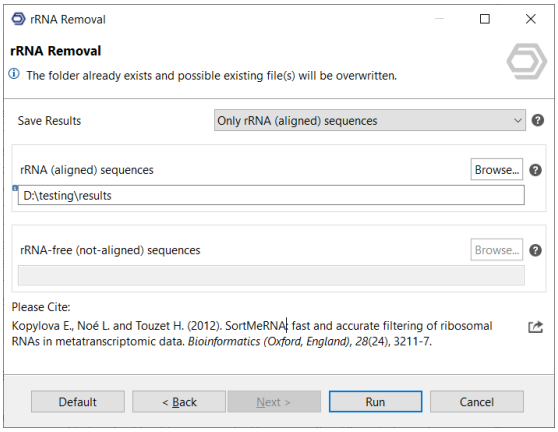


Figure 3

References

Evguenia Kopylova, Laurent Noé, Hélène Touzet, SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data, *Bioinformatics*, Volume 28, Issue 24, December 2012, Pages 3211–3217, https://doi.org/10.1093/bioinformatics/bts611

5. How to cite OmicsBox

If you have utilized OmicsBox for your data analysis, please remember to cite us, along with the specific bioinformatics algorithm employed. When citing the OmicsBox bioinformatics platform, please use the following format:

> *OmicsBox – Bioinformatics Made Easy, BioBam Bioinformatics, March 3, 2019, https://www.biobam.com*

In OmicsBox, all bioinformatics tools come with their respective citations, which can be found on the wizard page and in the results reports. For instance, if you've utilized the Blast2GO functional annotation methodology within OmicsBox, it's important to cite both OmicsBox and Blast2GO:

> *Götz S., Garcia-Gomez JM., Terol J., Williams TD., Nagaraj SH., Nueda MJ., Robles M., Talon M., Dopazo J. and Conesa A. (2008). High-throughput functional annotation and data mining with the Blast2GO suite. Nucleic acids research, 36(10), 3420-35.*