TriFusion

User Guide

Streamlining phylogenomic data gathering, processing and visualization

by Diogo N. Silva

CE3C
Centre for Ecology, Evolution and Environmental Changes

CoBiG²
http://cobig2.fc.ul.pt
Computational Biology & Population Genomics Group

Contents

1. Overview

1.1 About TriFusion

TriFusion is a GUI and command line application designed to streamline the workflow of phylogenomic projects. With the dramatic increase in size of data sets for phylogenetics and population genetics, programming has become a crucial tool to gather, process and analyze the data. However, this may still represent a hurdle that precludes the execution of such projects by a broader range of researchers. TriFusion aims to mitigate this issue by providing a user-friendly visual interface that empowers users without any programming knowledge with a set of tools and operations that can handle large sets of data. In general, TriFusion's features are separated in three main modules:

- **Orthology:** Detect orthologs across multiple proteomes and explore the resulting ortholog groups taking advantage of filters and several graphical analysis.
- **Process:** In its basic form, alignment sequences in multiple supported input formats can be converted, concatenated, reverse concatenated, filtered, collapse (and much more) into several output formats
- **Statistics:** Take advantage of dozens of graphical and statistical analyses to painlessly explore big (and small) sets of alignment data.

A more extensive overview of TriFusion's features is shown in the next section. TriFusion is an open source, cross-platform application written in Python 2.7 and using the Kivy framework

(https://kivy.org/).

1.2 Features

TriFusion provides several features across its three main modules. Here is an overview of what it can do for you.

1.2.1 Orthology

The **Orthology** module of TriFusion deals with the gathering of ortholog data from proteome data (fasta files with protein sequences), which marks the initial stages of many phylogneomic projects.

Detection of putative orthologs

Given a set of multiple proteome (protein sequence) files, TriFusion uses the *OrthoMCL* framework to identify putative ortholog groups [3]. Multiple sequential runs can be executed using different inflation values, which is the most important parameter in the *OrthoMCL* workflow.

Application of filters to ortholog groups

The *OrthoMCL* output includes ortholog groups with multiple copies and with no minimum taxa representation requirements. TriFusion offers filters on the maximum number of gene copies and minimum taxa representation required for an ortholog group to be selected. The effect that different values for these filters may have on the final number of orthologs can be later assessed through several plotting options.

Full graphical report of orthology search results

TriFusion employs a number of diverse plotting options that allow users to explore the results of the ortholog search. These plots can be generated within the application with the option of changing the filters on the ortholog groups and visualize their impact on the number of orthologs on the fly. Alternatively, a full report can be issued and plots automatically generated and visualized in HTML format.

Ability to import multiple independent search runs

TriFusion is able to import the main output of ortholog searches performed within TriFusion or using *OrthoMCL* directly. This means that multiple orthology searches can be performed independently in different systems and their results imported into TriFusion for comparative and joint analyses.

Automatic conversion of ortholog groups into protein and nucleotide sequence files

The ortholog groups identified in the search operation can be converted into protein and/or nucleotide sequence files in Fasta format, given protein and CDS database files, respectively. These conversions can be performed at any time during the exploration of the search results.

1.2.2 Process

The **Process** module of TriFusion deals with the conversion and manipulation of large sets of alignment files.

Handling of large alignment data matrices, made easy

TriFusion's **Process** module was designed to handle large sets of data ($\geq$7k alignment files) that are commonplace in phylogneomics/population genomics projects. The sheer size of such data can be quite intimidating when thinking of how it can be processed and it may be difficult to explore the data. This module aims to provide a diverse set of intuitive options that can be easily executed in order to modify and prepare these data matrices quickly and efficiently for downstream analyses. Even though all of its features are able to handle large data sets, most options can be equally useful for smaller data sets.

Performs basic conversion/concatenation operations

The main operations of the **Process** module are the basic conversion and/or concatenation of alignment files. Several commonly used input formats are supported (Fasta, Nexus, Phylip, PyRAD .loci, Stockholm) as well as several output formats (Fasta, Nexus, Phylip, MCMCTree, Stockholm, GPhoCS, IMa2), including variations of these formats for specific downstream software.

A diverse portfolio of secondary operations

Where TriFusion really shines is in the breadth of available secondary operations that can be performed along with the basic operations, perhaps because they are the most difficult (and necessary) to employ for larger data sets without programming assistance. These operations are optional but most can be easily combined in a single execution. For example, it is possible to execute the concatenation of 100 files and at the same time filter the resulting alignment so that columns with more than 75% of missing data are excluded and code gaps in binary format at the end of the sequence matrix. Alternatively, all secondary operations have the option to save their output in a separate file instead of modifying the main output file. It's up to you.

Collapsing alignments

For downstream analyses where only unique sequences are required, TriFusion is able to collapse an alignment as a secondary operation. Here, taxa with identical sequences are effectively collapsed and represented in the output alignment as a single sequence. An auxiliary file is also created mapping the new haplotype names to the original taxa names.

Creating consensus alignments

For downstream analyses that require only a single representative sequence per alignment, the consensus secondary operation is available. In this case, all sequences of an alignment will be merged into a single sequence and variation within the alignment can be handled in one of several ways available. If multiple input alignments are provided for the creation of the consensus, there is also the option to save all consensus sequences into a single file, instead of in separate files.

Filters, Filters and more Filters

Alignment filtering is one of the most important steps when processing phylogenomics data sets and this module provides several filtering options for taxa, codon positions, missing data and sequence variation:

- Taxa filter: Input alignments can be filtered depending on whether they contain or exclude a user-provided taxa group
- Codon filter: For protein-coding nucleotide sequences, alignments can be filtered in order to exclude certain codon positions
- Missing data filter: This filter can be applied both within alignments or among a set of alignments. Within an alignment, it is possible to remove columns that contain a proportion of gaps (usually "-") and/or missing data (usually "N") above user-defined thresholds. Among alignments, it is possible to remove ones that contain less taxa than minimum number allowed by the user.
- Variation filter: Input alignments can be filtered when they contain a number of variable and/or informative sites outside the range specified by the user.

Gap coding

When the information contained in the indel pattern of an alignment may be of use for downstream analyses, the Gcoder secondary operation can be used. Here, the method described in Simmons and Ochotenera [4] is employed to transform indel patterns into a binary matrix that is appended to the end of the sequence data. This operation can only be performed when the

output format is Nexus.

Creation of data subsets

The operations of the Process module need not be only applied to the complete data set that is currently loaded into the application. With a single click, individual alignment files or taxa can be toggled out of the active data set and ignored from subsequent operations. This can be very useful when, for example, a complete data set includes mitochondrial and nuclear genes and the user wishes to concatenate all genes, then only the mitochondrial genes, and finally only the nuclear genes. Likewise, it is possible to exclude certain groups of taxa from some operations. For larger or more complex scenarios, TriFusion provides creation dialogs that allows the definition of subsets of alignments and/or taxa, which can be easily switched.

Set gene partitions, codon partitions and substitution models

Gene partitions are fully customizable within the application or by importing partition schemes. For nucleotide sequences, all possible combinations of codon partitions can be specified along with the substitution model for each partition. This information is then saved on the output formats that support this information (Nexus and Phylip partition file).

Revert a concatenated alignmnet

A previously concatenated alignment can be reverted to its individual and separate files using the revert concatenation option, by providing information on its partitions. These partitions may be already defined in the alignment file (Nexus format), imported from a separate file with partitions definitions, or defined within the application.

Concatenation of alignment weight files

Many modern phylogenetic software support the attribution of weights to each alignment column as a measure of uncertainty from the alignment procedure. This alignment column weights can be obtained using software such as *Zorro*. However, there is usually lack of support when concatenating multiple alignment files with their corresponding weight files. TriFusion offers the option to joint concatenate alignment files and their corresponding weight files in a user-friendly way.

1.2.3 Statistics

Because figures may be worth a thousand words, the **Statistics** module offers a diverse set of plotting and statistical options that allows users to quickly and easily visualize many aspects of

their data set. In other words, it lets you explore the data, effortlessly. These plotting options are sorted into four main categories: *General information*; *Polymorphism and variation*; *Missing data*; *Outlier detection*.

Visualize your data set focusing on single genes, taxa or genes

For the majority of the plotting options, there are more than one way to look at the data. For example, when you wish to investigate the sequence similarity of your data set, this information can be displayed focusing on: (i) a single gene, using a sliding window approach; (ii) taxa, using a triangular heat matrix; (iii) gene average, using a distribution of sequence similarity values across all genes. Moreover, the active data sets can also be modified, as in the Process module, and all plots will be updated according to the new active data sets.

Get summary statistics and general alignment information instantly

After loading an alignment data set into TriFusion, the first operation performed by the **Statistics** module is to compute several summary statistics and general gene information unobtrusively in the background. When the calculations are done, they are revealed in the main **Statistics** screen.

General information plots

The *General information* category offers plotting options with information on the distribution of sequence size, proportion of nucleotides/residues, and distribution of taxa frequency.

Polymorphism and variation plots

The *Polymorphism and variation* category focus on plotting options that investigate the amount and distribution of sequence variation found in the alignment data set. The user may investigate the pairwise sequence similarity, number and distribution of segregating sites and compute the allele frequency spectrum (only for nucleotide alignments). An additional operation calculates the correlation coefficient between the length and sequence variation of the alignments.

Missing data plots

Missing data plots allows the user to investigate the amount and distribution of missing data in the data set. The plotting options include a gene occupancy plot, which allows a quick visualization of the amount of missing genes/taxa in the data set, distributions of missing taxa and data, and the cumulative distribution of missing genes to assist the user on the best minimum taxa representation threshold that maximizes the amount of data and minimizes the missing data.

Outlier detection plots

Outlier detection plots will assist the user in the identification of taxa or genes with outlier characteristics, which may be indicatives of problems during the creation of the alignment data set. Outlier taxa or genes can be identified based on the sequence length, number of segregating sites and amount of missing data.

Publication ready figures and tables

For nearly all options it will be possible to export the current plot as a figure, in a number of output formats, as well as in table format (in .csv). These plots have been extensively customized in order to be of high quality and publication ready and include the option to convert into black and white versions.

Plot fast switching

Some plotting options may take a while to complete, depending on the size of the active data set. To reduce waiting times, a number of techniques were employed to allow fast switching between plots that were already generated. In addition, costly computations are usually stored in SQLite data bases to reduce computing times when changing the active data set.

2. Download and Installation

2.1 Using binaries and installers

The easiest way to install TriFusion is through binaries and installers provided for Windows and major Linux distributions:

2.1.1 Windows

- TriFusion 64bit installer
- TriFusion 32bit installer

2.1.2 Linux

- Debian-based: trifusion-bin-*.deb
- RPM-based: trifusion-bin-*.rpm
- Arch Linux: trifusion-bin
- Other: trifusion-bin-*.tar.xz

2.2 Installation from source

TriFusion requires several dependencies to properly run. HOWEVER, for the most popular distributions package builds are as described below (Windows: 2.2.2,; Ubuntu: 2.2.3 ; Debian-based: 2.2.4 ; RPM-based: 2.2.5 ; ArchLinux: 2.2.6), which handle all the necessary dependencies for

you. If for some reason you have to install everything manually, here are all dependencies listed with instructions on how to install them.

2.2.1 Manual installation of dependencies

> **Note for Linux users:** Most, if not all of these dependencies, may already be present in the official repositories of your distribution (e.g. in Ubuntu: sudo apt-get install python-numpy).
> ■

- **python2.7** is required to run TriFusion. It should already be present in most Unix operating systems but not on Windows. For Windows users, I recommend installing a python distribution package (such as Anaconda) that already comes with several required python libraries.
- **kivy** is a python module responsible for the user interface of TriFusion. Instruction on how to install it are provided here.
- **matplotlib** (included in Anaconda)
- **numpy** (included in Anaconda)
- **psutil** (included in Anaconda)
- **scipy** (included in Anaconda)
- **configparser** (included in Anaconda)
- **seaborn** (included in Anaconda)

Assuming that **python2** and **kivy** are already installed, the remaining dependencies can be installed by running the following command inside TriFusion's directory (with super user privileges for system-wide installation):

```
1 python2 setup.py install
```

2.2.2 Windows

Assuming you have installed Anaconda (which automatically installs most of the dependencies), you will need to install kivy by executing the following commands on a command line prompt:

```
python -m pip install --upgrade pip wheel setuptools
python -m pip install docutils pygments pypiwin32 kivy.deps.sdl2
    ↪ kivy.deps.glew
python -m pip install kivy.deps.gstreamer --extra-index-url
    ↪ https://kivy.org/downloads/packages/simple/
python -m pip install kivy
```

Then, download the latest TriFusion code. Open a command line prompt (or power shell, Anaconda shell) with Administrator privileges and navigate to TriFusion's directory. Once there, run the following command to install TriFusion:

```
python2 setup.py install
```

2.2.3 Ubuntu-based

- Add one of the following PPAs:
  - Stable release:

```
sudo add-apt-repository ppa:o-diogosilva/trifusion
```

  - Daily release:

```
sudo add-apt-repository ppa:o-diogosilva/trifusion-daily
```

- Update your package list:

```
sudo apt-get update
```

- Install TriFusion

```
sudo apt-get install trifusion
```

2.2.4 Debian-based

- Add one of the following PPAs manually to the `sources.list` file:

  - Stable release:

    `http://ppa.launchpad.net/o-diogosilva/trifusion/ubuntu trusty main`

  - Daily release:

    `http://ppa.launchpad.net/o-diogosilva/trifusion-daily/ubuntu trusty main`

- Add the GPG key to your apt keyring:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
↪ D4F1E8E6
```

- Refresh your package list and install TriFusion

```
sudo apt-get update && sudo apt-get install trifusion
```

2.2.5 RPM-based (Fedora, OpenSuse, RedHat)

Before installing TriFusion, you will need to install Kivy:

```
dnf install python-pip python-devel redhat-rpm-config
↪ freeglut-devel SDL*
pip install cython==0.23
pip install kivy
```

Then, download the latest RPM release of TriFusion and install it as regular RPM package.

Note that the RPM package is not in any repository yet, which means that it will not receive the updates of the latest version of TriFusion in GitHub. However, it's possible to stay in the bleeding edge of the development, by following these steps:

- After installing the RPM package remove ONLY the TriFusion package, but leave the dependencies installed. For example, in Fedora24:

```
dnf remove TriFusion
```

- Clone the git repository of TriFusion to your computer

```
git clone https://github.com/ODiogoSilva/TriFusion.git
```

- Open a terminal in TriFusion's directory and execute:

```
sudo python setup.py install
```

Since all dependencies are already installed, this will install only the most recent version of TriFusion. To keep TriFusion up to date afterwards, you'll just need to pull the most recent code and re-install it.

```
git pull
sudo python setup.py install --force
```

2.2.6 ArchLinux

There are three AUR packages for TriFusion:

- trifusion: The latest release of TriFusion, based on source code.
- trifusion-bin: The latest release of TriFusion. in binary format. Does not require dependencies to be installed, as all the necessary libs are bundled with the distributed binary
- trifusion-git: The bleeding edge version directly from git. Requires dependencies to be installed, as it is also source code based.

Just use any AUR helper to handle the packages for you, or download the PKGBUILD you require and use makepkg.

2.3 Installing auxiliary software

USEARCH

The USEARCH software is necessary to perform the **Orthology Search** operation (section 5.2). Due to the licensing of the software, it cannot be bundled with TriFusion and you will have to download it yourself.

If TriFusion is unable to find the USEARCH executable, you will see a warning the **Orthology** module, *Search* operation, under the *USEARCH* tab of the additional options (Fig 2.1).

To fix this, you can click the Fix it button, where a dialog will show up with the link to download USEARCH. Once the single binary USEARCH file is downloaded, you can click the Search USEARCH executable button and provide the binary to TriFusion. If the binary checks out, TriFusion will store it locally and it will be available for all future sessions. The warning will then change to a positive check message (Fig 2.2).

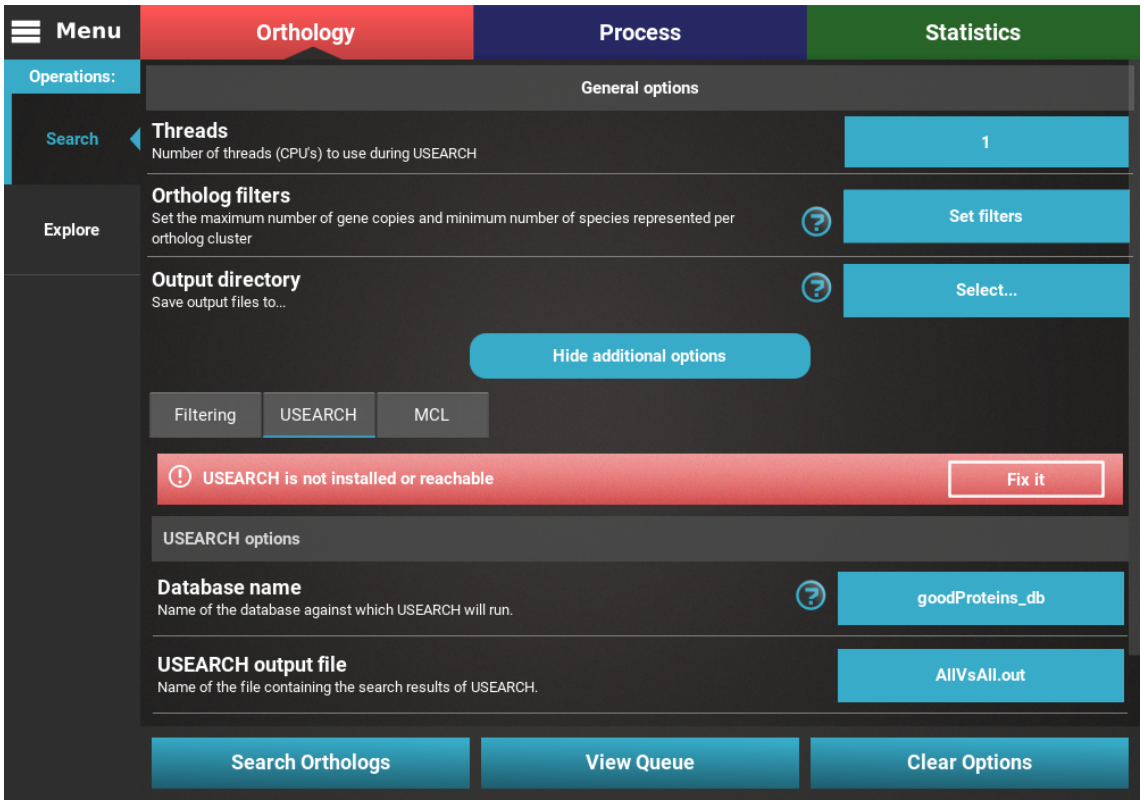


Figure 2.1: Warning shown by TriFusion when the USEARCH executable cannot be found

Figure 2.2: Check message when USEARCH is correctly installed and reachable by TriFusion

3. Getting help

3.1 Reporting bugs, request features or make pull requests

TriFusion is being continually developed, so we always appreciate bug reports and feature requests. These can be reported using the **Issues** page of TriFusion's GitHub page (https://github.com/ODiogoSilva/TriFusion/issues). Please ensure that a similar bug or feature have not yet been reported.

If you would like to contribute pull requests are also welcome!

TriFusion - GUI and CLi versions

4. TriFusion General Interface and Utilities

4.1 Import data

TriFusion deals with two main types of data: **proteomes** and **alignments**. Proteomes are Fasta formatted files containing protein sequences for a particular taxa (more details in section 5.2.1). Alignments are sequence sets of the same length that may be in one of several supported formats (more details in section 6.2). In either case, files can be loaded into TriFusion in three main ways.

File browser

Open the file browser by clicking the `Open File(s)` button in *Menu* > *Open/View Data*, or using the keybinding *Ctrl+O*. In the top of the screen the file type can be selected (alignment or proteome). Then, single files can be loaded simply by double-clicking. Multiple files can be loaded by selecting them and then clicking `Load selection`, to load data and remain in the file browser screen, or `Load & go back`, to load data and go to the previous screen.

> **Tips and tricks**
>
> - *Path edition and auto-completion*: The file browser path can be manually edited by clicking the  button. Path auto-completion is also supported using the *Tab* key.
> - *Interactive path*: If the current path is not in edit mode, any directory path can be clicked to go directly to that directory.
> - *File searching*: A file search box can be used (*Ctrl+f* to focus) to search for specific files

or directories. Searches can be case sensitive or insensitive.

- *Filter by extension*: A drop-down menu is provided to filter files according to their extension.

■

Drag and Drop

TriFusion supports drag and dropping of files from the system's file browser into the application window. This works in all screens, with the exception of TriFusion's main file browser screen. Once files are dropped, a dialog pops up displaying the number of files to be loaded and asking you whether the files represent proteomes or alignments.

Note to Linux users

The drag and drop functionality may not work correctly when using wayland.

■

Open files with TriFusion via terminal

For terminal lovers, files can be automatically loaded when launching TriFusion from the terminal. All arguments passed after the *TriFusion* keyword, and separated by spaces, are assumed to be input files:

```
1 TriFusion file1.fas file2.fas file3.fas
```

When the application opens, a dialog pops up displaying the number of files to be loaded and asking you whether the files represent proteomes or alignments.

4.2 Project management

Some data sets can be recurrently used and it may be a nuisance to be constantly loading the data manually. In these cases, users can save the currently loaded files as a new project by navigating to *Menu* > *Project Management* and clicking Save current project. A name for the project will be required to complete the process and then the project will be saved for future sessions. Depending on whether the loaded files are proteomes or alignments, a label is paired with the project button to inform the user on the project type.

To load saved projects, you may navigate to *Menu* > *Project Management* and click on the desired project. Alternatively, every time TriFusion launches, a *Quick Open Project* panel displays all saved projects for quick loading. When a project button is clicked, a dialog pops up

showing the number of files that will be loaded. **Note that whenever a project is opened, any previously loaded files are removed from the current session!** Projects can be removed at any time, by clicking the removal button.

4.3 How to visualize loaded data

All data loaded into TriFusion, including **files**, **taxa** and **partitions**, can be visualized in *Menu* > *Open/View Data* at their corresponding tabs. This panel offers several convenient functionalities to quickly explore and manipulate the data.

4.3.1 Files and Taxa tabs

The interface for the files and taxa tabs is similar. The functionalities include:

- File/taxa buttons can be toggled to add or remove them from the active data set (See 4.4). If the name is too big to show in the button, hovering the mouse over will display a label with the complete name after a time. **For taxon buttons only:** Double clicking a taxon button allows the name to be edited.
- Information button ( ). Displays a number of summary metrics for files and taxa (See 4.3 and 4.4).
- Removal button ( ). Permanently removes the file/taxon from the current session.
- Additional options button ( ), provides functionalities for manipulating the data sets from text files, such as removing, selecting and exporting files/taxa.
- Complete removal button ( ), removes all files and taxa and resets the current session in terms of input data.

Informative pop-up for Files

- **Input format:** The format of the input file (Fasta, Nexus, Phylip, etc.)
- **Sequence type:** Whether the file has protein or nucleotide sequences
- **Sequence size:** The size of the alignment in residues/nucleotides
- **Number of taxa:** It really is the number of taxa.

Informative pop-up for Taxa

- **Sequence length:** The combined sequence length for the current taxon. This includes gaps and missing data within files where the taxon is present but does not account for

the sequence length of files where it is absent.

- **Number of indels:** Number of gap symbols.
- **Number of missing data:** Number of missing data symbols ("**N**" for nucleotides; "**X**" for proteins).
- **Effective sequence length:** The sequence length minus the number of indels and missing data. The percentage value refers to the proportion of effective sequence length from the total sequence length.
- **File coverage:** The number of files where the taxon is present.

All these metrics are available for the total data set (the entire data set loaded into TriFusion) and for the active data set (See 4.4).

4.3.2 Partitions tab

When loading data into TriFusion, if no partition scheme is provided in the input (generally as a *charset* block in Nexus files), each individual alignment file is considered a partition in the data set. However, the resulting partition scheme can be easily changed within TriFusion (See 4.3.2) or imported from a file (See 4.3.2). Besides the edition of the partition scheme, the functionalities in this tab include:

- An alignment counter button (e.g., ), displaying the number of alignment files contained in the corresponding partition. Clicking it lists the individual alignment files.
- An edition button (), showing the name of the partition (which can be edited), its length and substitution model options. Since TriFusion automatically detects if the input alignments are protein or nucleotide sequences, the substitution models available for selection will be specific of the sequence type. For nucleotide sequences, there is also the option of setting codon partitions with specific substitution models.

Editing the partition scheme in TriFusion

Partitions can be either **merged** or **split** within the app:

- **Merging partitions:** Two or more partitions can be merged by selecting the desired partition buttons and clicking the merge button (). A unique name must be provided for the new partition. After the partition has been created, it will be added to the end of the partition list. If this list is too long, try searching for the partition name to check if everything is ok.
- **Splitting partitions:** Individual partitions can be split into two or more partitions by

clicking the split button (). If the partition already contains multiple alignment files, it can be easily split into one partition per file simply by clicking the Split by files button. Partitions for each alignment file will be created and named based on the file name. Alternatively, the partition can be split into two by setting the splitting point in the slider, and then providing the names for the resulting partitions.

Import partition scheme from file

TriFusion can import partition schemes formatted in one of two popular formats:

- **Nexus charset block:** This is the standard format for partition definition in Nexus files. Each line starts with a *charset* keyword, followed by the name of the partition, an equal ("=") sign, and the range of the partition. In the example below, five simple partitions are defined:

```
charset Teste1.fas = 1-85;
charset Teste2.fas = 86-170;
charset Teste3.fas = 171-255;
charset Teste4.fas = 256-340;
charset Teste5.fas = 341-425;
```

Codon partitions can also be defined using the "/3" notation. For example, if you wish to divide the partition Test1.fas into three codon partition, this could be defined as:

```
charset Test1_1.fas = 1-85/3;
charset Test1_2.fas = 2-85/3;
charset Test1_3.fas = 3-85/3;
```

Additional symbols are also supported. For instance, the "." symbol can be used to specify full length. If a data set has 1000bp, any of these partitions is correctly defined:

```
charset Test1 = 1-1000;
# Or
charset Test1 = 1-.;
```

Substitution models can also be specified in this file in a similar way as in the MrBayes software. This can be achieved by using the *applyto* keyword, the number of the partition to which the model will be applied, and then the model definition as it would be written

for MrBayes:

```
applyto=(1) lset nst=2 statefreqpr=dirichlet(1,1,1,1)
```

This would apply the *HKY* substitution model to the first defined partition.

- **RAxML partition file:** This partition file format offers less options than the Nexus charset block and is usually more suitable for simple (but possible large) partition schemes. Here, partitions are simply defined in each line by providing the substitution model, the name of the partition and then its range:

```
GTR, BaseConc1.fas = 1-85
GTR, BaseConc2.fas = 86-170
GTR, BaseConc3.fas = 171-255
GTR, BaseConc4.fas = 256-340
GTR, BaseConc5.fas = 341-425
```

4.4 Total and active data sets

Most of TriFusion's features in the **Process** and **Statistics** modules can be applied to either the total loaded data set (all files and all taxa), or to custom made active sets of alignments and/or taxa. By default, all files and taxa loaded into TriFusion are active and, therefore, the total and active data sets start out identical. The active data set can be modified in several ways, as described below. The number of active files/taxa can be seen at any point in the label below the list of files/taxa.

4.4.1 Toggle file/taxa buttons

All file and taxa buttons in their corresponding tab in *Menu* > *Open/View Data* can be toggled to add or remove them from the active data set. This method is suitable for quick one-time operations on the active data set.

4.4.2 Selecting file/taxa from text file

The active file/taxa set can be selected by providing a list of file/taxa in a text file. In *Menu* > *Open/View Data*, click the  button in the bottom of the panel to open the additional options box, and then click Select file/taxa names from .txt. The text file should be simply a file containing one file/taxa entry per line. As an example:

```
SpeciesA
SpeciesB
SpeciesC
```

In the case of file names, the extension of the file must be included as well. After selecting the text file, the active file/taxa set will be changed according to the entries in the file. Any entry with no match in the current data set is ignored.

As a convenience, both total and active data sets can be exported to text files for future sessions using the export functionality also present in the additional options box.

4.4.3 Create data set groups

Multiple file and taxa groups can be defined in *Menu* > *Dataset Groups*, by clicking the Set new file group or Set new taxa group buttons. In either case, groups can be defined manually inside the application (recommended for smaller data sets), or provided via a text file (recommended for large data sets) similarly to the previous section. Once the datasets are created, they will appear listed in their corresponding panel where they can also be removed.

Manual definition of data set groups

The data set creation dialog allows the definition of files/taxa set directly from TriFusion (Fig 4.1). Files/taxa can be transferred between panels using the center arrow buttons. Once the desired files/taxa are selected, clicking the Apply button will create the data set group but retain the creation dialog open for further group definitions. Created groups will appear in the left panel and can be clicked to visualize their members and use it as a starting point to define other groups. Alternatively they can also be deleted using the button.

Create data set groups from file

Datasets can also be created from file in the same way files/taxa are selected from files (See 4.4.2). The only difference is that a name for the data set group is required. **This method is strongly recommended for larger data sets (>1k entries) as a large number of files/taxa may decrease the performance in the creation dialog.**

4.5 In-app help

Throughout the application, in-app help for several options is available when the button is associated. Clicking this button will display a short pop-up with information relevant to that

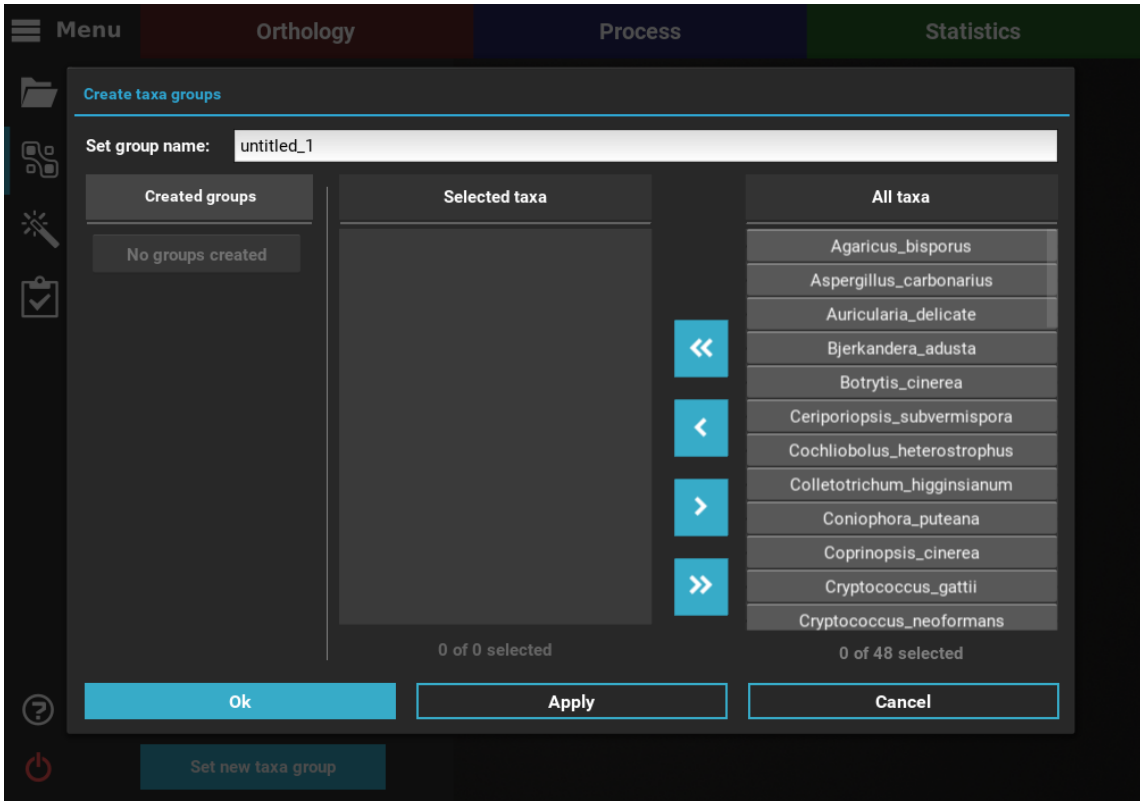


Figure 4.1: Manual data set creation dialog

option.

5. Orthology module

5.1 Overview

The **Orthology** module was designed to streamline the acquisition and exploration of ortholog sequences across a number of species, taking advantage of the *OrthoMCL* framework [2]. In fact, the complex process of searching for ortholog groups across multiple proteome files is mostly automatic in TriFusion, requiring user input for only a few options. In this way, the user may focus most of its attention to the most relevant task of investigating the data characteristics of the identified ortholog groups (missing data, gene copies, etc.). TriFusion offers a number of filters and graphical tools to help in this regard (see Section 5.3). At any point during this investigation, users will also be able to export ortholog groups into protein and/or nucleotide sequences. All these tasks are sorted in two operations: ***Search*** and ***Explore***.

> **Note for command line users:**
>
> Throughout this section, the options included in the **Orthology** module will focus mainly on the GUI version. However, when available, each option will also include the command line version. A reference guide to the command line options is described in section 8.1. A complete pipeline usage is also show in section 8.2.1. ■

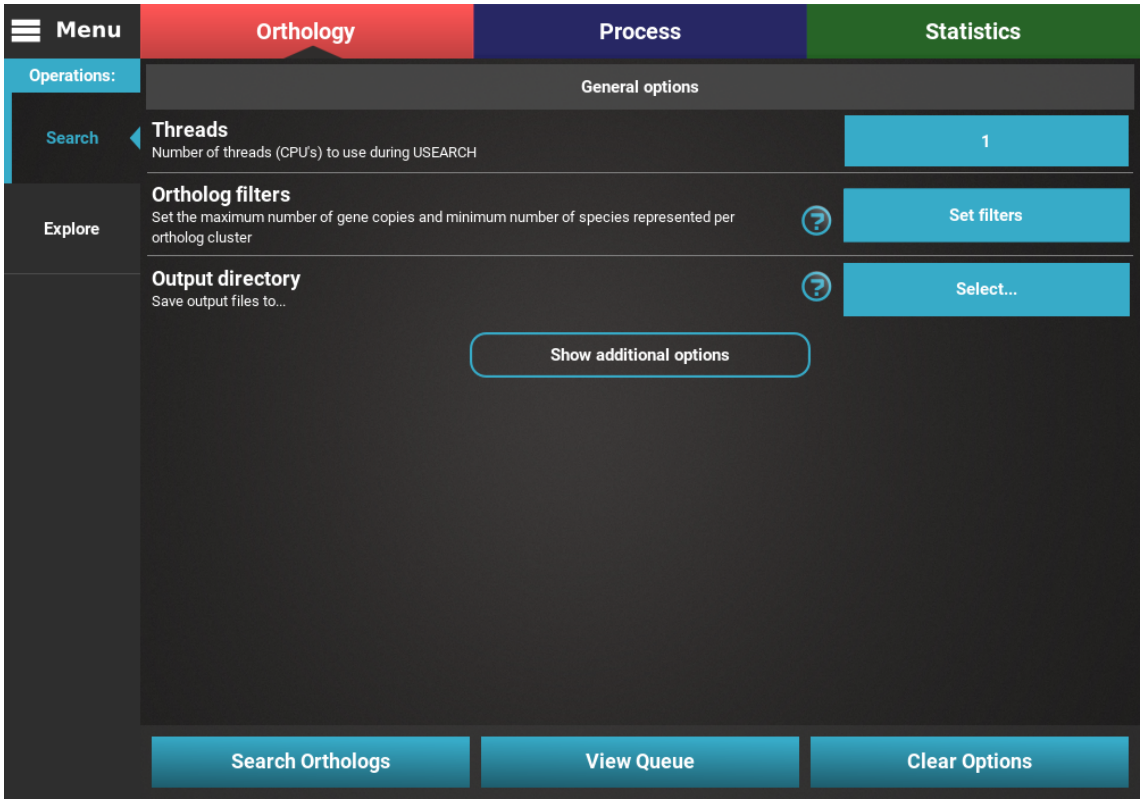


Figure 5.1: Main screen of **Orthology Search** operation

5.1.1 Search - Overview

Using the ***Search*** operation (Fig 5.1) it is possible to search for potential ortholog groups from a set of multiple proteome files in Fasta format using an adapted code of *OrthoMCL* that runs faster and relies on SQLite instead of MySQL for relational database operations. There are several paramenters that can be changed by the user, but the most relevant are only three:

- Maximum number of gene copes allowed (section 5.2.2)
- Minimum number of taxa allowed (section 5.2.2)
- Inflation value for the *MCL* analysis (section 5.2.5)

The main output of this operation will be the creation of a directory containing the unaligned sequence files for each ortholog group, and a file containing information on the sequences of all ortholog clusters. The latter file is the main output of OrthoMCL and is called a **group** file. Multiple **group** files can be created in a single ***Search*** run by specifying a range of inflation values and can be later provided as input to the ***Explore*** operation for comparison and investigation of the results.

5.1.2 Explore - Overview

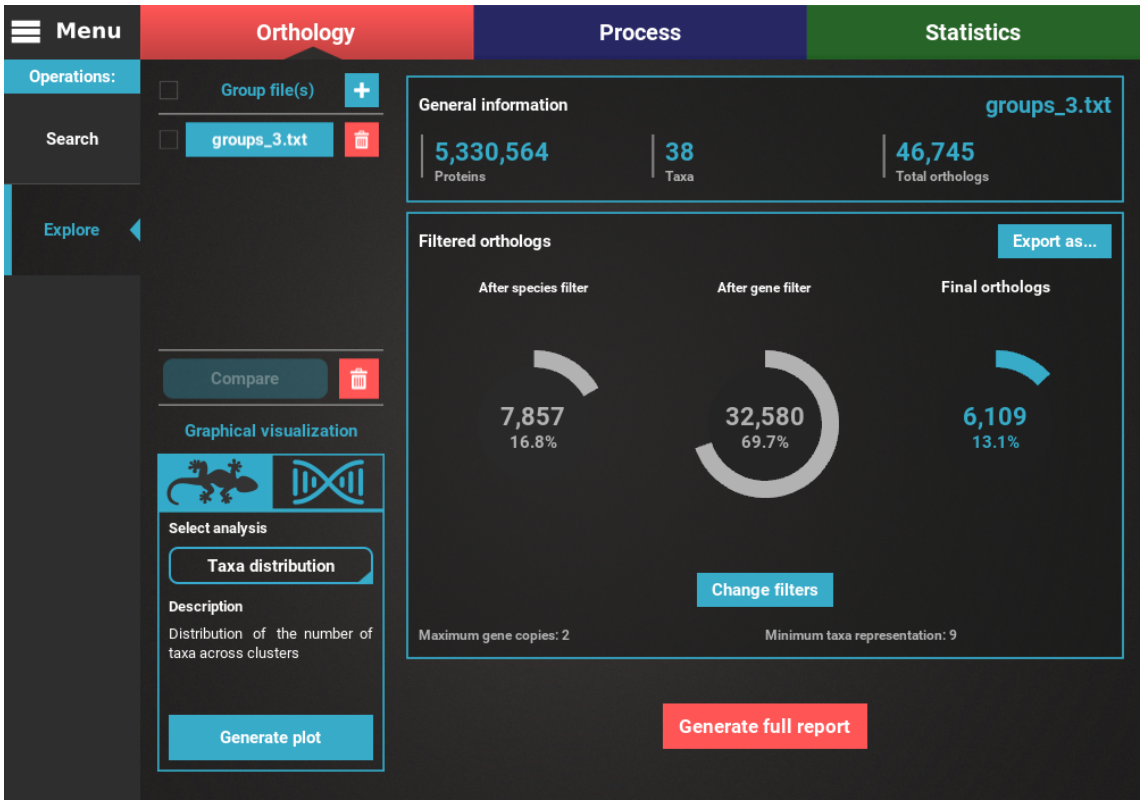


Figure 5.2: Main screen of **Orthology Explore** operation

The ***Explore*** operation (Fig 5.2) allows users to compare different **group** files, whether they were generated by OrthoMCL directly, the ***Search*** operation or TriFusion's command line pipeline, ***orthomcl_pipeline***. Here, users can modify filters and visualize diverse aspects of the obtained ortholog groups. During this investigation, it is possible to evaluate the effect of the filters on several parameters (missing data, gene copies, etc.) and at any point the users will have the option to easily export the orthologs with the current filters as protein or nucleotide sequences.

5.2 Search operation

> **For command line users:**
>
> The ***Search*** operation may be performed via command line using the *orthomcl_pipeline* program. For a complete reference of the available options see section 8.1. ■

5.2.1 Import data

The required input for the Search operation are proteome files (**amino acid sequences**) in Fasta format, one for each taxon. TriFusion will interpret the name of the proteome file (minus

extension) as the taxon name, so it is recommended that these files are named accordingly. The only requirements for the input files is that the headers must have one or more fields separated by a "|" symbol, and at least one of those fields must be different for all sequences. TriFusion will automatically detect that field for you and show a warning if there are no unique fields.

Consider the following example of a proteome file named *Afumigatus.fasta* that contains all annotated proteins for the *Aspergilus fumigatus* species (note that the name identifier of this file will be *Afumigatus* for TriFusion):

```
>Afumigatus|predicted protein|231
MSV(...)
>Afumigatus|predicted protein|232
MHS(...)
>Afumigatus|apolipoprotein E|233
MTA(...)
```

This is a properly formatted Fasta file with headers containing three fields, the third one being the unique ID field. Most genome projects already include protein sequence files as part of the assembly and annotation process that are compliant with this format. Data files can be loaded into the application as described in 4.1.

Command line version:

Use the *-in* option to provide the path to the directory containing the proteome files. The directory should contain ONLY the proteome files. See reference section 8.1.1.

```
1 orthomcl_pipeline -in path/to/proteome/dir (...)
```

■

5.2.2 General options

Option: *Threads*

Set the maximum number of CPUs that will be used during the All-vs-All search (the most computer intensive part of the pipeline). TriFusion was designed to use USEARCH to perform this step instead of NCBI's BLASTp, since it provides almost identical results in a fraction of time [See section on USEARCH vs. BLASTp]. To avoid specifying more CPUs than those available, TriFusion automatically detects the number of CPUs in the current system and sets them as the maximum value.

Command line version:

Use the *-np* option to specify the number of CPUs to be used in the USEARCH search operation.

```
# Perform USEARCH using 4 cpus
orthomcl_pipeline -np 4 (...)
```

■

Option: *Ortholog filters*

These filters are applied at the end of the ortholog ***Search*** for each specified inflation value (5.2.5), and determine which ortholog groups are selected according to the numbers of gene copies and taxa (Fig 5.3):

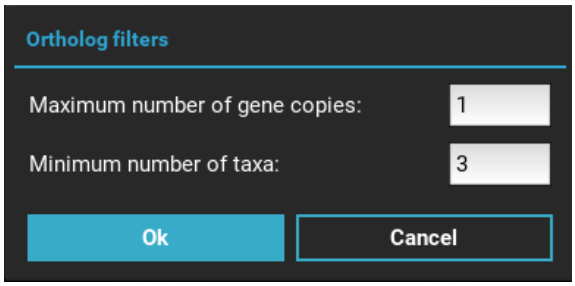


Figure 5.3: Ortholog filters general option

- **Maximum number of gene copies:** Sets the maximum number of gene copies in each ortholog group. To remove any upper limit on the number of gene copies set this value to 0. It is set to 1 by default, to include only single copy genes.
- **Minimum number of taxa:** Sets the minimum number of taxa that must be present in an ortholog group. Values can be either in absolute or in proportion (e.g. *0.5*). Once the proteome files have been loaded into trifusion, the default value of this option will be the same as the number of proteomes.

Note that this will not remove ortholog groups that do not pass these filters. TriFusion will always retain the complete results from the ***Search*** operation, but most future tasks in the **Explore** section (see Section 5.3) will only be carried out with the selected ortholog groups.

Command line version:

Use the *--max-gene-copy* option to set the maximum number of gene copies allowed per

ortholog group (see reference section 8.1.5):

```
# Allow only single copy genes
orthomcl_pipeline --max-gene-copy 1 (...)
```

Use the *--min-species* option to set the minimum number of taxa required to save an ortholog group (see reference section 8.1.5):

```
# Ortholog clusters need to have at least 25% of the taxa
orthomcl_pipeline --min-species 0.25 (...)
```

■

Option: *Output directory*

Set the directory where all orthology search results will be stored. The directory structure will follow these general rules:

- **backstage_files:** Intermediate files that are necessary during the execution of the ***Search*** operation will be stored here. They can be useful for error checking. Also contains the protein database file used for the All-vs-All search (*goodProteins_db*, by default).
- **Orthology_results:** Contains the final results of the ***Search*** operation, including the final **group** file(s). For each inflation parameter that was selected, a sub-directory will be created with the results of the corresponding parameter value and containing the sequence files of the ortholog clusters after applying the *ortholog filters* (5.2.2).

The final output directory is detailed in section 5.2.7.

Command line version:

Use the *-o* option to set the output directory (see reference section 8.1.5):

```
orthomcl_pipeline -o path/to/outputdir (...)
```

■

5.2.3 Filtering options

These include sequence quality control options.

Option: *Filter protein sequences*

Sets filters that remove poor quality sequences based on sequence length and stop codon percentage, for each input proteome file. Poor quality sequences are removed at the beginning of the ***Search*** operation and stored in *backstage_files* → *poorProteins.fasta*.

- **Minimum sequence length:** Removes amino acid sequences with less than the specified residues.
- **Maximum codon stops (%):** Removes sequences with excess of stop codons, in percentage format.

Command line version:

Use the *--min-length* and *--max-stop* options to set the minimum sequence length and maximum codon stops, respectively (see reference manual sections 8.1.4 and 8.1.4):

```
1  orthomcl_pipeline --min-length 10 --max-stop 5
```

■

5.2.4 USEARCH options

Note:

Given the licensing of the *USEARCH* software, it cannot be distributed together with TriFusion. For now, you will need to download *USEARCH* and provide it manually to TriFusion (see section 2.3). However, once provided, TriFusion will store the executable locally and it will work for all future sessions. ■

Options that are specific to the *USEARCH* All-vs-All search.

Option: *Database name*

Set the name of the protein sequence database that will be used to perform All-vs-All searches with *USEARCH*. This database will also be subsequently used to convert **group** files into protein and/or nucleotide sequences, by fetching the references of the groups clusters from the database file (see section 5.3.6).

Command line version:

Use the *--db* option to set the database name (see reference section 8.1.4).

■

Option: *USEARCH output file*

Set the name of the output file of the *USEARCH* All-vs-All search. The output of *USEARCH* follows the standard tab delimited format required for *OrthoMCL*.

Command line version:

Use the *--search-out* option to set the USEARCH output file (see reference section 8.1.4).

■

Option: *E-value cutoff*

Sets the minimum necessary Expected (e) value cut-off to retain *USEARCH* hits. This e-value parameter works in the same way as the one in the *BLAST* suite, where lower values generally provide less, but "statistically more significant" hits (More information on `https://blast.ncbi.nlm.nih.gov/blast/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=FAQ#expect`).

Command line version:

Use the *-evalue* option to set the e-value cutoff. The option accepts integers and scientific notation (see reference section 8.1.4):

```
1  orthomcl_pipeline -evalue 1e-05 (...)
```

■

5.2.5 MCL options

Note:

An *MCL* executable is bundled with the TriFusion package, and it should be readily available out of the box. However, if for some weird reason TriFusion fails to locate the *MCL* executable, you can follow the simple troubleshooting steps to provide the executable to TriFusion (see section 2.3 for a similar method). ■

Option: *Inflation*

The inflation is a parameter of the *MCL* algorithm that regulates the tightness of the ortholog clusters, with conceivable values ranging from 1.1 to 10, but rarely useful at the extremes. Using **higher** values (higher tightness) results in an increase of ortholog clusters but reduces the number of sequences included in each cluster. Conversely, lower values result in the inclusion of more sequences in fewer groups. According to the *OrthoMCL* paper [2], coarsed-grained clustering (1.5 value) provides sufficient tightness for identifying coherent ortholog clusters and

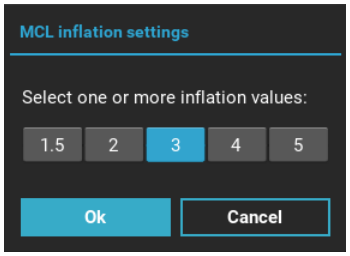


Figure 5.4: MCL inflation settings

appears to strike a balance between sensitivity and sensibility. However, they also note that this parameter may have little impact on the final number of clusters. A sensible approach that TriFusion easily allows, would be to test several values and evaluate how the final number of ortholog groups varies with this parameter.

Command line version:

Use the *-inflation* parameter to set one or more inflation values for the current run (see reference section 8.1.4):

```
# Run inflation values 2, 3 and 4
orthomcl_pipeline -inflation 2 3 4
```

■

Option: *Ortholog group prefix*

Sets the prefix for the name of the ortholog groups. This will be the prefix of the sequence files containing the sequences for each ortholog group.

Command line version:

Use the option *-prefix* to set the ortholog group prefix (see reference section 8.1.5).

■

Option: *Group file*

Sets the name of the final output of *OrthoMCL* containing the ortholog groups. If more than one inflation value was specified, this name will be used as a prefix to the inflation value. For instance, if one run was performed with inflation values of 1.5 and 3, there will be two group files named *group_1.5.txt* and *group_3.txt*.

Command line version:

Use the *--groups-file* option to set the prefix of the group files (see reference sectio 8.1.5) ■

5.2.6 Final Search report

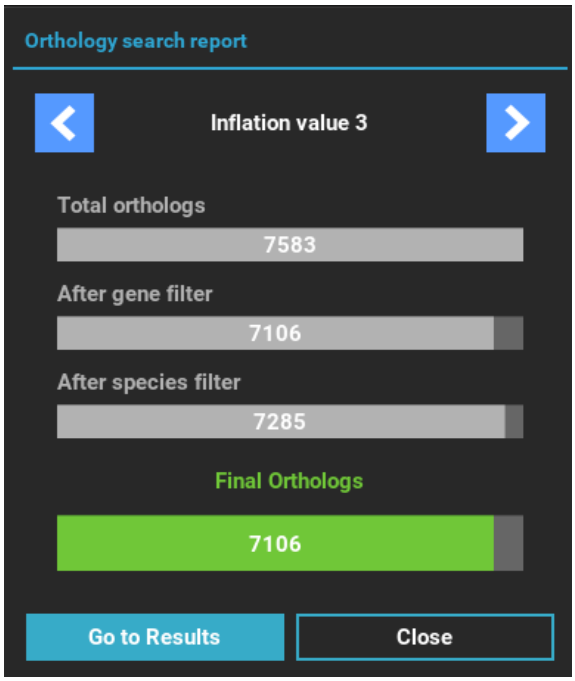


Figure 5.5: Final ***Search*** report

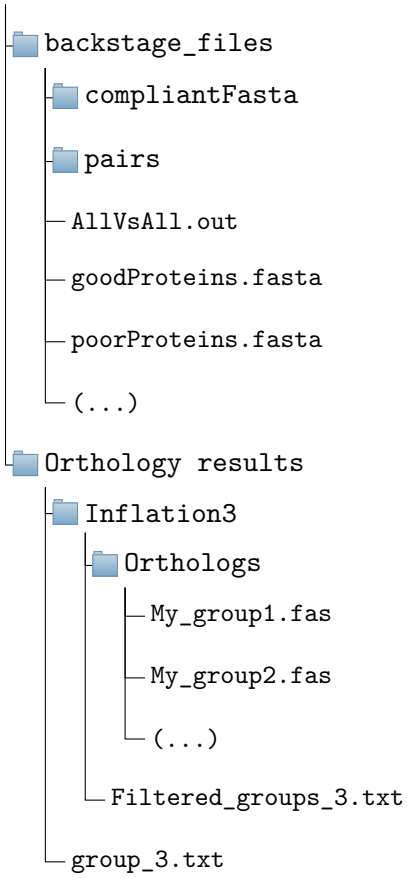
After the ***Search*** operation is complete, a report dialog pops up with summary information for each inflation value run. If multiple inflation values were selected, their results can be cycled using the top left and right key butons. For each inflation value the following information is provided in a graphical format:

- **Total orthologs:** The total number of ortholog groups identified for all species, regardless of the *ortholog filters* applied.
- **After gene filter:** The number of ortholog groups after applying ONLY the *maximum number of gene copies* filter.
- **After species filters:** The number of ortholog groups after applying ONLY the *minimum number of taxa* filter.
- **Final Orthologs:** The final number of ortholog groups after applying both filters.

After this, you can go directly to the ***Explore*** section, by clicking the Go to Results button. All main **group** output files will be automatically loaded into TriFusion and the protein database used for the conversion of the **group** files into proteins sequences will be also set.

5.2.7 Output directory structure

The typical ***Search*** run will produce the following directory structure in the *output directory* (this example depicts a run with a single inflation value of 3):



5.2.8 Main outputs

Group file

The main output of the ***Search*** operation (file *group_3.txt* in section 5.2.7) is a simple text file that contains all ortholog groups identified in the OrthoMCL and MCL analyses. Even ortholog clusters that were removed using the *ortholog filters* will be present in this file.

```
1 My_group1000: Afumigatus_proteins|433 Anidulans_proteins|4605 (...)
2 My_group1001: Afumigatus_proteins|3278 Afumigatus_proteins|9183
  ↪ (...)
3 My_group1002: Anidulans_proteins|36 Anidulans_proteins|9893 (...)
4 My_group1003: Afumigatus_proteins|1300 Afumigatus_proteins|1608
  ↪ (...)
```

Each line contains the name of the ortholog group and a list of sequence references separated by whitespace. Each reference (e.g., *Afumigatus_proteins|433*) corresponds to an actual protein sequence that is stored in the All-vs-All database file specified in section 5.2.4 using the same header. These **group** files can then be loaded into TriFusion and further investigated in the ***Explore*** tab (5.3). Alternatively, if you want to make your own filters, these two files are all you require (though we encourage the submission of new features to TriFusion).

In addition to the main **group** file, a filtered **group** file is also provided for each selected inflation value (file *Filtered_groups_3.txt* in section 5.2.7). Both files have the same format, but the latter only includes ortholog groups that pass the *ortholog filters* that were set before the ***Search*** operation.

Ortholog sequence files

Actual sequence files are stored in *Orthology results* → *Inflation** → *Orthologs* as unaligned Fasta sequences. The name of the files will be determined by the *Ortholog group prefix* option, which is *My_group* by default. The header for each individual sequence will be based on the species code used in TriFusion.

5.2.9 Secondary outputs

Protein database

The protein database file (*goodProteins.fasta* in section 5.2.7), will contain the protein sequences of all proteomes used in the ***Search*** operation. This file will be required in future session of TriFusion when converting **group** files into protein and nucleotide sequences (see section 5.3.6). This means that it is probably a good idea to save this file if you plan on re-analyzing the **group** files of the corresponding ***Search*** run.

5.2.10 How to cite

The workflow of the ***Search*** operation is adapted from *OrthoMCL* and relies on two third-party programs to work. Therefore, if you use this operation on your research, please provide the following citations as well:

- ***OrthoMCL***: Li Li, Christian J Stoeckert, and David S Roos. "OrthoMCL: identification of ortholog groups for eukaryotic genomes." In: *Genome research* 13.9 (2003), pages 2178–2189. ISSN: 1088-9051. DOI: 10.1101/gr.1224503. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=403725{\&}tool=pmcentrez{\&}rendertype=

abstract

- ***USEARCH***: Robert C Edgar. “Search and clustering orders of magnitude faster than BLAST”. in: *Bioinformatics* 26.19 (2010), pages 2460–2461. DOI: 10.1093/bioinformatics/btq461
- ***MCL***: Stijn van Dongen, Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht, May 2000.

5.3 Explore operation

5.3.1 Import data

The input data of the ***Explore*** section is the **group** file generated by either the ***Search*** operation, OrthoMCL or the command line pipeline *orthomcl_pipeline* (see section 5.2.8). One or more **group** files can be loaded by clicking the + button at the top left of the screen.

5.3.2 Explore section interface

The ***Explore*** section was designed to provide a simple and immediate visual feedback on the constitution of each ortholog group (Figure 5.6).

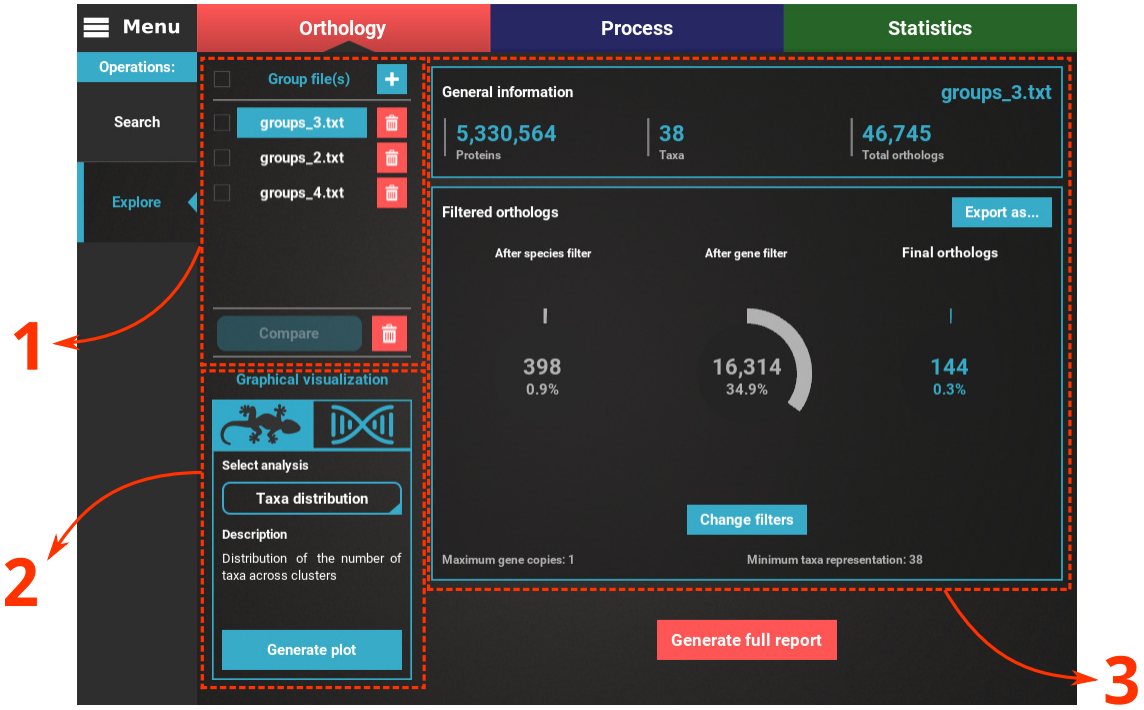


Figure 5.6: Overview of the ***Explore*** interface

The general interface of this section breaks down into three main parts:

1. Group file management

Here, you can visualize the **group** files that have been currently loaded into TriFusion, which are three in this example. At any time you can include additional **group** files using the add button (), or remove them using the remove () button. The removea all button () at the bottom of the group file list allows the removal of all **group** files. Clicking an individual **group** file will display the corresponding informative cards on the part 3 (see section 5.3.2) of the interface (the transition between informative cards of different group files is smooth and quick, regardless of the size of the group files). To compare the number of ortholog groups with or without the application of filters across multiple groups, the desired groups can be selected by clicking their corresponding check boxes. When multiple groups are selected this way, the Compare button will become available and generate a plot screen with comparative data for the selected groups (see section 5.3.3).

2. Graphical visualization

This section provides several graphical options to explore individual **group** files, focusing on *taxa* () or *orthologs* (). For each of these tabs, there are available analysis to choose from the drop-down menu. When selected, a brief description of the analysis is provided and the corresponding plot can be generated by clicking the Generate plot button. This will open a special plot screen that is described in section 5.3.4.

3. Informative cards

The informative cards provide summary information about the currently selected **group** file. The first card contains general information, such as the total number of proteins, taxa and ortholog groups regardless of the *ortholog filters*. The second card focus on the filtered orthologs. Three gauge charts present the number (and percentage) of filtered ortholog groups according to either the species, gene or both filters. Unless specified otherwise during the ***Search*** operation, the default *ortholog filter* values will be set to the maximum taxa representation value possible (ortholog groups containing all taxa) and to single copy orthologs. These filters can be changed by clicking the Change filters button (Fig 5.7).

Changing *ortholog filters* can take effect only on all **group** files, or only on the currently selected **group** file, depending on whether the *Apply filters to all group files* is checked or not, respectively. At the top right, there is also the option to export the ortholog groups that pass the currently selected filters by clicking the Export as... button (see section 5.3.6).

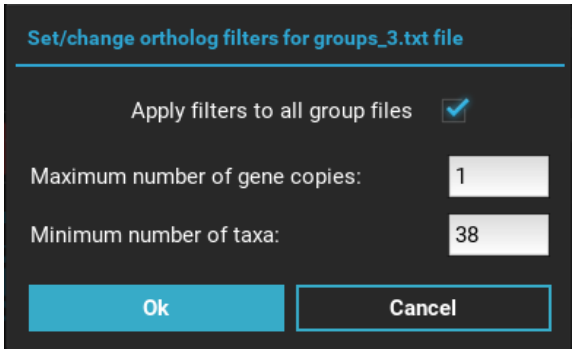


Figure 5.7: Dialog for setting ortholog filters for a group file in the ***Explore*** section

5.3.3 Comparing multiple group files

As show in section 5.3.2, multiple group files can be selected and compared by clicking the Compare button. This will open a special plot screen that displays a graphical comparison of the orthologs groups across group files. The comparison is based on a bar plot that displays the number of ortholog with different filters for each selected group file.

Figure 5.8: Dedicated plot screen for graphical comparison of multiple group files

This screen is composed of a (1) header, where several group and plot settings can be changed, (2) an interactive plot viewer, and (3) a lateral side bar.

1. **Header:** Here, filter and display settings can be changed that affect the plot immediately. Two sliders in the **Filter settings** section allow for the *ortholog filters* to be changed to new values. When the *ortholog filter* values change, the refresh button ( ) can be clicked to update the plot according to the new filters. By default, all ortholog group filter variants are displayed, but you can change which combination of the four options in the **Display** section are actually present in the plot.
2. **Interactive plot viewer:** The generated plot can be explored with some liberty. It is possible to zoom in and out (*Ctrl + mouse wheel*; or side panel zoom buttons) and to pan through the plot using the mouse.
3. **Side bar:** This sidebar is shared across all **Orthology** plot screens, and is described in section 5.3.4.

5.3.4 Graphical visualization of individual group files

Several aspects of the individual group files can be visually explored with a focus on either taxa or orthologs. As in the *group comparison* screen (section 5.3.3), the graphics are displayed in a dedicated plot screen (e.g. Fig 5.10).

Plot screen header

All plot screens in this section share the same header (Fig 5.9).

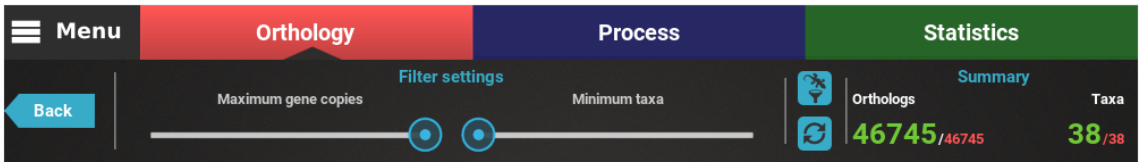


Figure 5.9: Example of the *Orthology* plot screen header

Here, the *ortholog filters* can be changed using the appropriate sliders. When the filters change relatively to the values of the current plot, clicking the refresh button ( ) will update the plot. It is also possible to exclude particular taxa from the plot analysis, by clicking the taxa filter ( ) button. All taxa are included by default, but you can toggle off any number of them and update the plot. The final section of the header is a *Summary* of the active/total number of orthologs and taxa currently being used in the analysis. Using more stringent filters will reduce the number of active orthologs, while filtering taxa will reduce the numbe of active taxa. This short *Summary* gives you an idea of the impact of the filters on the active data set and how much data is actually used when generating the plot below.

Plot screen sidebar

The plot screen side panel allows for some interaction with the plot figure and with the active data sets.

- Plot figure zoom in.
- Plot figure zoom out.
- Resizes the plot figure to the original size.
- Opens the export dialog, described in section 5.3.6, for the currently active group file.
- Exports the current plot as an image in one of several available formats. The image may be in full color, as displayed in TriFusion, or transformed into grayscale, by checking the *Grayscale* box.
- Exports the current plot in a tabular format as a .csv file.

Species focused exploration

Plot screen: *Taxa distribution*

Displays the number of orthologs in function of the exact number of unique taxa represented in a given ortholog. Here, taxa represented by multiple copies are scored only once but the range of the number of taxa is determined by the minimum taxa representation filter. In the absence of a minimum taxa representation filter, it is entirely possible to have groups with a single taxon, which represent clusters of recent paralogs exclusive to that taxon. In the example of Fig 5.10, the vast majority of the ortholog groups contain only between 1-3 taxa.

However, if we apply filters to include only single copy genes and a minimum taxa representation of ≈25%, the plot changes drastically (Fig 5.11). Considering the 3 191 filtered ortholog groups, the majority of these groups contain a high proportion of taxa.

Plot screen: *Taxa coverage*

Here, we present the proportion of actual data and missing data for each taxa (Fig 5.12). The maximum *y-axis* value for all taxa is set to the number of active ortholog groups. As the legend shows, available data is represent with dark blue while missing data is in light blue. In this context, missing data refers to the number of ortholog groups in which a certain taxa is absent, not the number of missing data characters in the protein sequence (at this point, sequences are not aligned). The horizontal dashed line represents the mean of available data for the entire data set. As can be seen, when we apply the same filters as in section 5.3.4, there is a group of taxa that is missing from quite a number of ortholog clusters. In particular, the *Popla* taxon is present

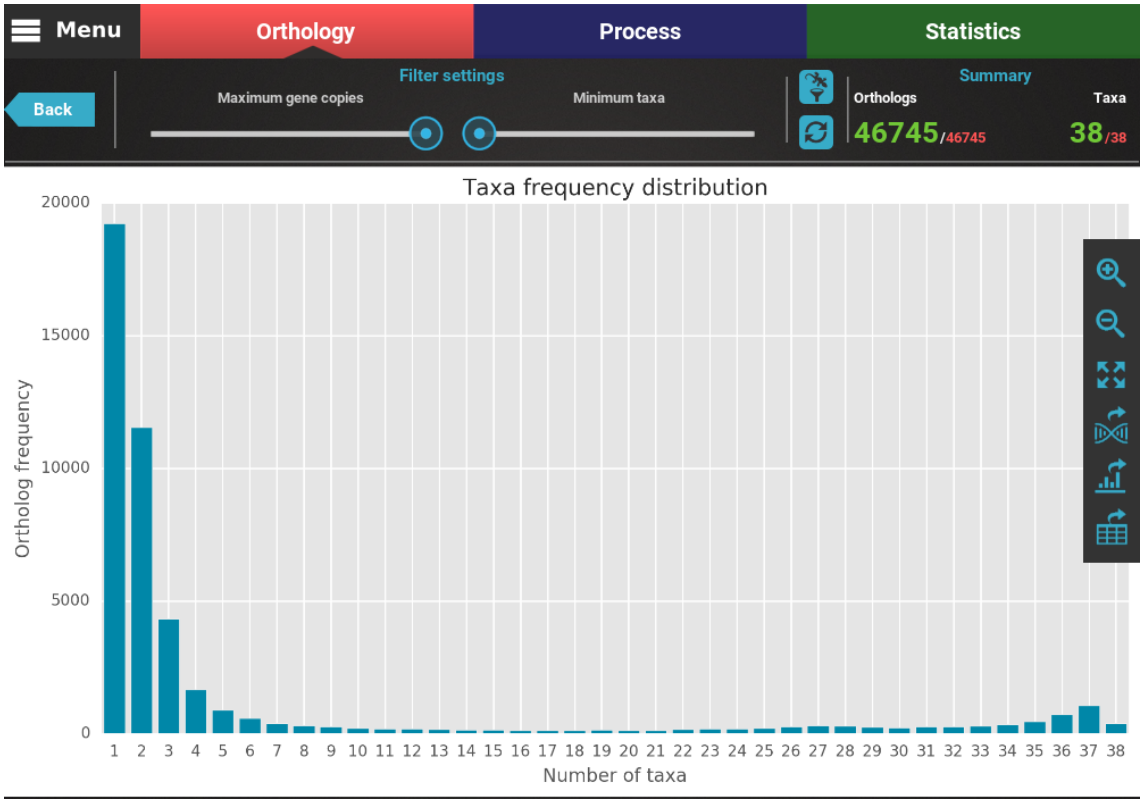


Figure 5.10: Plot screen for the taxa frequency distribution analysis

in less then half of the selected orthologs. This may indicate the presence of issues during the assembly of the orthologs, or simply the fact that some species have a lower proportion of shared orthologs due to some biological reason.

Plot screen: *Taxa gene copies*

This plot displays the number of extra gene copies for each taxon 5.14. For instance, if a given taxon has three copies in an ortholog group that means there are two extra copies. These will accumulate across the ortholog groups and taxa with higher propensity for having gene copies will appear at the right side of the distribution. If a taxon has only one copy in the entire data set, it scores 0.

Ortholog focused exploration

Plot screen: *Gene copy distribution*

Displays the number of orthologs in function of the maximum number of gene copies. Here, the number of gene copies in each ortholog cluster is obtained from the taxa with the highest number of gene copies. Therefore, if an ortholog cluster with 10 taxa has a single copy for 9 taxa and 10 copies for 1 taxon, that ortholog cluster scores a value of 10. Orthologs with a single gene

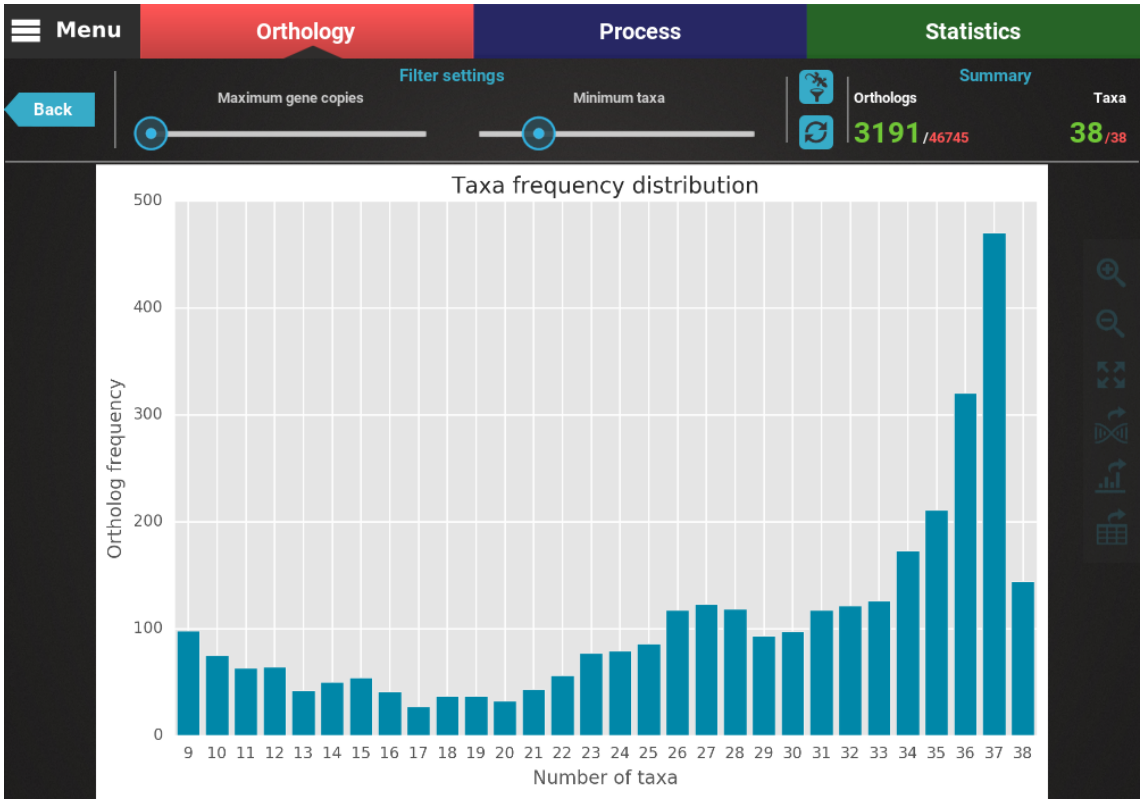


Figure 5.11: Plot screen for the taxa frequency distribution analysis after applying filters

copy effectively represent single copy genes. We recommend creating this plot without applying a maximum gene copy filter in order to get the full distribution of the data set. The example in Fig 5.14, reveals that the majority of the ortholog groups have a small number of maximum gene copies. However, there is also a large tail of ortholog clusters with a large number of gene copies.

5.3.5 Generate full *Explore* report

The complete set of graphical exploration plots can be automatically generated by clicking the Generate full report button and selecting an output directory. This will create an *HTML* report that compiles all plots, which can be visualized in any modern web browser. The individual figures can also be found in the *Figures* sub-directory, in *.png* format.

5.3.6 Exporting groups as protein or nucleotide sequences

In the group export dialog, you can export the ortholog groups in the selected **group** file and with the last set filters into three formats (Fig 5.15):

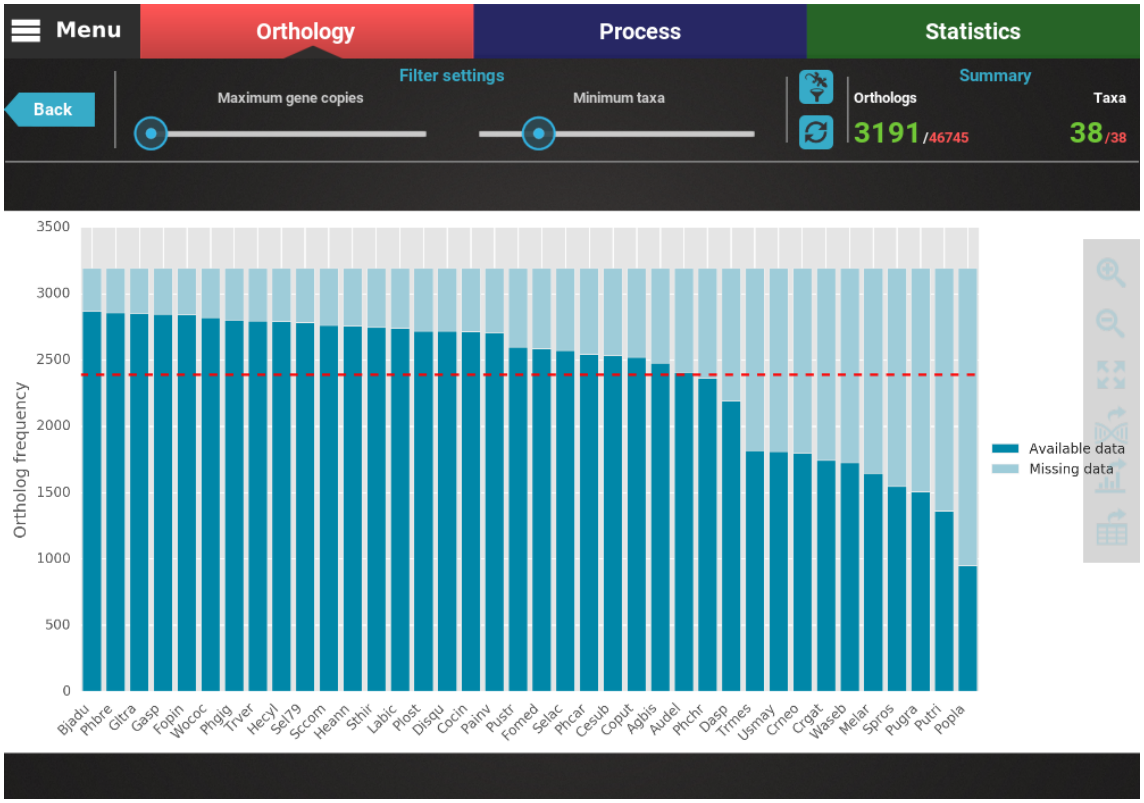


Figure 5.12: Plot screen for the taxa coverage distribution analysis after applying filters

Groups file

To export the filtered ortholog groups into a new **group** file, no additional files are required. The resulting file will have the same format as all **group** files.

Protein sequences

This exports each filtered ortholog group into a Fasta file containing protein sequence for each taxa available in the group. This will require the protein database file, which is always generated during the ***Search*** operation (section 5.2.4). If the ***Search*** operation was performed in the current session, the protein database should already be automatically set to the *database file*. Otherwise, you will have to specify the location of this file. The way TriFusion converts the group file sequence references into actual protein sequences is by fetching the references to the Fasta headers in the protein database file.

Nucleotide sequences

To export the filtered ortholog groups into Fasta files containing the coding nucleotide sequences, the protein database and a CDS database file are required, both in Fasta format. The reason why TriFusion requires both files instead of only the CDS database file is because the sequence

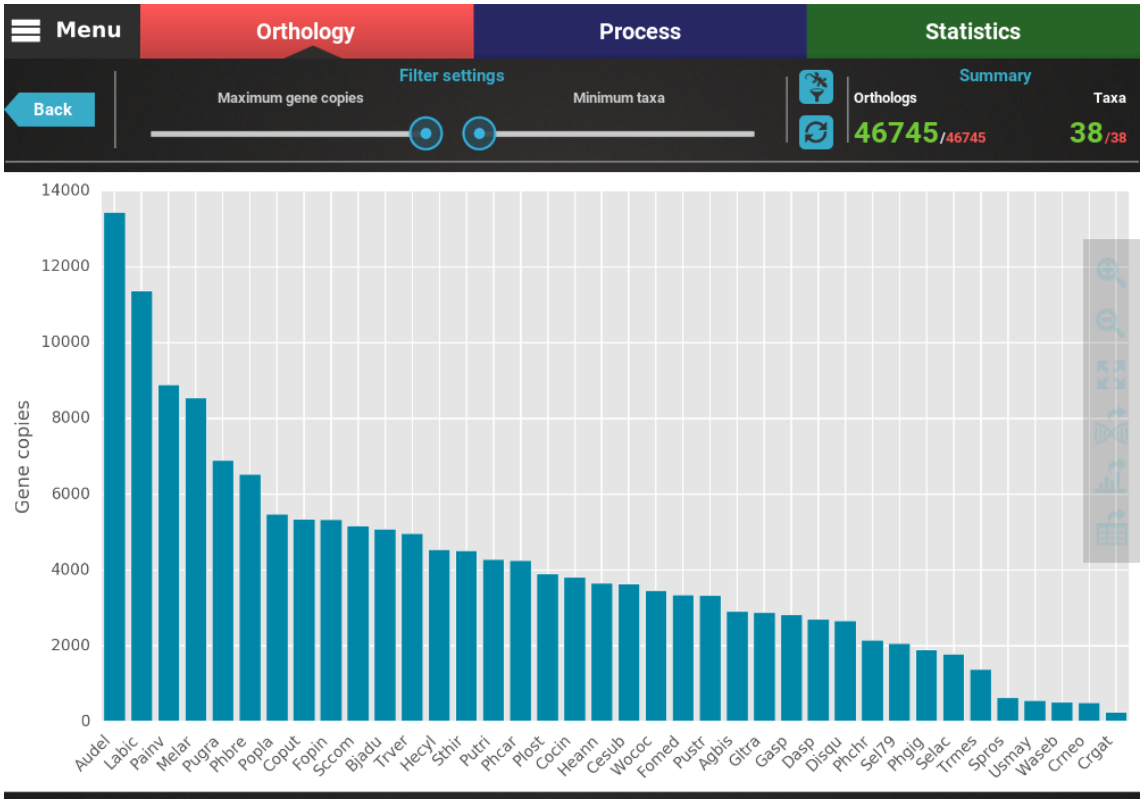


Figure 5.13: Plot screen for the taxa gene copies distribution analysis

references are not always a perfect match in both files. Consequently, instead of fetching the nucleotide sequences directly, the protein sequences are retrieved initially, as in the previous section. These sequences are compiled into a database file, and then TriFusion uses USearch to find perfect matches of nucleotide-protein searches (equivalent of the BLASTx). Finally, these perfect matches are used to populate the final nucleotide sequence files in Fasta format.

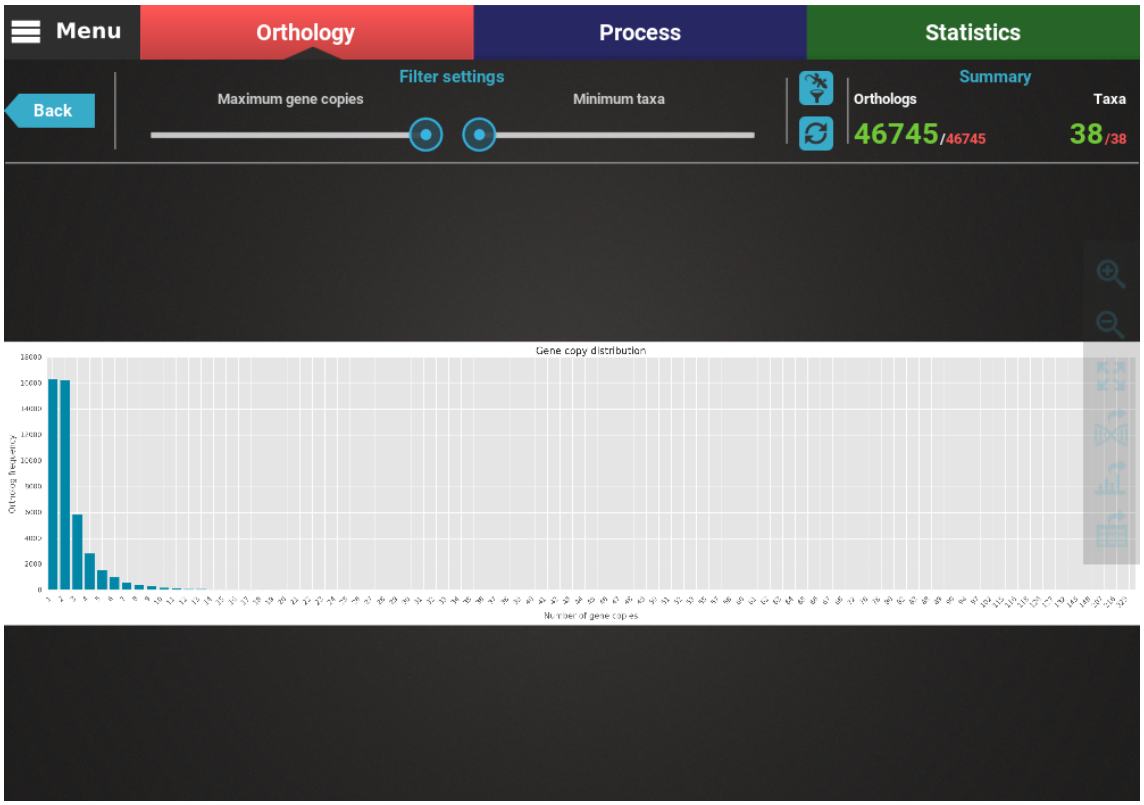


Figure 5.14: Plot screen for the gene copy distribution analysis

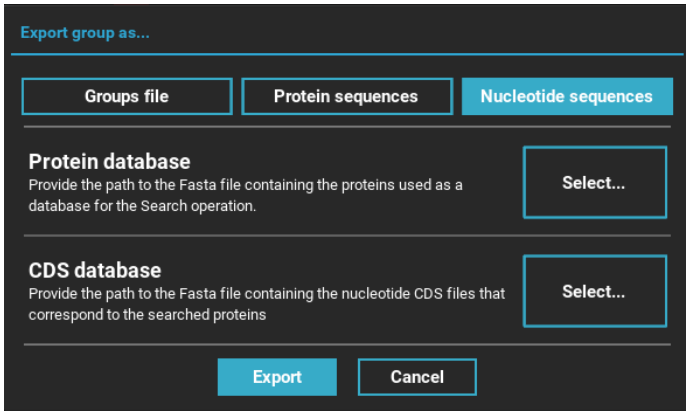


Figure 5.15: Dialog to export ortholog groups in several possible formats

6. Process module

6.1 Overview

In the **Process** module, users can **Concatenate**, **Convert** or **Reverse Concatenate** alignment files into multiple output formats. In addition to these three **main operations** (6.4), a variety of **secondary operations** (6.5) can be applied to further transform/manipulate/filter the data. For example, a set of 500 files can be concatenated (main operation) and then the resulting concatenated alignment can be collapsed (secondary operation) into unique haplotypes and filtered to remove excessive missing data - all in the same run. Multiple secondary operations can be combined to produce a single output, or additional output files can be generated for each secondary operation. Despite the numerous options available, this module was designed with simplicity in mind. Simple (but frequent) operations, such as concatenating several files into a certain output format, should require only minimal input without overwhelming the user with a barrage of options. When more complex operations are desired, the secondary operation tabs can be toggled on and further explored. At any time, so that you do not lose track of which operations are active, a list with the queue of main operations, secondary operations and expected output files is continuously updated and can be consulted by clicking the View Queue button (See 6.6).

> **Note for command line users:**
>
> Throughout this section, the options included in the **Process** module will focus mainly on

the GUI version. However, when available, the options of the command line version, **TriSeq**, will also be included. A complete reference to the command line options is described in section 9.1. ■

6.2 Input data formats

The **Process** and **Statistics** modules share the same input data types. The supported input alignment formats are:

- Fasta
- Phylip
- Nexus
- Loci (PyRAD)
- Stockholm

The input format, sequence type (nucleotide or protein) and string formatting (leave or interleave) of the provided alignment files is automatically detected by TriFusion.

Files can be loaded via the file browser, drag and dropping or terminal, as shown in section 4.1.

6.3 Output formats

After performing all Process tasks, output files can be written in several output formats

- Fasta (*.fas)
- Fasta LDhat variant (*.fas)
- Phylip (*.phy)
- Phylip MCMCTree variant (*_mcmctree.phy)
- Nexus (*.nex)
- Stockholm (*.stockholm)
- GPhoCS (*_gphocs.txt)
- IMa2 input (*_ima2.fas)

Additional settings for certain formats may be available through the settings button (). In the case of the IMa2 format, these additional settings are mandatory.

6.3.1 Additional output format settings

Fasta

- ***Produce Fasta compliant with LDhat:*** Creates a variant of the Fasta format that includes information required by LDhat in the first line.

Phylip

- ***Create partition file when concatenating:*** Creates a *.part file with the definition of partitions for the concatenated alignment. This option is ignored when not using the **Concatenation** main operation.
- **Truncate taxa names at 10 characters:** Some downstream programs require taxa names to have no more than 10 characters.

Nexus

- ***Write the partition block, if any:*** Creates a *charset* block at the end of the file with the definition of partitions for concatenated alignments.
- ***Write the substitution model, if any:*** Writes the substitution model block, compliant with MrBayes' format, if the substitution model is specified for one or more partitions (See section 4.3.2 on how to set substitution models for partitions).

IMa2

These additional options are required to produce the correct IMa2 input format (Fig 6.1).

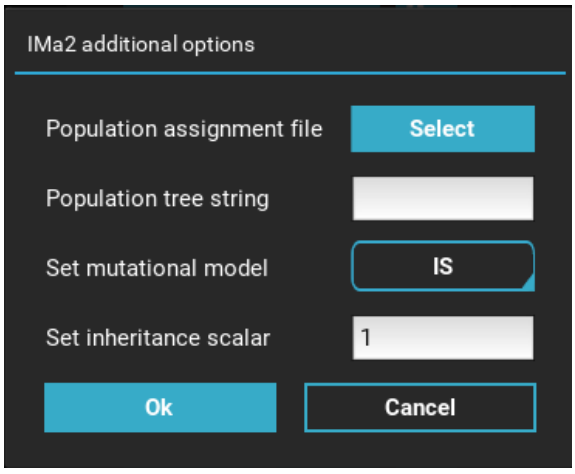


Figure 6.1: IMa2 additional options

- ***Population assignment file:*** Provide a text file mapping each taxon to its corresponding population. This file should be a simple tab delimited file with two columns, with the taxon name in the first column and population name in the second.

```
# Example of population assignment file
taxonA          pop1
taxonB          pop1
taxonC          pop2
taxonD          pop1
taxonE          pop2
taxonF          pop3
```

- ***Population tree string***: The population tree in Newick format, as specified in the IMa2 manual. An example of a tree string with three populations could be:

```
((0,1):4,3):5
```

- ***Set mutational model:*** Selects one of the available mutation models of the IMa2 software. This model will be applied to ALL partitions.
- ***Set inheritance scalar:*** The inheritance scalar is related with the chromosome of the molecular marker. As explained in the IMa2 manual, it should be 1 for autosomes, 0.75 for X-linked and 0.25 for Y-linked or mtDNA.

6.4 Main operations

The first available option in the Process module is the choice of the main operation (Fig 6.2). The **Conversion** and **Concatenation** options are readily available. The **Reverse concatenation** option can be turned on after selecting the **Concatenation** option.

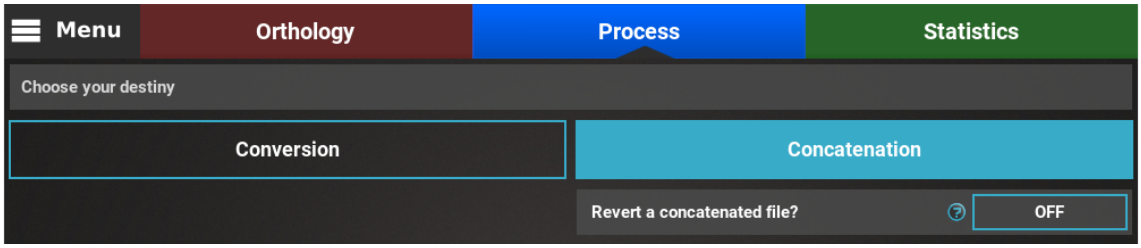


Figure 6.2: Process' main operations

6.4.1 Conversion

Conversion is the simplest operation where one or multiple input files are simply converted into other output formats. No additional changes to the files are performed, unless specified by other secondary operations.

Command line version:

Use the *-c* flag to specify conversion (reference section 9.1.2):

```
# Convert three files into phylip format
TriSeq -in file1.fas file2.fas file3.fas -of phylip -c
```

6.4.2 Concatenation

When multiple input files are provided, they can be concatenated into a single alignment file. This operation handles the merging of multiple alignments using the appropriate schemes of the output formats. For example, the concatenation of multiple files into MCMCTree format requires alignments to be concatenated in a specific way.

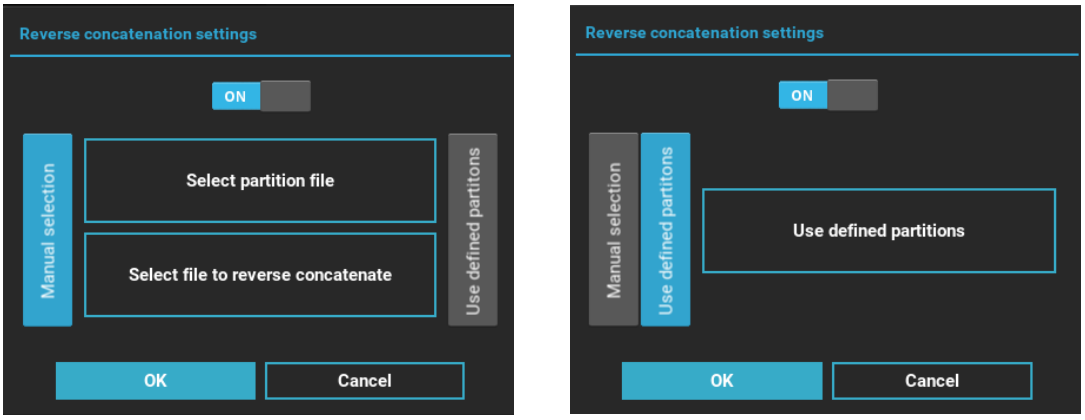
Command line version:

Concatenation is the default main operation of **TriSeq**. Therefore, no flag is required for this:

```
# Concatenate three files into phylip format
TriSeq -in file1.fas file2.fas file3.fas -of phylip -o
    ↪ my_concatenated_file
```

6.4.3 Reverse concatenation

Reverse concatenation allows partitioned alignments to revert to their individual original files. This can only be performed for **a single** partitioned alignment using one of two methods (Figure 6.3):



(a) (b)

Figure 6.3: Reverse concatenation settings for manual (a) and user defined partition (b) methods

Manual selection (Fig 6.3a)

This method can be used even when multiple files have been loaded into TriFusion. Select the *Manual selecion* tab (expanded by default). Here, specify a file to reverse concatenate by clicking the Select file to reverse concatenate button, and a partition file by clicking the Select partition file button. The partition file could be in Nexus of RAxML's partition format (see section 4.3.2 for further details on partition files). If the partition file is not in a valid format and/or the defined partitions are not consistent with the selected alignment, an error is issued.

Using partitions defined in TriFusion (Fig 6.3b)

Select the *Use defined partitions* tab. This method only works when a single partitioned alignment was loaded into TriFusion. When setting the Use defined partitions button, the partitioned alignment will be reversed according to the partitions defined inside TriFusion (See section 4.3.2 on how to create and modify partitions within TriFusione j).

Using either method, the result of this operation will be a set of individual alignments in accordance to the specified partitions. Therefore, secondary operations will be performed on this set and not on the original concatenated alignment.

> **Command line version:**
>
> Use the *-r* option, providing a partition file, to revert a concatenate file (reference section 9.1.2). The format of the partition files should be one of the two described in section 4.3.2.

```
# Reverse a concatenated file using my_part_file.txt partition
  ↪ file into individual fasta and nexus files
TriSeq -in concatenated.fas -of fasta nexus -r my_part_file.txt
```

6.4.4 General options

After selecting the main operation, there are three main options that are required and transversal to all operations 6.4.

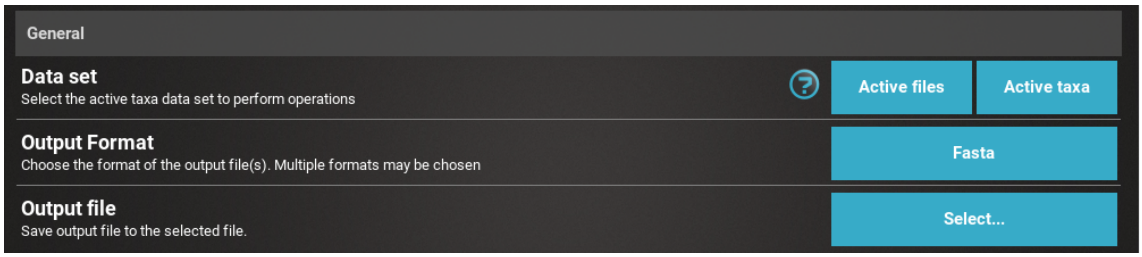


Figure 6.4: Process' main options

Data set

Determines upon which files/taxa the Process' task will be performed. Information on how to modify/create new active data sets is explained in section 4.4. By default, the *All files/taxa* and *Active files/taxa* options are available and the **active data set** is selected. If additional data set groups are created, they will also appear in their respective drop-down menus and can be selected here. When multiple data set groups of interest are created, this allows for the swift and easy execution of several tasks for different active data sets.

Usage example #1:

My currently loaded data set includes mtDNA and nuDNA. I would like to concatenate and filter the nuDNA alignments, and only concatenate the mtDNA. To achieve this in the same session, two file set groups could be created, each containing the mtDNA and nuDNA genes. Then, I could select the nuDNA file set in the *Data set* option, and perform the concatenation and filer operations. Finally, I could deactivate the filter operation and select the mtDNA file set for the concatenation only operation.

Usage example #2:

I have a large alignment data set that includes taxa from several taxonomic families, for

which I would like to reconstruct its phylogeny. However, I am also particularly interested in some specific sub-groups. After performing the desired Process tasks on the total data set, taxa sets could be created for each sub-group of interest. Then, executing the same Process tasks for each group of interest would only require the selection of the appropriate taxa set group.

■

Output format

One or multiple output formats can be selected here. Output files with the appropriate extension will be generated after executing the **Process** tasks for each selected format. Some formats may have additional settings ( ). Once set, the output format (in case only one was selected) or the number of selected formats will be displayed.

Command line version:

Use the *-of* option to specify one or more output formats (See reference section 9.1.1).

```
# Generate output in fasta, nexus and phylip formats
TriSeq -in file1.fas file2.fas file3.fas -of fasta nexus phylip
    ↪ -o my_conc_file
```

■

Output file/Output directory

Set the file (**Concatenation**) or directory (**Conversion** and **Reverse concatenation**) where the output will be generated. In the case of a **Conversion** operation, the output file names will be based on the corresponding input files. For instance, when converting a set of 5 files named "*File1.fas*", "*File2.fas*", etc., into Nexus format, the resulting output files will be "*File1.nex*", "*File2.nex*", etc. In the case of a **Reverse concatenation** operation, the output file names will be based on the partition names. For instance, reverting a concatenated file with 5 partitions named *File1*, *File2*, etc., into Nexus format will generate 5 files named *File1.nex*, *File2.nex*, etc.

Command line version:

In the case of **Conversion** or **Reverse concatenation**, the output files will always be generated in the current working directory. When using **Concatenation**, the *-o* option can be used to specify the full path of the output file.

■

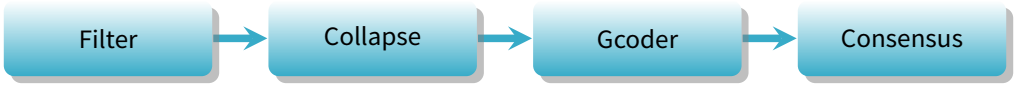
6.5 Secondary operations

Secondary operations are hidden by default but can be visualized by clicking the Show additional options button after selecting the main operation. The operations are then sorted by tabs that contain additional options and each can be switched on or off (except *Formatting*):

- **Filter**: Filters alignments according to taxa, codon position, missing data and sequence variation
- **Collapse**: Merges sequences into unique haplotypes
- **Gcoder**: (Nexus output only) Codes gaps into a binary matrix that is appended to the end of the alignment
- **Consensus**: Creates single consensus sequences for each alignment

6.5.1 Sequential order of secondary operations

When multiple secondary operations are activated, they always follow a specific sequential order:



The reason behind this ordering has to do with increased data processing efficiency, and also to prevent nonsensical combinations. For example, it makes little sense to **Collapse** an alignment and then **Filter** out the missing data. The latter will remove alignment columns according to user-specified thresholds, which could change the output of a **Collapse** operation.

6.5.2 Activating/Deactivating secondary operations

All secondary operations (except *Formatting*) can be activated/deactivated by using the ON OFF switch at the top of each operation tab. In fact, the options for each specific secondary operation will only become available when the switch is ON. Any previously set options will be ignored if the switch is OFF when the **Process'** tasks are executed.

6.5.3 Saving secondary operation output in a different file

By default, secondary operations modify the main output file(s) in a cumulative fashion (See the sequential order of secondary operations in section 6.5.1). This behavior can be modified by checking the *Save in new output file* option in the respective tab of each secondary operation (except *Formatting*). For each operation with this option checked, an additional output file will be generated that will be independent of other secondary operations.

Example:

If you activate the **Collapse**, **Consensus** and **Filter** operations during a concatenation, but check the *Save in new output file* option in the **Collapse** tab, there will be two main output files:

- *main output*: Will have the name defined in the *Output file* option with the **Filter** AND **Consensus** modifications on the concatenated alignment.
- *collapse additional output*: Will have a modified named (*_collapsed suffix) with ONLY the **Collapse** modifications on the concatenated alignment.

■

6.5.4 Formatting

This is not a secondary operation *per se* but contains formatting options for the output files.

Option: *Interleave*

Set whether the output format should be in *leave* or *interleave* format. *Leave* is highly recommended, unless downstream software requires otherwise.

Command line version:

Use the *-interleave* option in **TriSeq** (see reference section 9.1.5):

```
1 TriSeq -in *.fas -of nexus -o output -interleave
```

■

Option: *ZORRO weights support*

This option is used for **Concatenation** operations **ONLY**. It allows the user to concatenate alignment weight files that are generated by the ZORRO software [5] along with the actual alignment files.

To enable ZORRO support, turn the switch on (Fig 6.5). Then, provide the ZORRO file suffix in order to establish the correspondence between alignments and their respective weight files. ZORRO files should begin with the name of the respective alignment file and end with the provided suffix. For example, if the files *File1.fas* and *File2.fas* are to be concatenated, and we set "*_zorro*" as the suffix, the corresponding zorro weight files should be *File1_zorro.txt* and *File2_zorro.txt*. By default, TriFusion will search for the associated weight files in the same directory of the respective alignment files. If this is the case, no additional input is required in

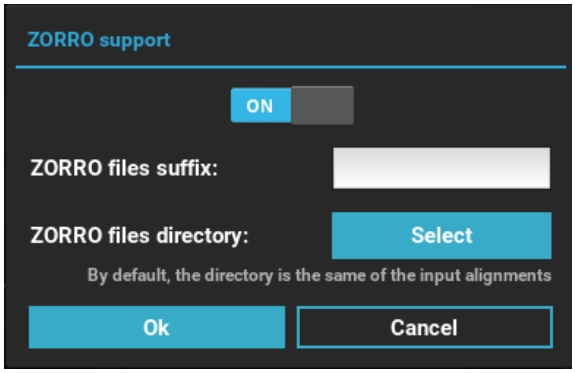


Figure 6.5: ZORRO support dialog

this dialog. However, if the weight files are stored in a different place, the correct location can be specified by selecting the appropriate *Zorro files directory*.

After selecting *OK*, and if the switch is ON, all input alignments will be checked in the appropriate directory for the presence of the corresponding ZORRO files. If one or more files have no matching weight file, an error pops up with the first missing match. The final result of the **Process'** tasks execution with this option ON will be the creation of a main concatenated alignment and an accessory ZORRO weights file, which correctly matches the weight columns to the alignment columns.

Command line version:

Use the *-z* option in **TriSeq** (reference section 9.1.5):

```
% Specifying "_zorro" as the zorro files suffix
TriSeq -in *.fas -of nexus -o output -z _zorro
```

■

6.5.5 Collapse

Flash F.A.Q.

How to activate/deactivate a secondary operation: 6.5.2 ■

The **Collapse** operation will merge all identical sequences into unique haplotypes. By default, gap and missing data symbols are considered valid characters for this operation, but this behavior can be changed (See option 6.5.5). As a consequence, taxa names will be replaced by haplotype names, which can be partially customized by the user using the *Haplotype prefix* option. The correspondence between haplotype names and the original taxa names will be written into an

auxiliary **.haplotypes* file.

As an example, the output of a collapse operation should be something like:

File1.phy

```
Hap_1 ATG
Hap_2 ATT
Hap_3 AGG
```

File1.haplotypes

```
Hap_1: taxonA; taxonC; taxonD;
Hap_2: taxonB;
Hap_3: taxonE; taxonF
```

For command line lovers:

Use the *--collapse* option in **TriSeq** (reference section **??**):

```
TriSeq -in *.fas -of nexus -o output --collapse
```

■

Option: *Save in new output file*

See section 6.5.3.

Option: *Ignore missing data*

Checking this option will remove gap and missing data characters from the alignment before collapsing (This is effectively equivalent to setting the within alignment filters to 0% of gaps/missing data percentage allowed). By default, the **Collapse** operation considers missing data symbols as valid characters. Consider the following example:

```
TaxonA        ATG
TaxonB        ATN
TaxonC        AT-
```

These sequences will be considered different haplotypes, even though they are identical in sites

with no missing data. This option restricts the assessment of sequence equality to sites that have no missing data. In cases where the proportion of missing data is low, this may prevent otherwise identical sequences to be split into different haplotypes due to the presence of a single gap. However, the final alignment will be completely stripped of sites with missing data, which may not be ideal for data sets with a high proportion of missing data.

Option: *Haplotype prefix*

Sets the prefix for the haplotype names, which will always end with "_[*number*]" (e.g. *Haplotype_1*; *Hap_1*, etc.).

6.5.6 Consensus

> ***Flash F.A.Q.***
>
> How to activate/deactivate a secondary operation: 6.5.2 ■

The **Consensus** operation will merge all sequences in an alignment into a single consensus sequence. Alignment columns with variation can be handled in multiple ways. They can be either converted into the corresponding IUPAC symbols (nucleotide sequences only), soft masked (replaced with missing data) or removed. Alternatively, the first sequence of each alignment can be selected as a representative.

This operation is useful when only a single representative sequence of an alignment is desired for further analyses, such as input for functional annotation software.

> **Command line version:**
>
> Use the *--consensus* option, providing a variation handling method (reference section 9.1.3):
>
> ```
> 1 # Create consensus sequences, soft masking variation into
>   ↪ missing data
> 2 TriSeq -in file1.fas -of fasta --consensus "soft mask"
> ```
>
> ■

Option: *Save in new output file*

See section 6.5.3.

Option: *Save consensus sequences in a single file*

By default, an output file with a single consensus sequence is generated for each alignment. (**Note:** if the main operation is the **Concatenation** of multiple files, the **Consensus** will be applied to the single concatenated alignment, not the original files). By checking this option, all consensus sequences are combined in a single output file instead.

Command line version:

Use the *--consensus-single-file* flag (reference section 9.1.3):

```
# Create consensus sequences, soft masking variation into
↪ missing data
TriSeq -in file1.fas file2.fas file3.fas -of fasta --consensus
↪ "soft mask" --consensus-single-file
```

■

Option: *Consensus variation handling*

Sequence variation within the multiple sequence alignment can be handled using one of the following options:

- **IUPAC:** (Nucleotide sequences only) Converts variable sites using the IUPAC nucleotide ambiguity codes (e.g. a column with *C* and *T* variants will appear with a *Y* character in the consensus).
- **Soft mask:** Replaces variable sites with the missing data symbol (*N* for nucleotides and *X* for protein sequences).
- **Remove:** Removes variable sites from the consensus.
- **First sequence:** An arbitrary method that simply uses the first sequence without modification.

6.5.7 Filter

Flash F.A.Q.

How to activate/deactivate a secondary operation: 6.5.2 ■

This operation includes several options that allow the filtering of alignments and/or alignment columns. Broadly, this filtering can be done in accordance to taxa, codon position, missing data and sequence variation. Any combination of filters can be used, which means that besides activating the **Filter** secondary operation, users must also activate the individual desired filters

in their respective dialog boxes.

Option: *Save in new output file*

See section 6.5.3.

Option: *Taxa filter*

This option filters entire alignments depending on whether they contain or exclude a given set of taxa (Fig 6.6).

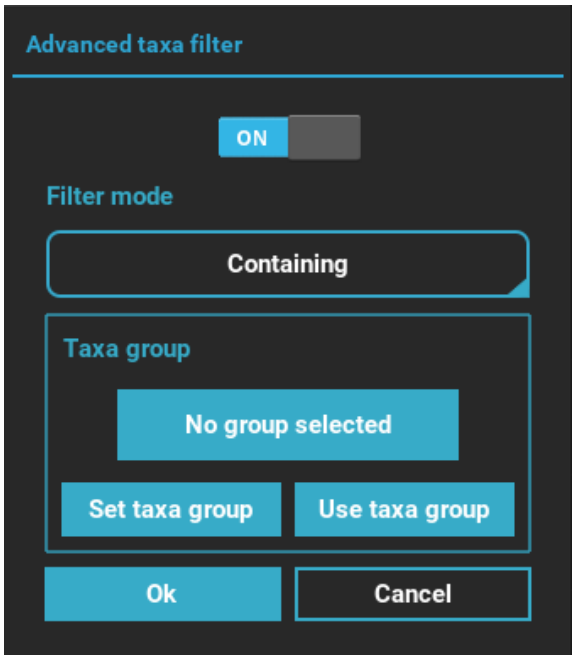


Figure 6.6: Taxa filter dialog

When choosing the *Containing* filter mode, only those alignments containing **at least** the taxa specified in the taxa group will be process. When choosing the *Exclude* filter mode, alignments that contain **all** taxa specified in the taxa goup will be excluded.

If the desired taxa group has already been created, it can be selected by clicking the Use taxa group button, which will list all user defined groups. When a group is selected, the text of the No group selected button in Fig 6.6 will change to display the group name.

If the desired taxa group does not yet exit, it can be created by clicking Set taxa group. The group creation can be done as described in section 4.4.3. The last created group will become automatically selected.

> **Command line version:**
>
> Use the *--contain-taxa* option to select alignments containing the specified taxa and the

--exclude-taxa to filter out alignments containing the specificed taxa (see reference sections 9.1.3 and 9.1.3):

```
# Only process alignments containing the specified taxa
TriSeq -in *.fas -of fasta -o output_file --contain-taxa taxon1
    ↪ taxon3 taxon5
# Only process alignments that DO NOT contain the specified taxa
TriSeq -in *.fas -of fasta -o output_file --exclude-taxa taxon2
    ↪ taxon10
```

■

Option: *Codon position filter*

Note: ONLY available for nucleotide sequences. ■

This option allows the filtering of specific codon positions in alignments (Fig 6.7).

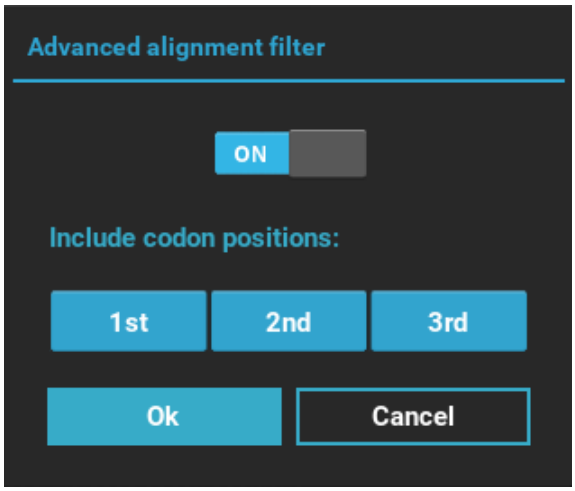


Figure 6.7: Codon filter dialog

By default, all positions will be included. To exclude particular position, simply toggle the respective buttons.

Command line version:

Use the *--codon-filter* option to select which codon position will be processed (see reference section 9.1.3):

```
#Process only the first and second codon position
TriSeq -in file1.fas -of fasta --codon-filter "1 2"
```

Option: ***Gap/Missing data filter***

This filter can be applied at two levels: **within alignments** (filters columns); and **among multiple alignments** (filters entire alignments) (Fig 6.8). They can be set independently by checking/unchecking their respective check boxes, that is, you can perform only one of these filters instead of both.

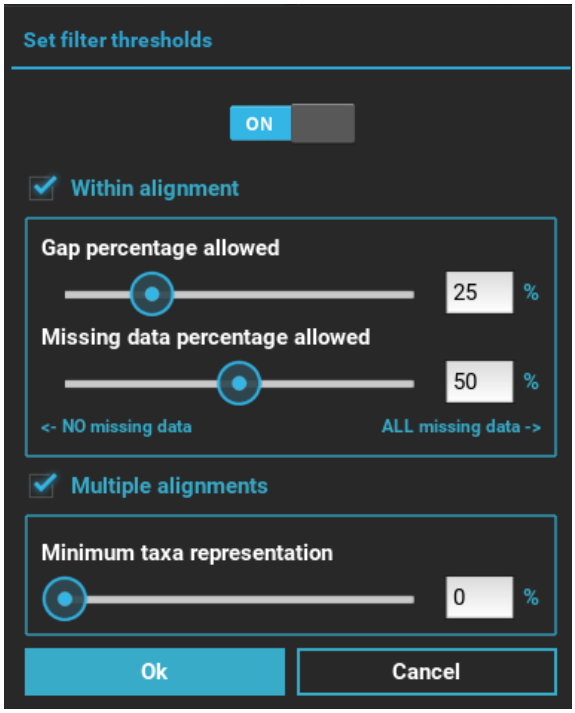


Figure 6.8: Gap/Missing data filter dialog

- **Within alignment:** When checked, two actions are performed.
  - (***This option is performed automatically when this filter is active***) Replaces the gap symbols in the extremities of the alignment with missing data symbols. For most downstream analyses, this modification will be inconsequential because gaps/missing data are usually ignored. Nevertheless, the gaps in the extremities are technically missing data and not indels.
  - The *Gap percentage allowed* and *Missing data percentage allowed* sliders define maximum thresholds for the presence of gaps and missing data in alignment columns,

respectively. Gaps refer to only indels ("-", for example) while missing data refers to the combination of indels and actual missing data ("N" or "X" symbols). Columns with either gap or missing data percentage above one of the defined thresholds are removed from the alignment.

To illustrate, consider the following example:

```
# Before filter
# Positions   123456
taxonA        -ANT-T
taxonB        -ANATT
taxonC        -ANA-T
taxonD        -TNTTT
taxonE        -ACT-T
taxonF        -AN-TT

# After filter (first position converted to Ns and 2 columns removed)
# Gap percentage allowed: 25%
# Missing data percentage allowed: 50%
# Positions   1246
taxonA        NATT
taxonB        NAAT
taxonC        NAAT
taxonD        NTTT
taxonE        NATT
taxonF        NA-T
```

- **Multiple alignments:**
  - The *minimum taxa representation* filters sets the minimum proportion of taxa required for an alignment to be further processed. For instance, if the complete data set has 10 taxa and the minimum taxa representation is 50%, then only alignments with at least five taxa will be saved.

Command line version:

Use the *--missing-filter* option for the within alignment filtering, providing the maximum gap and missing data percentage (see reference section 9.1.3):

```
# Filter columns with more than 50% gaps and 75% missing data
TriSeq -in *.fasta -of nexus -o output_file --missing-filter 50
↪ 75
```

Use the *--min-taxa* option for the multiple alignments filter, providing the minimum taxa representation percentage (see reference section 9.1.3):

```
# Filter alignments with less than 25% of the total taxa
TriSeq -in *.fasta -of nexus -o output_file --min-taxa 25
```

■

Option: *Sequence variation filter*

This option allows entire alignments to be filtered according to the number of columns (sites) with a certain type of variation. Here, we consider two types of variation:

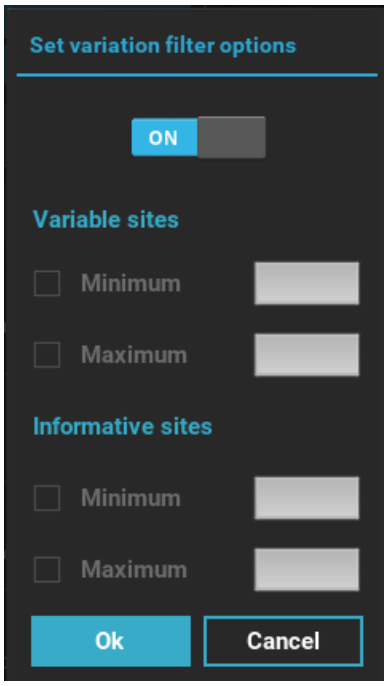


Figure 6.9: Variation filter dialog.

- **Variable sites**: Refers to all variable sites regardless of the number of taxa with the alternative SNP (e.g. *Column:* A A A A T A C).
- **Informative sites**: Refers only to variable sites in which at least one of the alternative SNPs is present in two or more taxa (e.g. *Column:* A A A T T A A).

In either type, *minimum* and *maximum* values for the number of sites with sequence variation can be set by checking the appropriate check box and providing a value. If a box is left unchecked, it is assumed that there will be no lower nor upper threshold depending on whether it refers to the *minimum* or *maximum* options, respectively. For example, checking the *minimum* variable sites with a value of 2 and leaving the *maximum* option unchecked will only filter alignments with one or less variable sites. There would be no maximum limit of variable sites. If the *maximum* was later checked and set to 10, alignments with 11 or more variable sites would also be filtered.

Use the *--variable-filter* option to filter variable sites, providing the minimum and maximum range values (see reference section 9.1.3):

```
# Filter alignment with less than 2 and more than 10 variable sites
TriSeq -in *.fas -of nexus -o output_file --variable-filter 3 10
```

Use the *--informative-filter* option to filter informative sites, providing the minimum and maximum range values (see reference section 9.1.3)

```
# Filter alignment with less than 2 informative sites
TriSeq -in *.fas -of nexus -o output_file --informative-filter 3 None
```

■

Final report on filtered alignments

After applying filtering options that remove alignments during the execution of **Process'** tasks, a filter report dialog is shown at the end of the run informing how many alignment were removed 6.10. The total number of removed alignments is displayed at the bottom of the report. Additionally, this information is also discriminated individually for each of the four filters that are able to remove alignments. If a given filter was not activated, a *Not applied* message is provided. Otherwise, the number of alignments removed by that specific filter is displayed. This

information can be useful to assess which filters are removing more alignments.

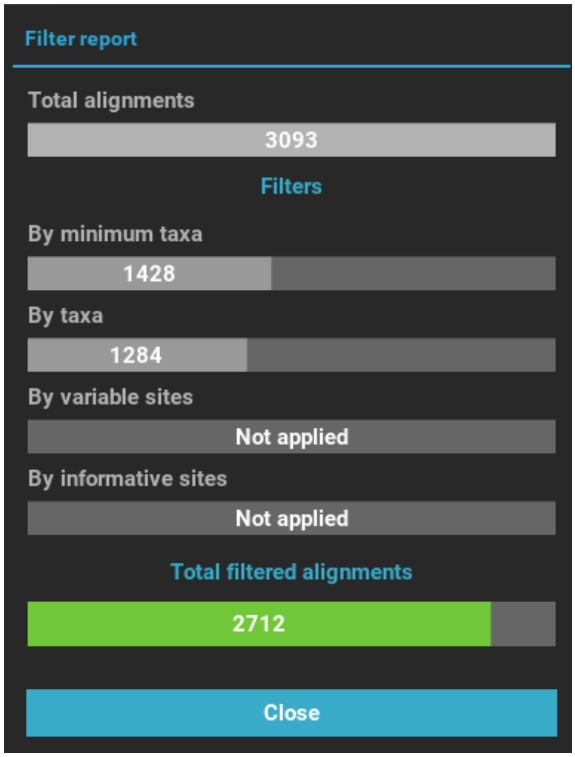


Figure 6.10: Final filter report informing how many alignments were filter and by which active filters.

6.5.8 Gcoder

> ***Flash F.A.Q.***
>
> How to activate/deactivate a secondary operation: 6.5.2 ■

The **Gcoder** option will transform the information of indel patterns contained within an alignment into a binary matrix that will be appended to the end of the alignment. **This option can only be set when Nexus is the sole output format**.

> **Command line version:**
>
> Use the *--gcoder* option to code gaps at the end of the alignment matrix (see reference section 9.1.3):
>
> ```
> 1  TriSeq -in *.fasta -of nexus --gcoder -o output_file
> ```
>
> ■

Option: ***Save in new output file***

See section 6.5.3.

Option: ***Gap coding method***

Currently, the only gap codification method implemented in TriFusion is the one proposed in Simmons and Ochoterena [4].

6.6 Process operations queue

Many options in the **Process** module can be simultaneously active in a single session and it can be easy to get lost. As a convenience, you can easily keep track of the active main and secondary operations, output formats and expected number of output files by clicking the Viewe Queue button at the bottom of the **Process** screen.

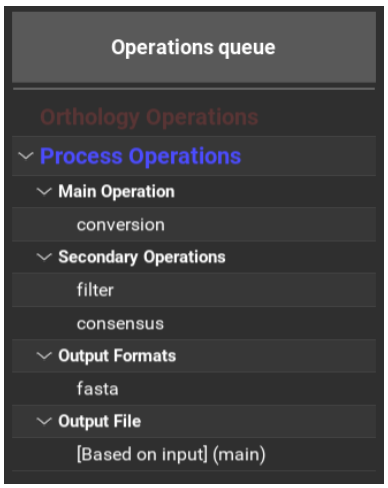


Figure 6.11: Process queue tab

This will open the side panel in the *Operations Queue* tab, where the information will be sorted by *Main operation*, *Secondary operations*, *Ouput Formats* and *Output File*. When the main operation is **Conversion** or **Reverse concatenation**, the name of the output file(s) will be based on the input or partitions, as shown in Fig 6.11. In the case of **Concatenation**, the defined output file will be displayed. For every secondary operation with the *Save in new output file* option checked, an additional output file with the appropriate suffix will also appear.

7. Statistics module

7.1 Overview

The **Statistics** module provides a set of graphical and statistical analyses that allow you to explore different facets of your alignment data. Since a picture may be worth a thousand words, most of these analyses generate a plot of some kind that allows you to quickly an easily visualize the data. These analyses were optimized for large alignment data sets, which are also the most difficult to explore in an efficient and timely fashion (nevertheless, most of these analyses will also prove useful for smaller data sets). The numerous options are sorted into thematic categories that focus on a particular aspect of the data, such as *polymorphism and variation* or *missing data*, for example. Moreover, each individual option can also have several plot types. For instance, measures of pairwise sequence similarity can be calculate with a sliding window across a single gene, between pairs of taxa, or as an average from the entire alignment data set. This ensures a greater flexibility for the user when exploring a specific aspect of the data.

In short, the main goal of this module is to get you acquainted (and even intimate) with your phylogenomic data, particularly if its big data.

> **Note for command line users:**
>
> Throughout this section, the options included in the **Statistics** module will focus mainly on the GUI version. However, when available, the options of the command line version, **TriStats**, will also be included. A complete reference to the command line options is described

in section 10.1 ■

7.2 Import data

As previously stated, this module deals with the same input data as the **Process** module, that is, multiple sequence alignments that can be in one of several supported formats (see section 6.2). Input alignment files can be loaded into TriFusion as described in section 4.1.

Command line version:

Like in **TriSeq**, use the *-in* option to provide input data to **TriStats**. ■

7.3 Visualizing summary statistics

When the appropriate data is loaded into TriFusion and you switch to the **Statistics** screen, the first task it will undertake is the calculation of several summary statistics. If the input data is large enough this computation may take some time to finish but do not fret - it will be doing so in the background (unless you cancel it). You may continue to perform other tasks in TriFusion in parallel, and return to the *Summary statistics overview* panel at any time by clicking the Σ button. When this task is completed, two informative panels will become available: (i) an overview of the summary statistics across the entire active data set, and (ii) a gene table listing these summary statistics discriminated for each alignment. By default, the summary statistics overview panel is displayed first and the gene table can be viewed by clicking the Display gene table button at the bottom of the screen. In either case, the summary statistics can be exported in table format by clicking the Export as table button.

7.3.1 Summary statistics overview panel

An example of the summary statistics overview panel is shown in Fig 7.1. This overview is sorted into three categories:

General information

Contains basic general information, like the number of genes, taxa and the total length of the alignment in nucleotides or residues in the active data set. This section is usually useful to confirm that the appropriate file/taxa active set is selected.

Missing data information

Displays the information about gaps and missing data across the active data set. This is shown in the form of the total number of gaps and missing data across the alignments, along with the

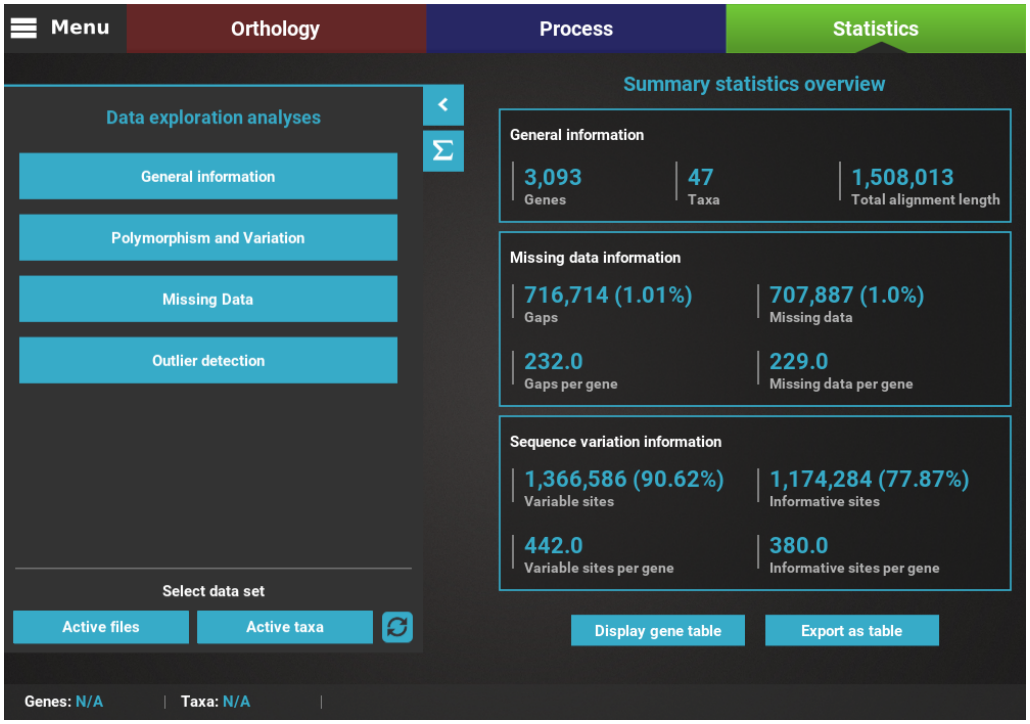


Figure 7.1: Example of the summary statistics overview in the **Statistics** screen.

percentage relative to the total alignment length. Additionally, the average of gaps and missing data per alignment is also shown.

Sequence variation information

Calculates the amount of sequence variation in the active data set. Variable sites refer to alignment columns with at least one variant, while the informative sites refer to alignment columns where an alternative allele is present in at least two taxa. As in the *Missing data information* category, this information is show as a total value or as an average for each alignment.

7.3.2 Summary statistics gene table view

In this view, roughly the same summary statistics of the previous section are displayed but discriminated for each input alignment as rows. The same three categories of general, missing data and variation information are color coded in the table to facilitate visualization. A legend at the top informs the meaning of the codes in the table headers.

Due to performance issues, only the first 50 alignments are shown but additional alignments can be added to the list sequentially by clicking the Show more 25 button at the end of the table. However, for larger data sets it may be better to export this as a gene table and continue the exploration in MS Excel or LibreOffice's Calc.

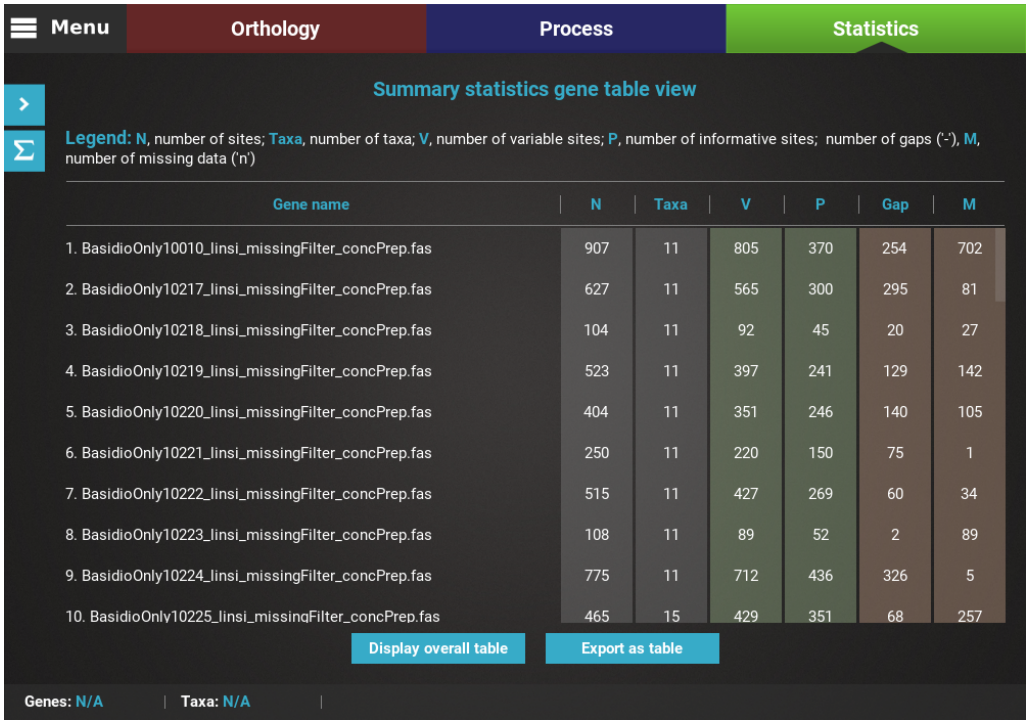


Figure 7.2: Example of the gene table listing of summary statistics in the **Statistics** screen.

7.4 Data exploration analyses

The majority of the **Statistics**' module analyses are contained within the *Data exploration analysis* side panel and sorted across four main categories: *General information*, *Polymorphism and Variation*, *Missing data* and *Outlier detection*. Clicking any of these categories buttons will show all the available individual analyses. Coupled with each analysis is an information (i) button that provides a description of the analysis, available plot types and what the axis represent in the generate plot.

7.4.1 The three plot types: *Single Gene*, *Per Species* and *Average*

In most cases, clicking an option button will ask for a plot type (Fig 7.4). There are up to three available plot types, *single gene*, *per species* and *average*, each of which provides a different perspective to the same analysis. **Note that not all options have the three plot types available**. Such cases occur when a particular plot type does not make sense for that analysis. In certain cases, plots can be very specific, such as the analyses *Alignment length/Polymorphism correlation* or *Gene occupancy*, in which case there is only one plot type available.

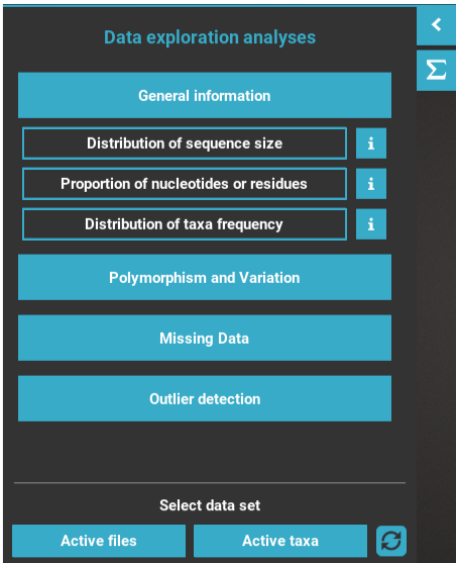


Figure 7.3: Side panel of the **Statistics** screen showing the *Data exploration analyses*.

Figure 7.4: Dialog for the selection of plot types.

Single gene

Single gene plot types are always sliding window analysis of a certain metric across the length of a given alignment. When this plot type is selected, you must choose a single gene of interest (Fig 7.5) and you may change the default sliding window size of 10. As a convenience, particularly for large data sets, there is a search field to find particular files.

Per species

This plot type focus the data visualization on the individual taxa that comprise the active data set. This usually means that plots distribute taxa along the *x-axis*, or along triangular heat matrices that are constructed from pair-wise comparisons across the alignment data set.

Average

In the *average* plots, the analysis focuses on the average value of a certain metric across the entire alignment data set. These plot types are usually more diverse but they tend to give a broad distribution of metrics averaged across the entire data set.

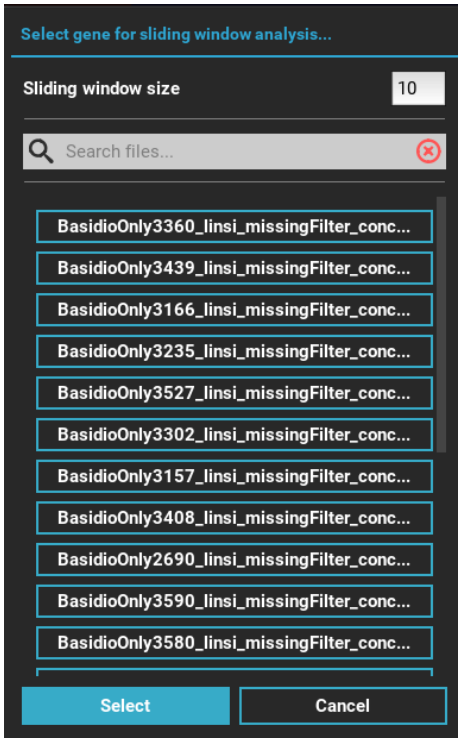


Figure 7.5: Dialog for the selection of sliding window size and gene of interest for *single gene* plot type.

7.4.2 Updating the active data set

By default, all **Statistics** analyses are performed on the *active data set* (see more in section 4.4 about total, active and custom data sets). However, you can change this by clicking on the drop-down menus at the bottom of the **Statistics**'s side panel and choose the data set you wish to analyze. As in the **Process** module, custom file and taxa groups can be created (see section 4.4.3) and selected in these menus. After selecting the appropriate data sets, the currently selected plot analyses can be refreshed by clicking the refresh ( ) button. Changes in the active data set being used can be observed in the *Genes* and *Taxa* counters at the bottom of the **Statistics** screen.

7.4.3 Interactive plot viewer interface

After selecting a particular analysis and plot type, the generated plots are displayed in an interactive viewer (Fig 7.6), with several similarities to the plot viewers of the **Orthology** module (section 5.3.4). It is possible to zoom in and out (*Ctrl* + *mouse wheel*) and pan through the plot using the mouse. The plot sidebar is also very similar (section 5.3.4). When there are multiple plot types available for the selected analysis, a box appears in the top right corner of the screen

where the plot types for the current analysis can be quickly switched.

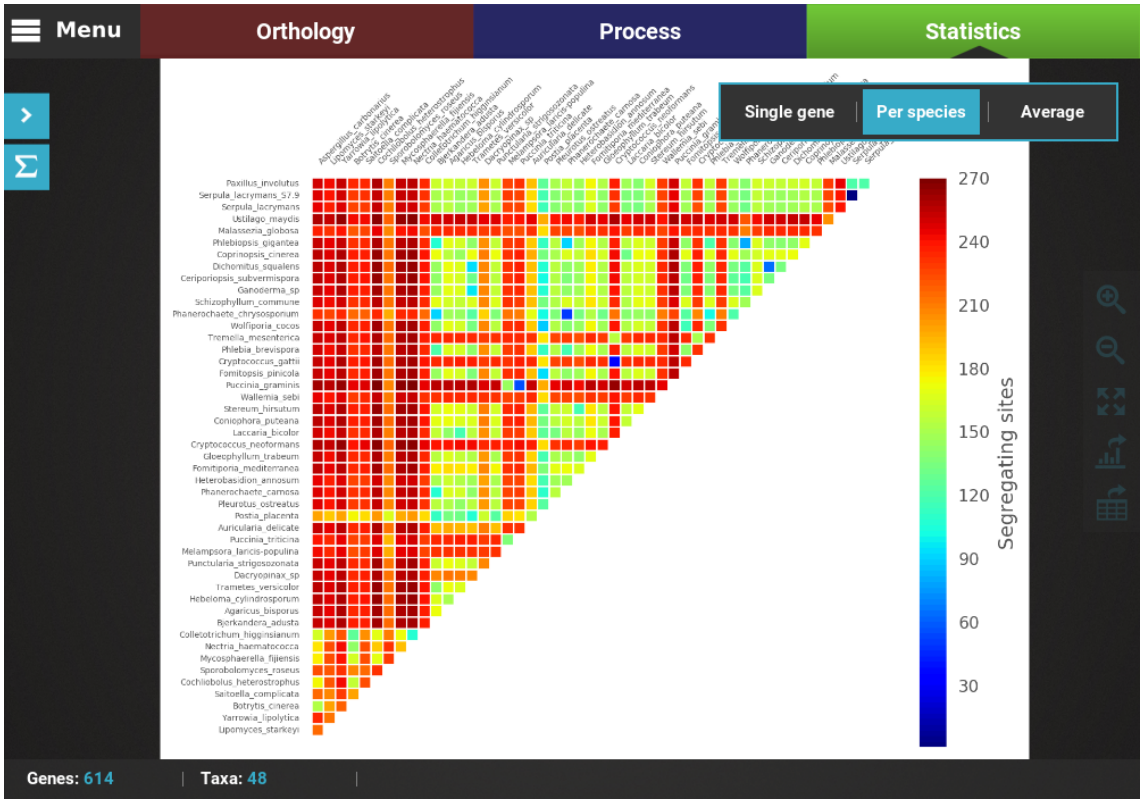


Figure 7.6: Example of the **Statistics**' plot viewer interface.

7.4.4 Plot analyses

General information

Statistics analysis: ***Distribution of sequence size - Per species***

Plots the distribution of the average sequence size (excluding sites with gaps or missing data) for each species. This distribution is provided as a whisker plot, where the box limits represent the Q1 and Q3 quartiles.

y-axis: Sequence size in nucleotides (DNA) or residues (Protein).

x-axis: Taxon name.

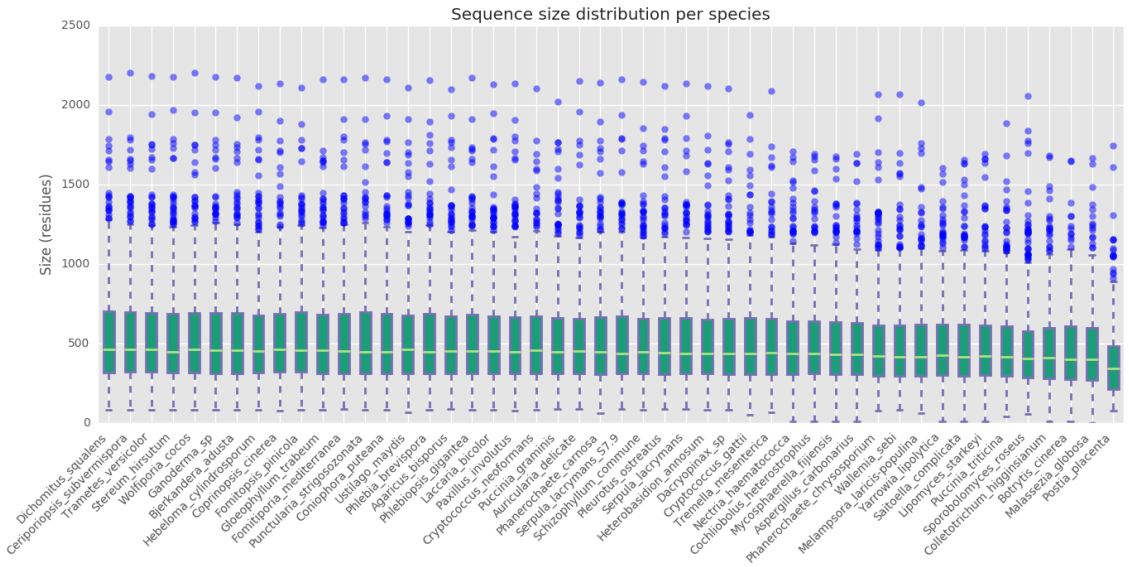


Figure 7.7: Example of distribution of sequences sizes per species.

Statistics analysis: ***Distribution of sequence size - Average***

Plots the distribution of the average sequence size (excluding sites with gaps or missing data) for the entire data set. The vertical dashed line represents the data set's mean.

y-axis: Absolute frequency of alignments.

x-axis: Sequence size in nucleotides (DNA) or residues (Protein).

> **Command line version:**
>
> These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

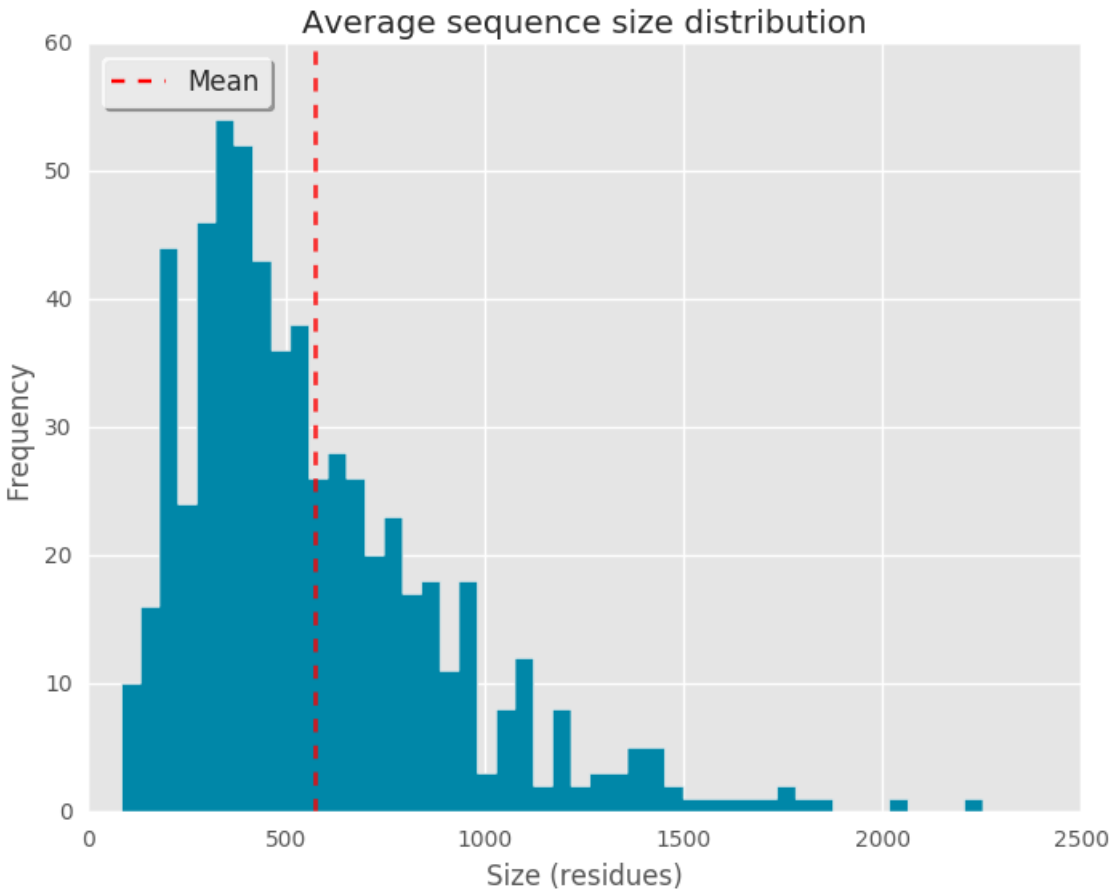


Figure 7.8: Example of distribution of sequences sizes averaged across the data set.

```
[General Information]
# Options available: species average
distribution_sequence_size: species average
```

Statistics analysis: ***Proportion of nucleotides or residues - Per species***

A stacked bar plot with the proportions of each nucleotide (DNA) or residue (Protein) for each species. A sorted color legend is provided above the plot. In the case of residues, since there are many possible variants, the color code is repeated once, but the order in the legend and plot are the same. For instance, in Fig 7.9, the bottom red bar corresponds to the first residue in the legend (*A*), while the second red corresponds to the later *P* residue.

y-axis: Proportion of nucleotide or residue.

x-axis: Taxon name.

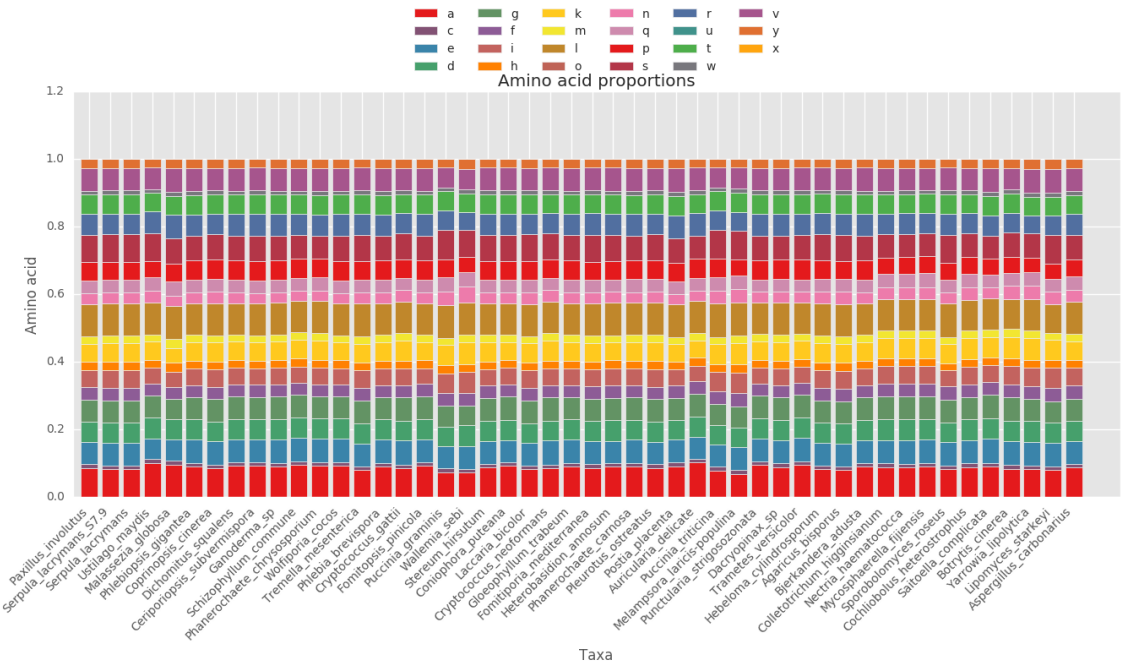


Figure 7.9: Example of the distribution of residue proportions per species.

Statistics analysis: ***Proportion of nucleotides or residues - Average***

Bar plot with the average proportions for each nucleotide or residue across the alignment data set (Fig 7.10).

y-axis: Proportion.

x-axis: Nucleotide or residue.

> **Command line version:**
>
> These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[General Information]
# Options available: species average
proportion_nucleotides_residues: species average
```

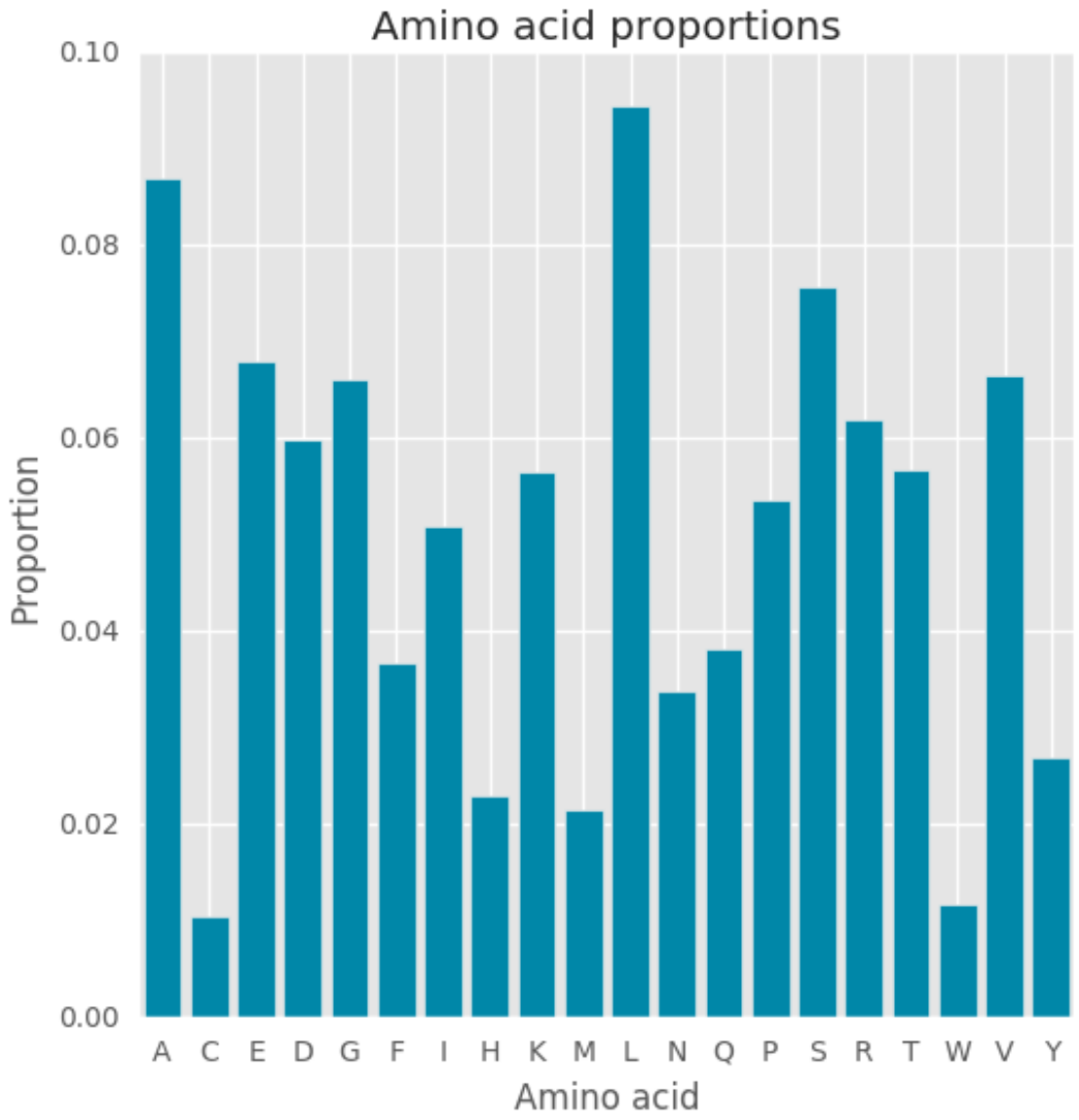


Figure 7.10: Example of the proportion of residues averaged across the data set.

Statistics analysis: ***Distribution of taxa frequency - Average***

A histogram with the frequency of alignments containing a given number of taxa. The vertical dashed line represents the mean of the entire data set. This plot can be useful is large data sets to evaluate the number of genes containing many or few taxa. Dense data sets (few missing taxa) tend to have distributions shifted to the right side, while sparse data sets tend to have distributions shifted to the left side. Considering the plot in Fig 7.11, we can see that the data set has relatively few genes missing in most of the taxa, since the distribution is almost abutting the highest value.

y-axis: Absolute frequency of alignments.

x-axis: Number of taxa present.

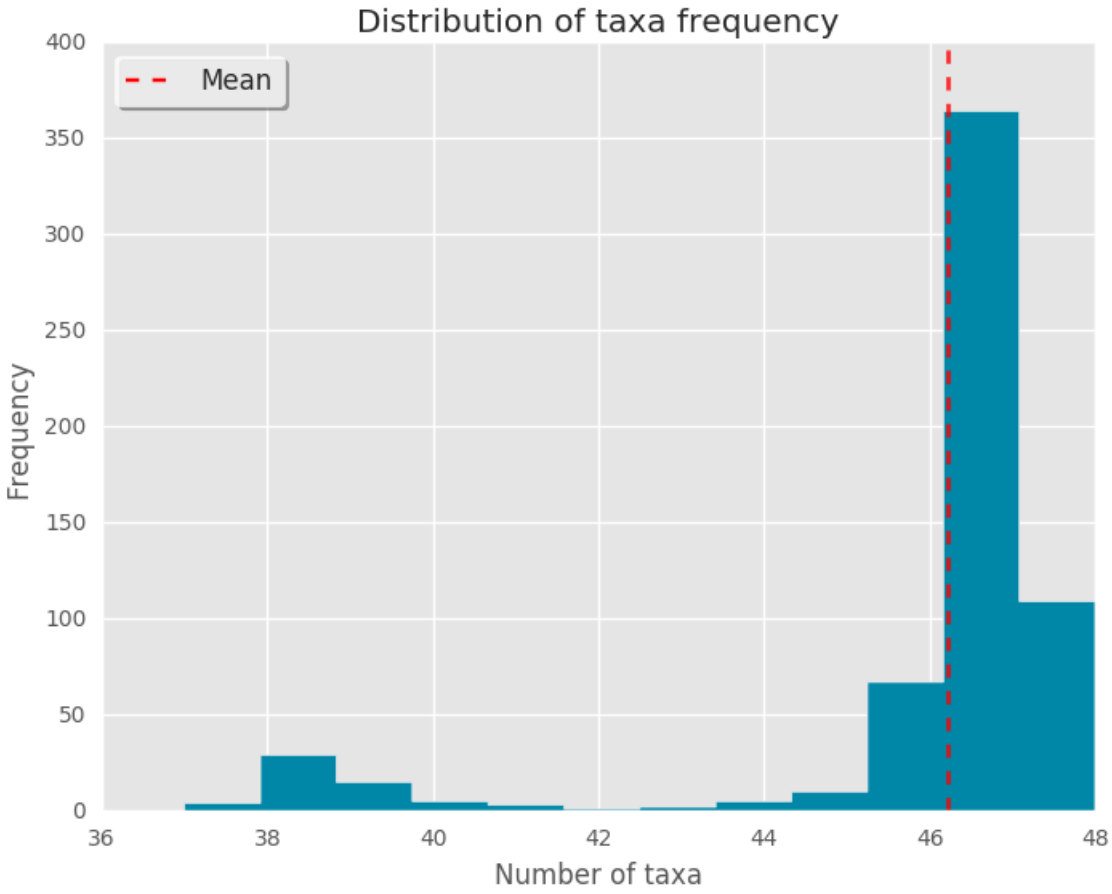


Figure 7.11: Example of the distribution of taxa frequency across the entire data set.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[General Information]
# Options available: average
distribution_taxa_frequency: average
```

Polymorphism and variation

Statistics analysis: ***Pairwise sequence similarity - Single gene***

Sliding window line plot with the average sequence similarity across the length of the selected alignment (Fig 7.12).

y-axis: Sequence similarity, in percentage.

x-axis: Alignment nucleotide/residue position.

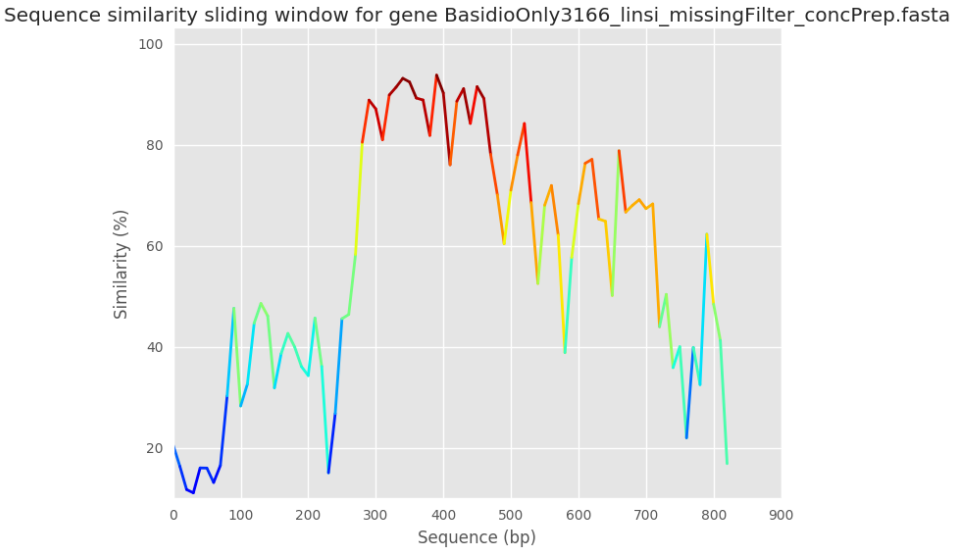


Figure 7.12: Example of the pairwise sequence similarity across a single gene.

Statistics analysis: ***Pairwise sequence similarity - Per species***

Creates a triangular heat map matrix with the sequence similarity values for each pairwise combination of taxa (Fig 7.13). Sequence similarity values are color coded according to the color bar at the right of the plot.

y-axis: Taxa names.

x-axis: Taxa names.

Statistics analysis: ***Pairwise sequence similarity - Average***

> **Note:** This analysis is computationally intensive and may take some minutes to finish. However, TriFusion uses an hash lookup table that considerably speeds the execution of subsequent analyses related with sequence similarity and segregating sites. See more details in appendix .2. ■

Plots an histogram with the distribution of sequence similarity values across the active data set

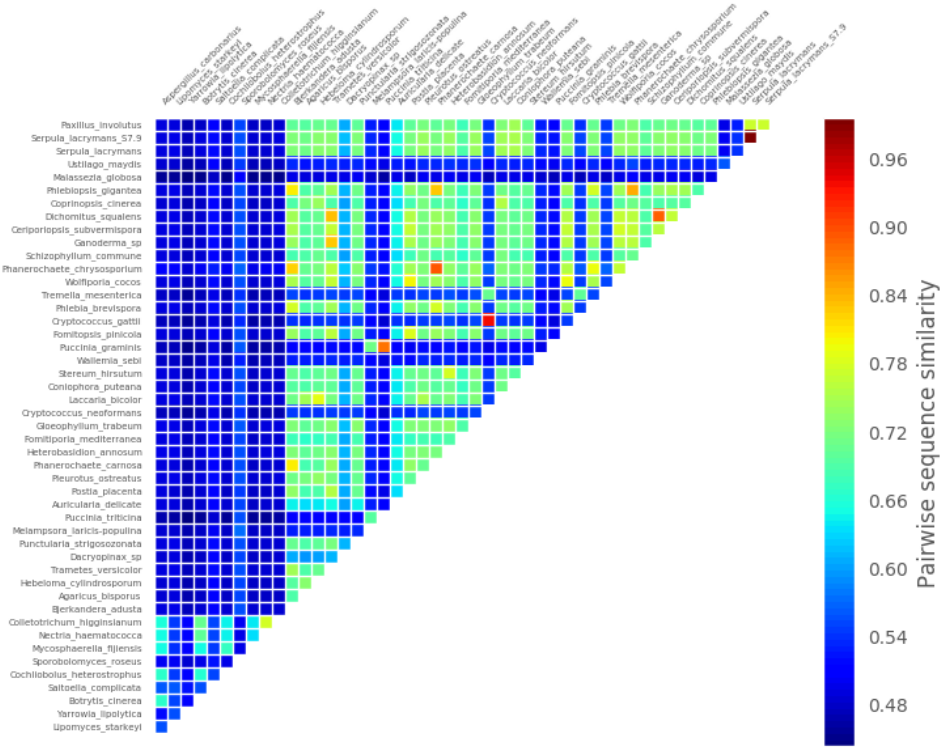


Figure 7.13: Example of the pairwise sequence similarity per taxa.

(Fig 7.14). Each entry in the histogram corresponds to the average sequence similarity of one alignment.

y-axis: Absolute frequency of alignments.

x-axis: Sequence similarity, in percentage.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Polymorphism and Variation]
# Options available: gene species average
sequence_similarity: species average
```

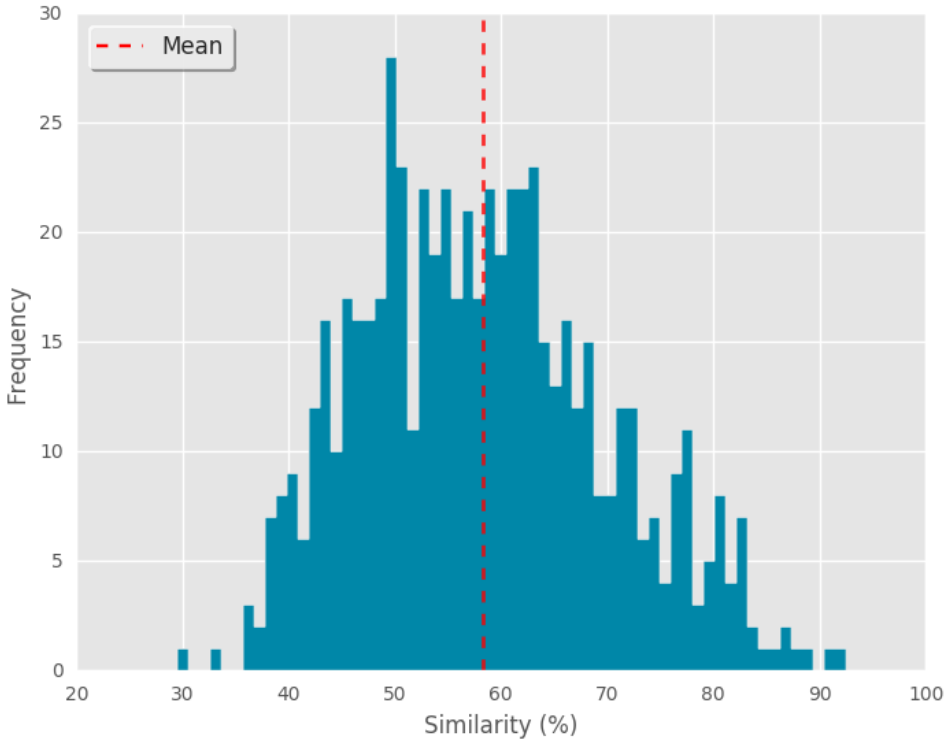


Figure 7.14: Example of the pairwise sequence similarity average across the data set.

Statistics analysis: ***Segregating sites - Single gene***

Sliding window line plot with the number of segregating sites across the length of the selected alignment (Fig 7.15).

y-axis: Number of segregating sites.

x-axis: Alignment nucleotide/residue position.

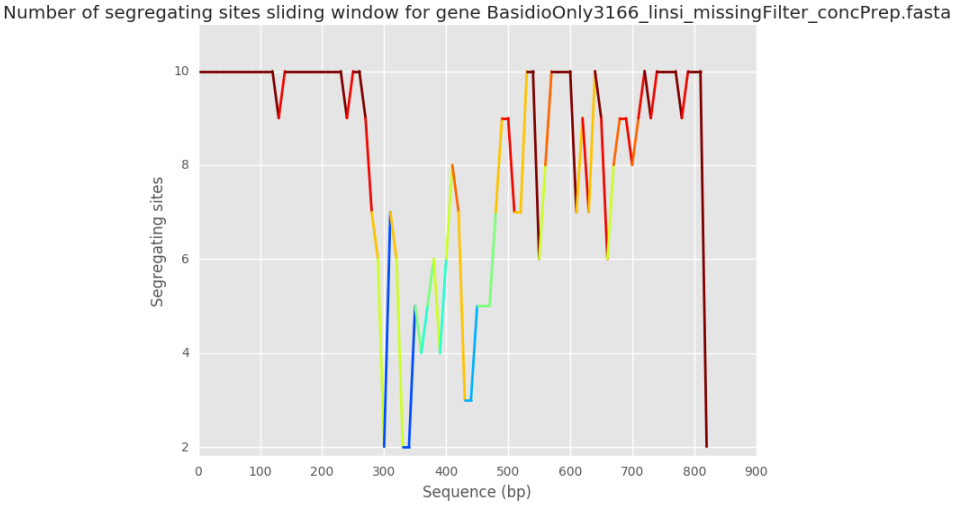


Figure 7.15: Example of the number of segregating sites across a single alignment.

Statistics analysis: ***Segregating sites - Per species***

> **Note:** This analysis is computationally intensive and may take some minutes to finish. However, TriFusion uses an hash lookup table that considerably speeds the execution of subsequent analyses related with sequence similarity and segregating sites. See more details in appendix .2. ■

Triangular heat map matrix with the number of segregating sites for each pairwise combination of taxa (Fig 7.16). The number of segregating sites are color coded according to the color bar at the right of the plot.

y-axis: Taxa names.

x-axis: Taxa names.

Statistics analysis: ***Segregating sites - Average***

> **Note:** This analysis is computationally intensive and may take some minutes to finish. However, TriFusion uses an hash lookup table that considerably speeds the execution of

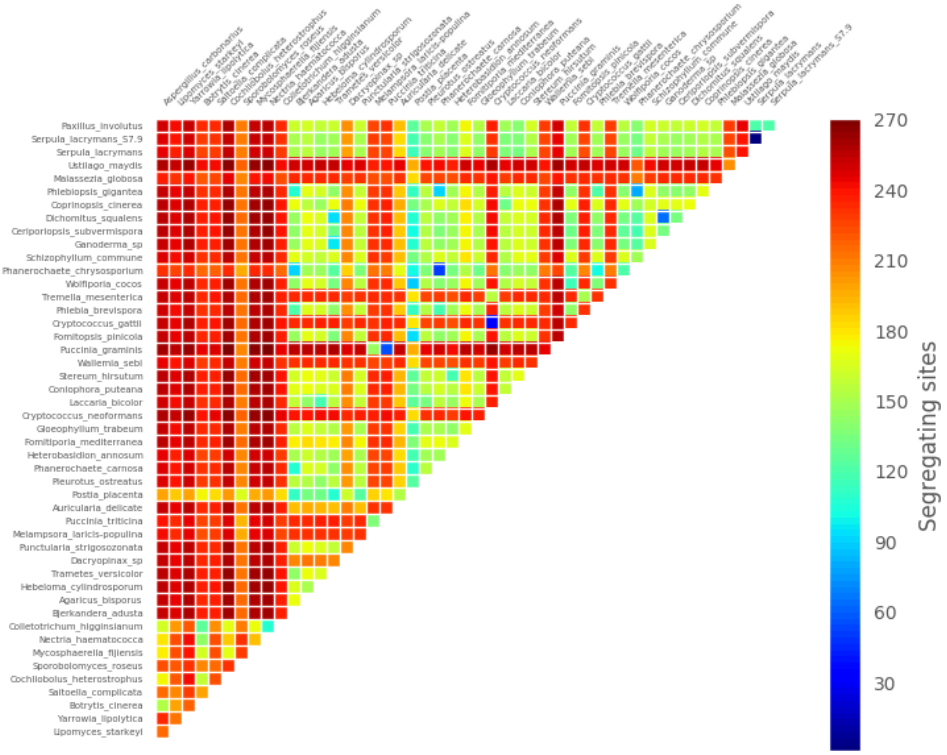


Figure 7.16: Example of the pairwise sequence similarity average across the data set.

subsequent analyses related with sequence similarity and segregating sites. See more details in appendix .2. ■

Plots an histogram with the distribution of the number of segregating sites, in absolute value or percentage, across the active data set (Fig 7.17). Each entry in the histogram corresponds to the averaged number of segregating sites in one alignment.

y-axis: Absolute frequency of alignments.

x-axis: Number of segregating sites, in absolute or percentage.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Polymorphism and Variation]
# Options available: gene species average
segregating_sites: species average
```

■

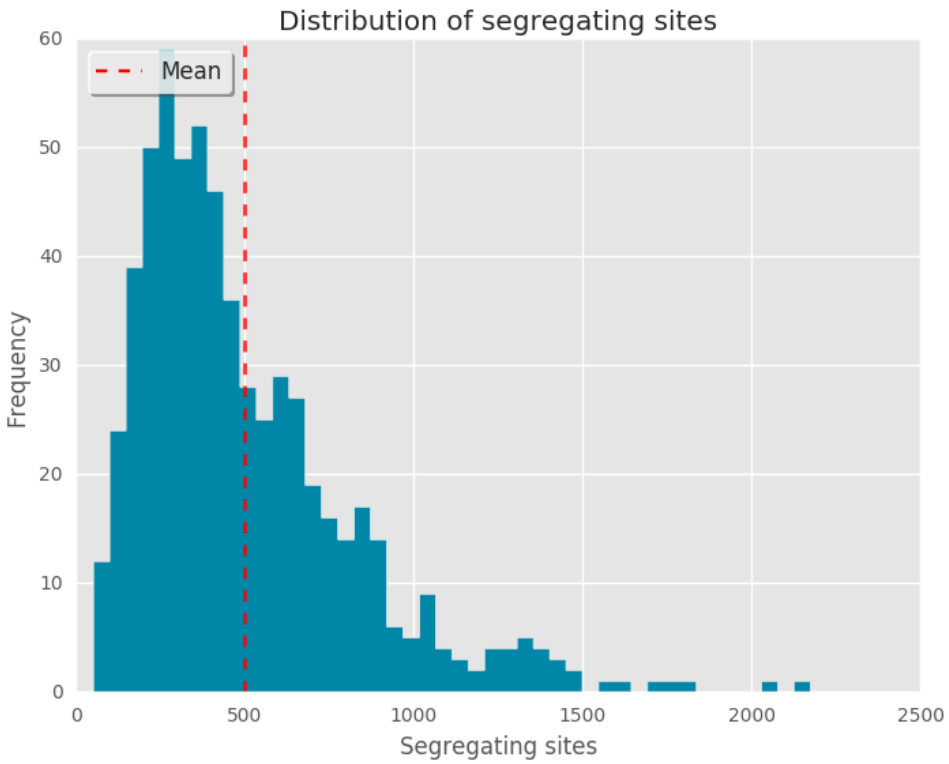


Figure 7.17: Example of the pairwise sequence similarity average across the data set.

Statistics analysis: ***Alignment length/polymorphism correlation - Average***

Calculates the non-parametric spearman's rank correlation coefficient between the alignment length and the number of variable sites for each alignment (Fig 7.18). A scatter plot depicting the relationship between these two variables is generated along with the best fit line. The correlation coefficient (ρ) and *p-value* are provided in the top left area of the plot.

y-axis: Number of informative sites.

x-axis: Alignment length.

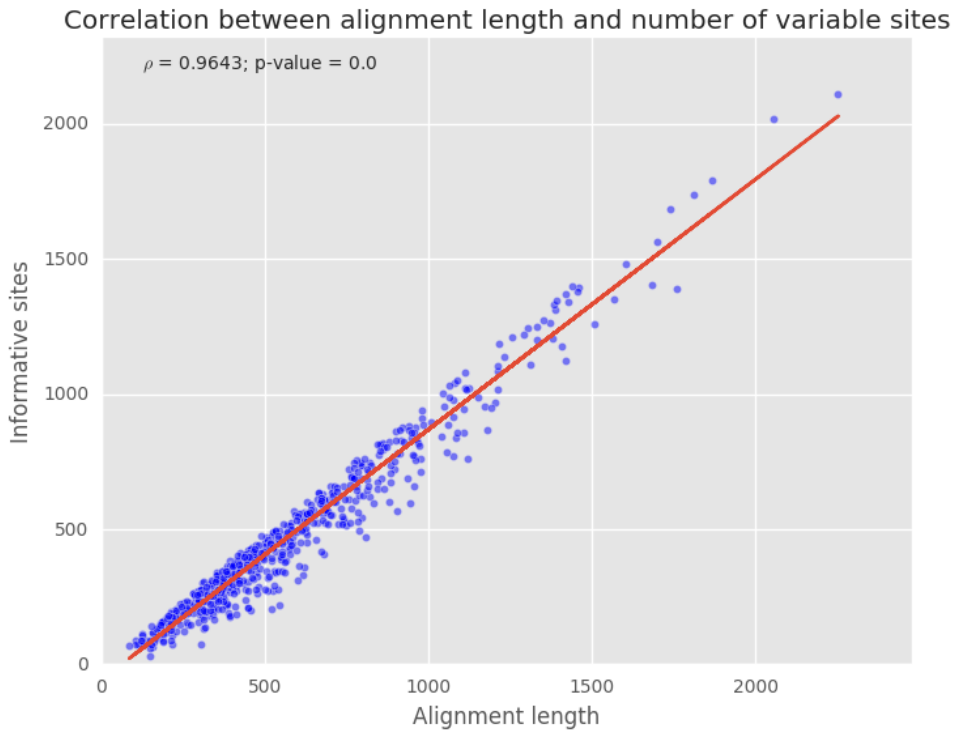


Figure 7.18: Example of a scatter plot depicting the correlation between variable sites and alignment length.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Polymorphism and Variation]
# Options available: average
alignment_pol_correlation: average
```

Statistics analysis: *Allele frequency spectrum - Single gene*

> **Note:** This analysis is only available for nucleotide sequences. ■

Plots the distribution of the allele frequency spectrum for a single gene.

y-axis: Frequency of occurrence.

x-axis: Derived allele frequency.

Statistics analysis: *Allele frequency spectrum - Average*

> **Note:** This analysis is only available for nucleotide sequences. ■

Creates the distribution of allele frequencies across the active data set (Fig 7.19). For each alignment, the distribution of the derived allele frequency is calculated and compiled into a single data matrix that depicts the distribution of allele frequencies for the entire data set.

y-axis: Frequency of occurrence.

x-axis: Derived allele frequency.

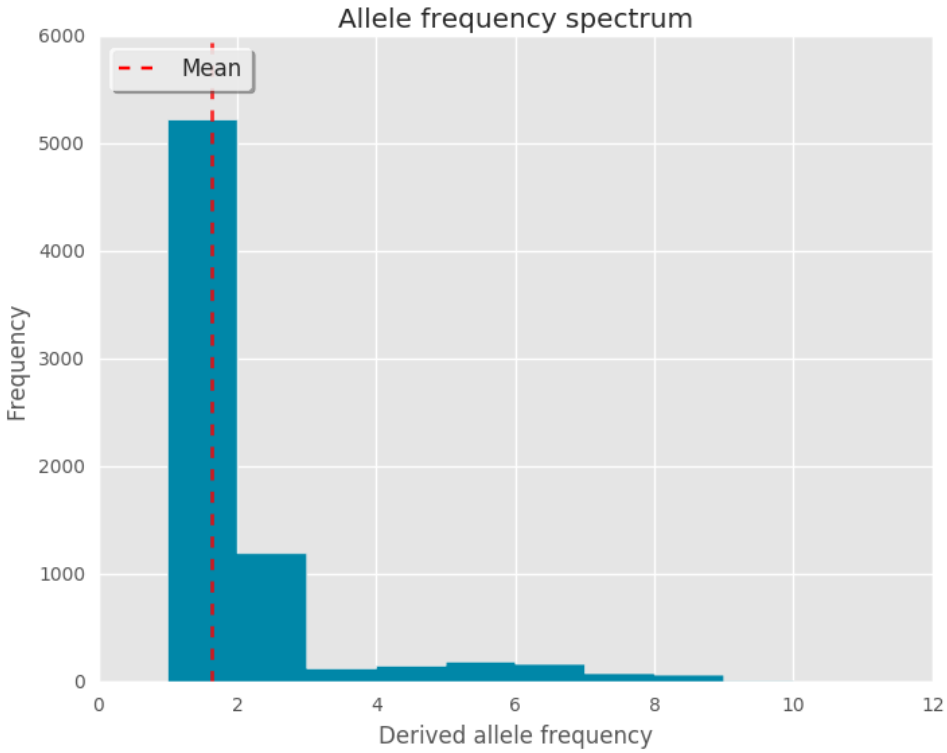


Figure 7.19: Example of an Allele Frequency Spectrum (AFS) plot averaged across the data set.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Polymorphism and Variation]
# Options available: gene average
allele_frequency_spectrum: average
```

■

Missing data

Statistics analysis: *Gene occupancy - Average*

Plots an interpolation matrix that scores the presence/absence of each taxon for every alignment in active data set (Fig 7.21). Each taxa is represented as a line in the *y-axis* and each alignment as a column in the *x-axis*. Whenever a taxon is absent in a given alignment, the coordinates for this taxon-alignment combination receive a score of 0 and a blank space is displayed. Conversely, if the taxa is present in a given alignment, the corresponding coordinate will receive a score of 1 and a black space is displayed. Sparse data sets with taxa missing in several alignments will tend to have more white spaces. Conversely, dense data sets with very few missing taxa will have fewer white spaces.

y-axis: Taxa index, each line represents one taxon.

x-axis: Alignment index, each column represents an alignment.

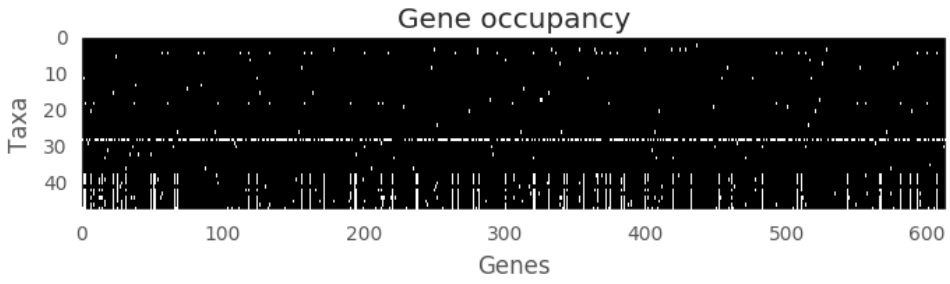


Figure 7.20: Example of a gene occupancy plot.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Missing Data]
# Options available: average
gene_occupancy: average
```

Statistics analysis: ***Distribution of missing taxa - Per species***

Plots the distribution of the number of genes missing for each taxon.

y-axis: Frequency, number of genes missing for a given taxon.

x-axis: Taxa names.

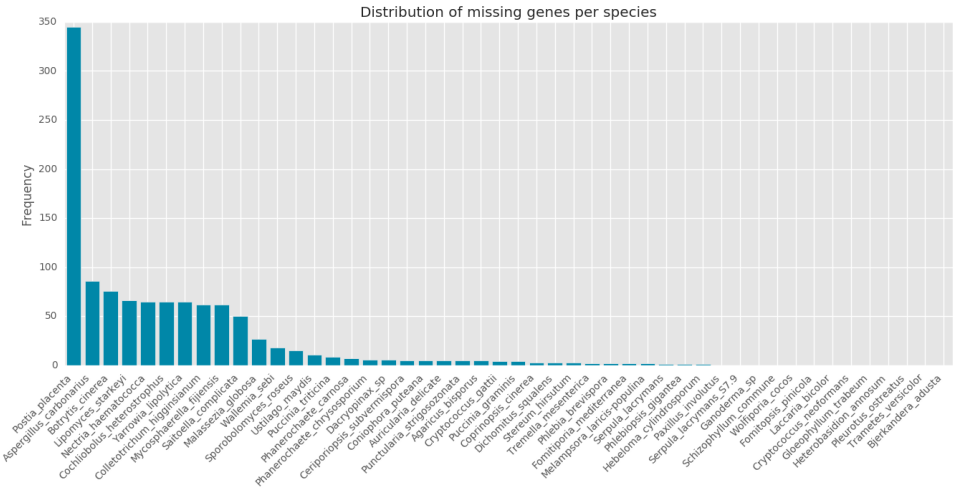


Figure 7.21: Example of the distribution of missing genes per taxon.

Statistics analysis: ***Distribution of missing taxa - Average***

Plots the distribution of the number of missing taxa for each alignment averaged across the entire active data set (Fig 7.22).

y-axis: Frequency, number of occurrences

x-axis: Taxa names.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Missing Data]
# Options available: species average
distribution_missing_genes: species average
```

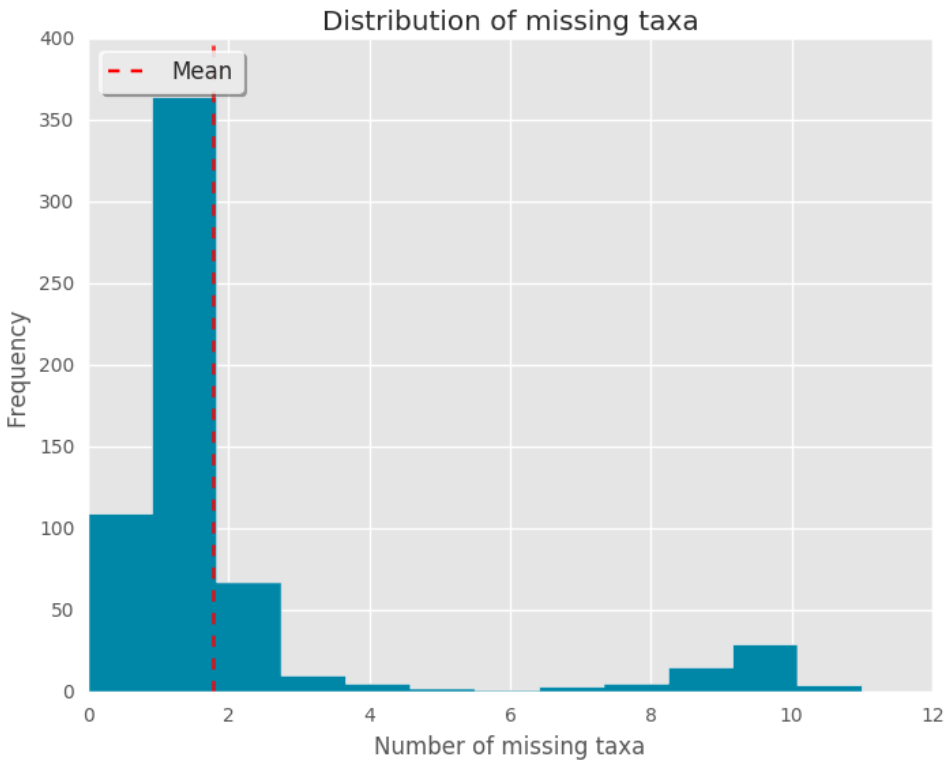


Figure 7.22: Example of the distribution of missing taxa across the entire data set.

Statistics analysis: *Distribution of missing data - Per species*

Stacked bar plot with the proportions of gaps, missing data and effective data for each taxon, averaged across the active alignments (Fig 7.23). In cases where a given taxon is absent from an alignment, this will score a missing data proportion of 1 for that alignment.

y-axis: Proportion.

x-axis: Taxa names.

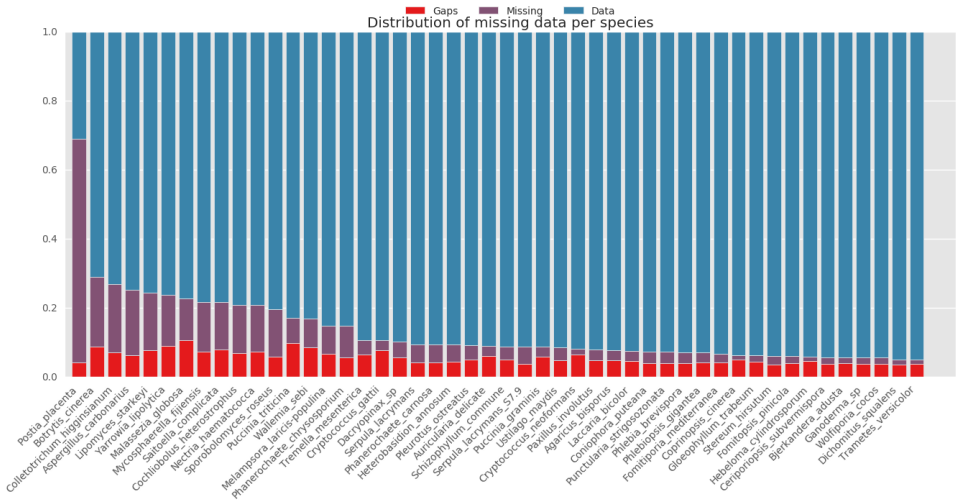


Figure 7.23: Example of the distribution of missing data for each species.

Statistics analysis: *Distribution of missing data - Average*

Smooth line histogram for three data categories: *Gaps*, *Missing data* and *Data* (Fig 7.24). For each alignment, the proportion of each of these categories is calculated and then compiled intro three data matrices. Each plot will then contain the distribution of the corresponding category across the entire active data set. As in the *Per species* plot, taxa that are missing from a given alignment will score a missing data proportion of 1 for that alignment. Data sets with little missing data will tend to have *"Data"* distributions abutting 100% values and *"Gaps"* and *"Missing data"* distribution abutting 0%. Increasing the amount of missing data will erase this pattern and tend to homogenize all three distribution or even to invert the trend.

y-axis: Frequency, number of alignments.

x-axis: Proportion, for each of the three categories.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

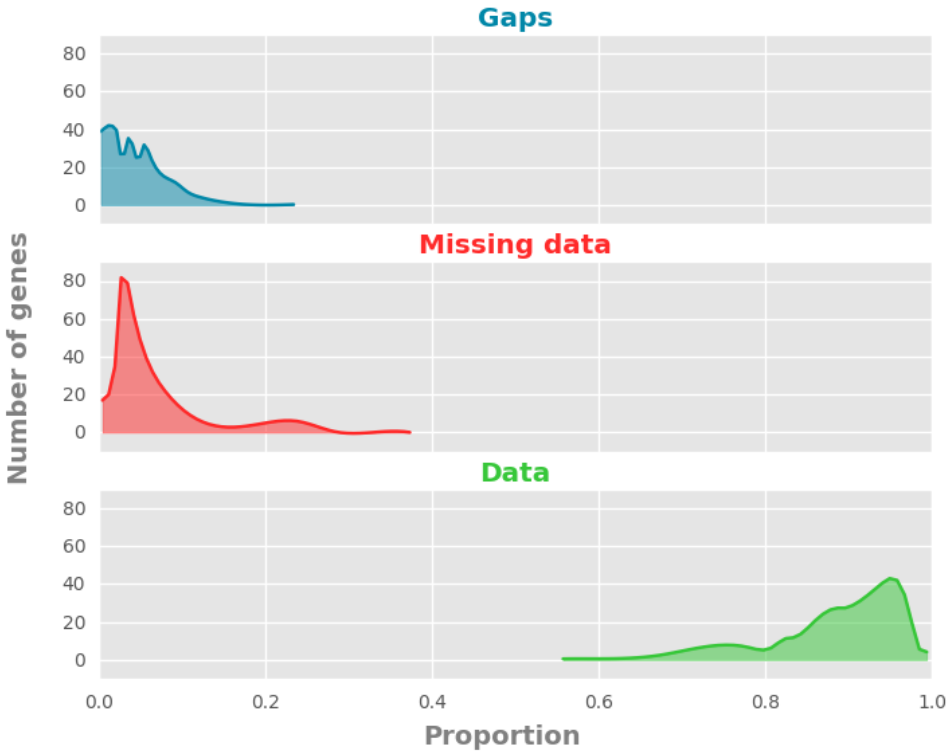


Figure 7.24: Example of the distribution of missing taxa across the entire data set.

```
[Missing Data]
# Options available: species average
distribution_missing_data: species average
```

Statistics analysis: ***Cumulative distribution of missing taxa - Average***

Plots the number of alignments available when applying consecutive minimum taxa representation filters (Fig 7.25). This range spans to 0% (All alignments) up to 100% (Alignment with no missing taxa) with steps of 5%. This plot can be useful when trying to decide the best value for the minimum taxa representation filter that minimizes missing data and maximizes the number of taxa/alignments.

y-axis: Frequency, number of alignments.

x-axis: Minimum taxa representation value, in percentage.

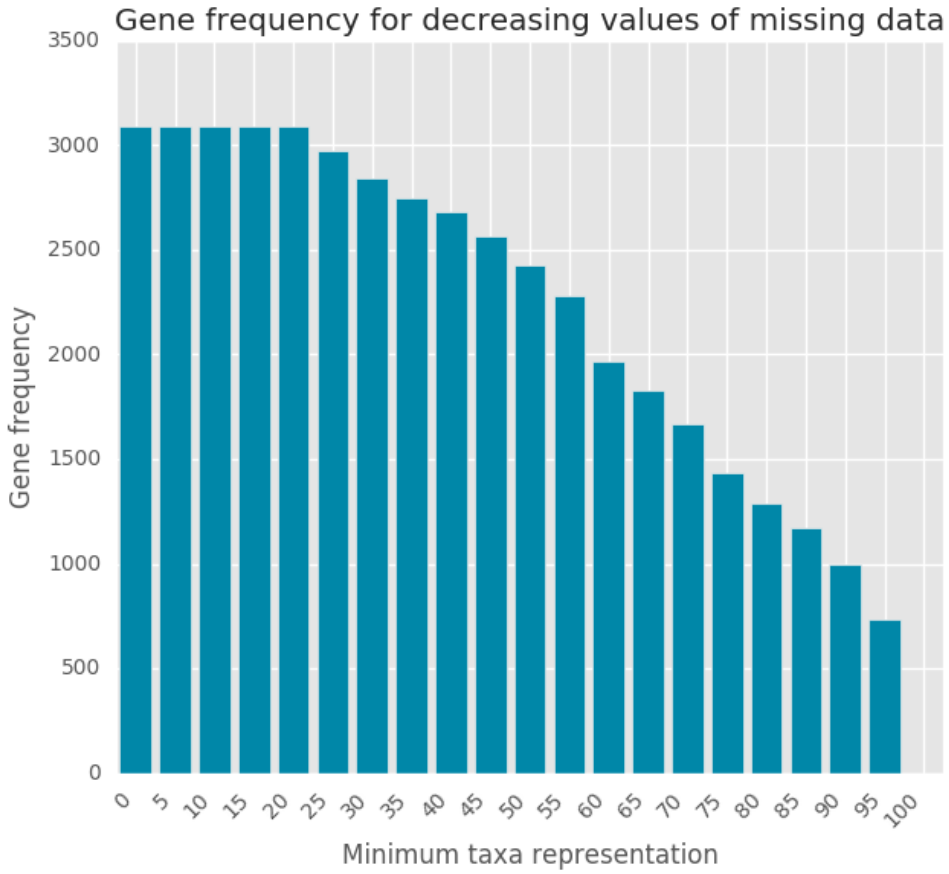


Figure 7.25: Example of the cumulative distribution of missing taxa across successive value of minimum taxa representation.

> **Command line version:**
>
> These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Missing Data]
# Options available: average
cumulative_distribution_missing_genes: average
```

Outlier detection

All outlier detection method implemented in TriFusion use a slightly modified Median Absolute Deviatin (MAD) method to find outliers. As the name implies, this method relies on the calculation of the median for the data set and then evaluate the distance of each data point from this value. The recommended threshold for this distance (above which a data point is considered an outlier) is 3.5, which is roughly equivalent to 3.5 standard deviations. The slight modification of the method is the inclusion of a consistency constant, which ensures that MAD provides a good estimate of the standard deviation independently of the sample size.

In each outlier detection plot, the number of outliers found will be displayed in the footer of the **Statistics** screen. If outliers are found, three functionalities will be available:

- **Remove:** Removes the outlier files/taxa from the current TriFusion session;
- **Export:** Exports the outlier files/taxa into a *.csv* file;
- **View:** Displays a dialog listing the outlier files/taxa.

Statistics analysis: *Missing data outliers - Per species*

Smooth line distribution of the amount of missing data for each taxon across the entire active data set (Fig 7.26). Data points will be based on the proportion of missing data symbols out of the possible total. For example, in an alignment with three taxa, each with 100 sites, the total possible missing data is 300 (100×3). Taxa that are absent from a given alignment are ignored, in order to avoid biasing outlier detection of missing **taxa** instead of missing **data**. Data points represent indivudal taxa and are depicted in a rug-like style along the *x-axis*. Outliers are marked with red dots.

y-axis: Frequency, number of occurrences.

x-axis: Proportion of missing data.

Statistics analysis: *Missing data outliers - Average*

Smooth line distribution of the average missing data for each alignment (Fig 7.27). Data points represent individual alignments and are depicted in a rug-like style along the *x-axis*. Outliers are marked with red dots.

y-axis: Frequency, number of occurrences.

x-axis: Proportion of missing data.

> **Command line version:**
>
> These analyses can be run in **TriStats** by adding the following lines to the configuration

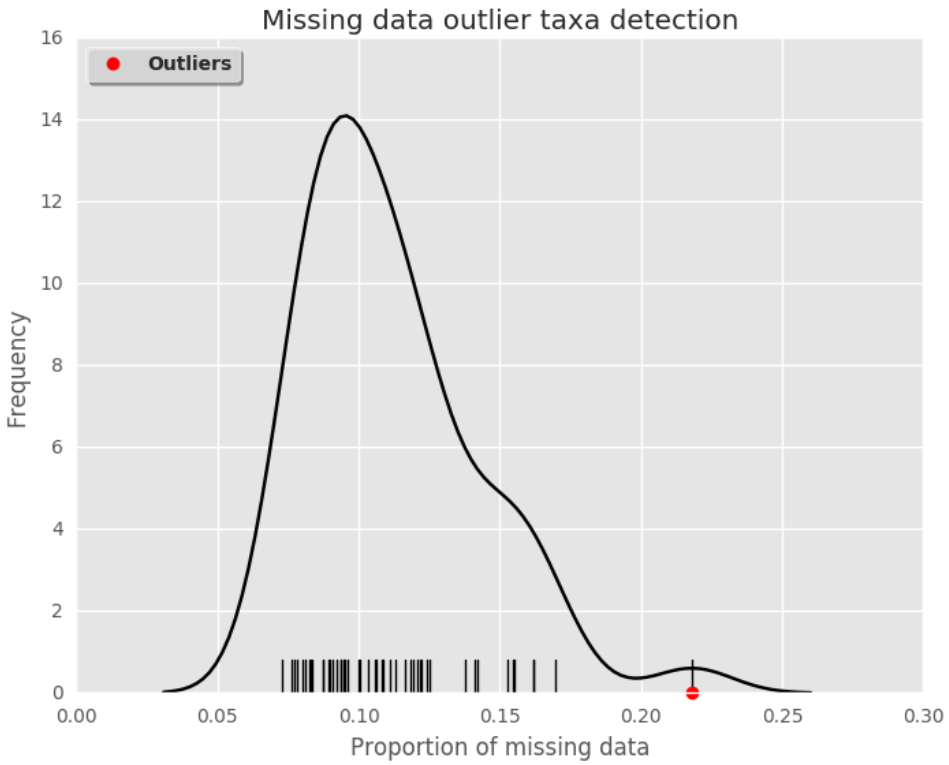


Figure 7.26: Outlier detection plot of missing data for each taxon.

file (see reference section 10.1.2):

```
[Outlier Detection]
# Options available: species average
missing_data_outliers: species average
```

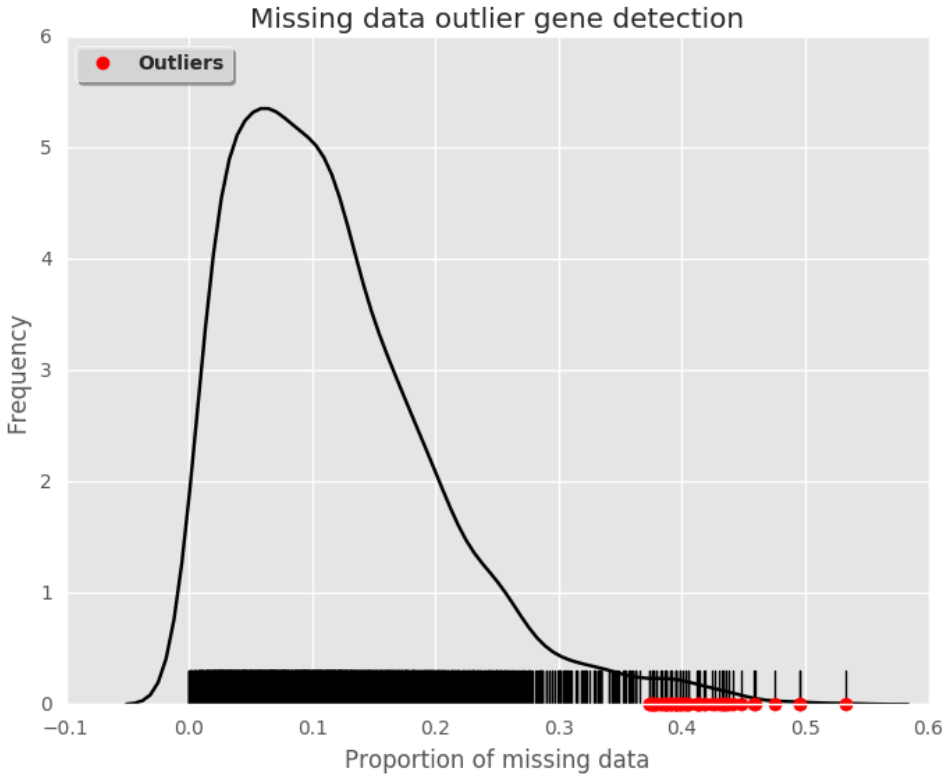


Figure 7.27: Outlier detection plot of missing data for each alignment.

Statistics analysis: ***Segregating sites outliers - Per species***

> **Note:** This analysis is computationally intensive and may take some minutes to finish. However, TriFusion uses an hash lookup table that considerably speeds the execution of subsequent analyses related with sequence similarity and segregating sites. See more details in appendix .2. ■

Smooth line distribution of the average pairwise proportion of segregating sites for a single taxon when compared with every other taxa (Fig 7.28). Taxa outliers will be identified when a given taxon has an average excess or lack of segregating sites with every other taxa. Alignment columns containing gaps or missing data are ignored. Data points represent individual taxa and are depicted in a rug-like style along the *x-axis*. Outliers are marked as red dots.

y-axis: Frequency, number of occurrences.

x-axis: Proportion of segregating sites.

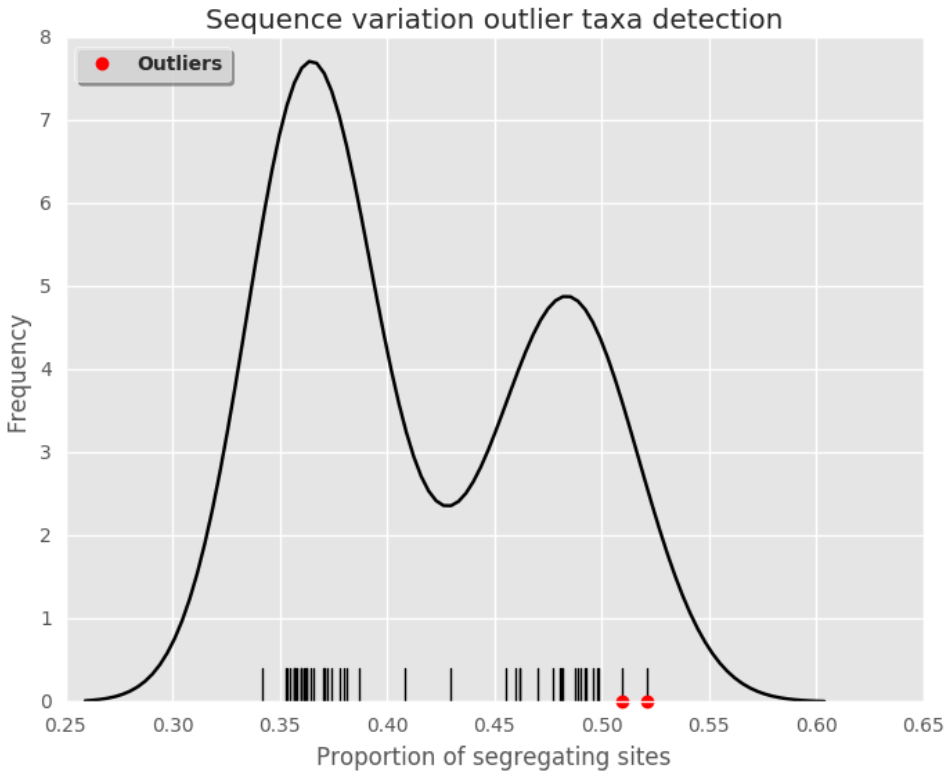


Figure 7.28: Outlier detection plot of segregating sites for each taxon.

Statistics analysis: ***Segregating sites outliers - Average***

Note: This analysis is computationally intensive and may take some minutes to finish. However, TriFusion uses an hash lookup table that considerably speeds the execution of subsequent analyses related with sequence similarity and segregating sites. See more details in appendix .2. ■

Smooth line histogram of the proportion of segregating sites for each alignment (Fig 7.29). Alignment outliers will be identified when they have an excess or lack of segregating sites, compared to the remaining alignments. Alignment columns containing gaps or missing data are ignored. Data points represent individual alignments and are depicted in a rug-like style along the *x-axis*. Outliers are marked with red dots.

y-axis: Frequency, number of occurrences.

x-axis: Proportion of segregating sites.

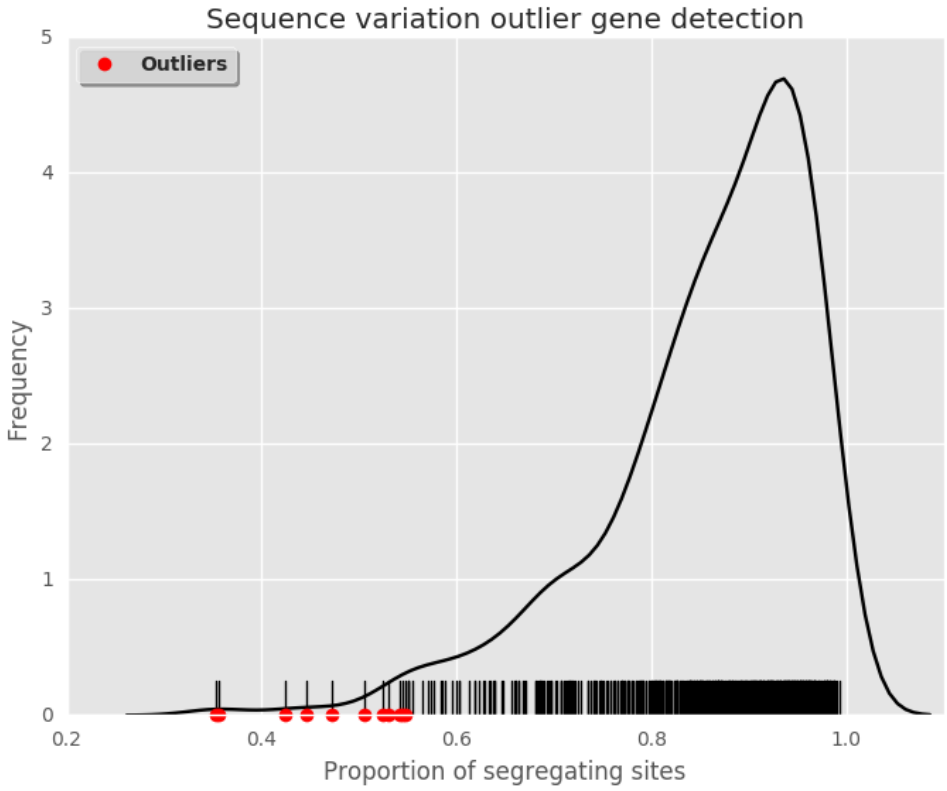


Figure 7.29: Outlier detection plot of segregating sites for each alignment.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Outlier Detection]
# Options available: species average
segregating_sites_outliers: species average
```

Statistics analysis: *Sequence size outliers - Per species*

Smooth line distribution of the average sequence size for each taxon. The sequence size measure excludes gaps and missing data (Fig 7.30). Taxa outliers will be identified when a given taxon consistently has sequences much longer or shorter than the median of the active data set. Data points represent individual taxa and are depicted in a rug-like style along the *x-axis*. Outliers are marked with red dots.

y-axis: Frequency, number of occurrences.

x-axis: Sequence size in nucleotides/residues.

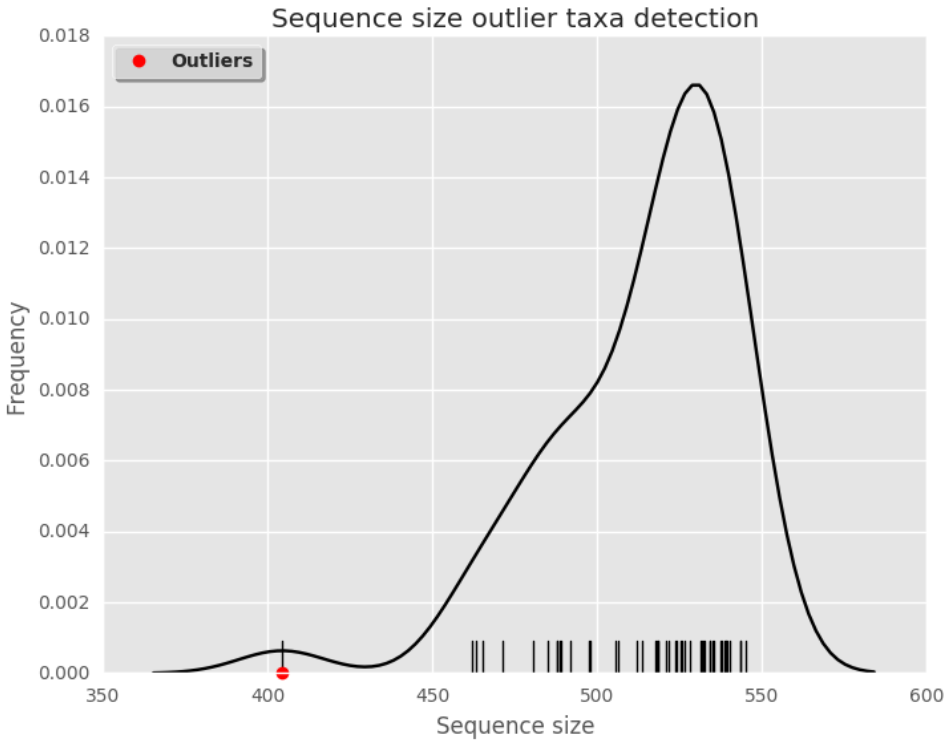


Figure 7.30: Outlier detection plot of sequence size for each taxon.

Statistics analysis: *Sequence size outliers - Average*

Smooth line distribution of the average sequence size for each alignment (Fig 7.31). The sequence size measure excludes gaps and missing data. Outlier alignments will be identified when the average sequence size of an alignment is much longer or shorted than the median of the active data set. Data points represent individual taxa and are depicted in a rug-like style along the *x-axis*. Outliers are marked with red dots.

y-axis: Frequency, number of occurrences.

x-axis: Sequence size in nucleotides/residues.

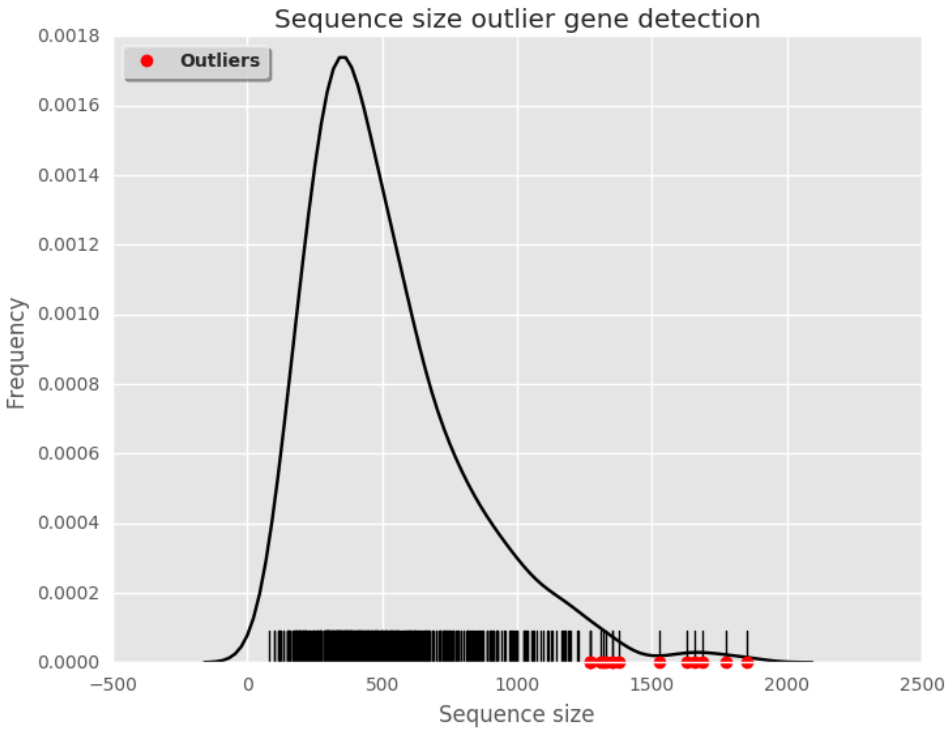


Figure 7.31: Outlier detection plot of sequence size for each alignment.

Command line version:

These analyses can be run in **TriStats** by adding the following lines to the configuration file (see reference section 10.1.2):

```
[Outlier Detection]
# Options available: species average
sequence_size_outliers: species average
```

II Command Line (CLi) options reference

8. Orthology - Cli (orthomcl_pipeline)

8.1 Options

8.1.1 General options

-in *Input_directory*

Provide the path to the directory containing the input proteome files. There should be one protome file in Fasta format for each taxon. See more details on the input format specifications in section 5.2.1.

8.1.2 Execution modes

-n

Perform the complete run of the *OrthoMCL* pipeline.

-a

Only perform the ajust fasta files, where the input proteomes are adjusted to be compliant with the rest of the pipeline.

-na

Perform the complete run of the *OrthoMCL* pipeline without adjusting the input proteome files.

8.1.3 Input formatting

-d

Do not convert input proteome files names because they are already in code (e.g. *Homo_sapiens.fas* → *Hosap.fas*).

-sep *Separator*

Specify the separator in the input files (e.g. '_' is the separator in 'Homo_sapiens.fas'). This parameter is ignored if the *-d* option is set.

8.1.4 Ortholog search options

--min-length *Min_length*

Default: 10

Set minimum length allowed for protein sequences.

--max-stop *Max_stop*

Default: 20

Set maximum percentage of stop codons in protein sequences.

--db *Database*

Default: goodProteins

Name of the search database that will be created from the input proteome files.

--search-out *Ouput_database*

Default: AllVsAll.out

Name of the *USearch* output of the All-vs-All protein homology search.

-evalue *Value*

Default: 1e-05

Sets the e-value cut-off for the *Usearch* protein homology search. Value can be float (0.0005) or in scientific notation (1e-04).

-inflation *Value1 Valu2 ...*

Set inflation values for ortholog group clustering using *MCL*. Multiple values may be provided but they are limited to the range [1, 5]

8.1.5 Output options

-o *Output_directory*

Sets the directory where the *OrthoMCL* results will be generated.

-prefix *Prefix*

Default: Ortholog

Set the prefix name for each ortholog group.

-id *ID_num*

Default: 1

Set the starting number for the ortholog groups.

--groups-file *groups_file_prefix*

Default: group

Sets the prefix for all group files generated at the end of the pipeline. The final names will be a combination of this prefix and the corresponding inflation value (e.g. *group_3.txt*).

--min-species *Minimum_species*

Set the minimum number of species required for an ortholog cluster to be converted into protein sequence. This option will only affect the protein sequence files, not the group file output.

--max-gene-copy *Maximum_copies*

Set the maximum number of gene copies from the same taxon for each ortholog cluster. This option will only affect the protein sequence files, not the group file output.

8.1.6 Miscellaneous options

-np *Cpus*

Default: 1

Nuber of CPUs to be used during the *Usearch* homology search operation.

8.2 Usage examples

8.2.1 Complete pipeline example run

Here is an example of a complete *OrthoMCL* run. 20 proteome files are stored in the directory */home/user/orto_dir* and I would like to search for single copy orthologs that contain at least 50% of the taxa. Since I am uncertain of which inflation value I should chose, a range of values will

be provided. I will also make use of my multi-core server and request 10 cpu's for the *USearch* operation.

```
1 orthomcl_pipeline -in /home/user/orto_dir -n -inflation 1.5 3 5
  ↪ --min-species 0.5 --max-gene-copy 1 -np 10 -o
  ↪ /home/user/orto_output
```

9. Process - Cli (TriSeq)

9.1 Options

9.1.1 General options

-in *Input_file1 Input_file2 ...*

Provide the input file name. If multiple files are provided, please separated the names with spaces.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
```

-if *Input_format*

Available options: *fasta*, *nexus*, *phylip*, *guess*

Format of the input file(s). This options does not need to be specified, since TriSeq will automatically detect the format and sequence type (nucleotide or protein) when the default *guess* option is specified.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -if fasta -of fasta -o
    ↪ output
```

-of *Output format*

Available options: *nexus*, *phylip*, *fasta*, *mcmctree*, *ima2*, *stockholm*, *gphocs*

Format of the output file(s). You may select multiple output formats simultaneously (default is 'nexus').

```
# Example; specifying three output formats
TriSeq -in file1.fas file2.fas file3.fas -if fasta -of fasta nexus
↪ phylip -o output
```

-o *Output file*

Name of the output file. No extension is required.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
```

9.1.2 Main operations

-r *Partition file*

Reverses a single concatenated file into its original single locus alignments. A partition file must be provided with this option in one the formats specified in section 4.3.2.

```
# Example
TriSeq -in concatenated_file.fas -of fasta -r partition_file
```

-c

Flag used to specify the conversion (instead of concatenation) of the input files passed as arguments with the *-in* option. This flag precludes the usage of the -o option, as the output file name is automatically generated based on the name of the corresponding input file.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -c
```

9.1.3 Secondary operations

--collapse

Use this flag if you would like to collapse the input alignment(s) into unique haplotypes. See section 6.5.5 for more information.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
 ↪ --collapse
```

--consensus *Variation handling method*

Available options: *First sequence*, *IUPAC*, *Soft mask*, *Remove*.

Creates a consensus of the final alignments specifying how variation is handled. See section 6.5.6 for more information.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
 ↪ --consensus IUPAC
```

--consensus-single-file

Merges the consensus sequences of multiple alignments in a single file.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
 ↪ --consensus "Soft mask" --consensus-single-file
```

--missing-filter *gap_threshold missing_threshold*

Use this option if you wish to filter the missing data from the alignments. Two thresholds must be provided with this option:

- **gap_threshold:** Maximum percentage allowed for gaps
- **missing_threshold:** Maximum percentage allowed for missing data (gaps + actual missing data)

```
# Example; Filter columns with more than 50% of gaps and 75%
  ↪ missing data
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
  ↪ --missing-filter 50 75
```

--min-taxa ***Minimum taxa percentage***

Set the minimum percentage of taxa that needs to be present in an alignment. Alignments with less taxa are ignored.

```
# Example. Output only alignments with more than 50% of the total
  ↪ number of taxa
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
  ↪ --missing-taxa 50
```

--contain-taxa ***Taxon1 Taxon2 ...***

Only processes alignments that **contain** the specified taxa.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
  ↪ --contain-taxa Taxon1 Taxon3 Taxon10
```

--exclude-taxa ***Taxon1 Taxon2 ...***

Only process alignments that **DO NOT** contain the specified taxa.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
  ↪ --exclude-taxa Taxon1 Taxon3
```

--codon-filter ***codon_positions ...***

Include only the codon positions specified by this option (DNA alignments only).

```
# Example; Include only 1st and 2nd codon positions
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
    ↪ --codon-filter "1 2"
```

--variable-filter *Minimum Maximum*

Provide minimum and maximum values of variable sites for each alignment. Filters alignments with a number of variable sites outside the specified range.

```
# Example; Include only alignments with less than 11 variable
    ↪  sites
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
    ↪ --variable-filter 0 10
# Example; Include only alignments with more than 2 variable sites
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output
    ↪ --variable-filter 2 None
```

--informative-filter *Minimum Maximum*

Provide minimum and maximum values of informative sites for each alignment. Filters alignments with a number of informative sites outside the specified range. See the option above for the example.

--gcoder

Use this flag to code the gaps of the alignment into a binary state matrix that is appended to the end of the alignment. Only works for Nexus output format.

9.1.4 Utilities

-p / --partition-file *Partition file*

Using this option and providing the partition file will convert it between a RAxML or Nexus format.

```
# Example
TriSeq -p partition_file
```

-s *Taxa1 Taxa2 ...*

Selects alignments containing the provided taxa (separate multiple taxa with whitespace).

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output.fas -s
↪ Taxon1 Taxon2 Taxon3
```

-rm *Taxon1 Taxon2 ...*

Removes the specified taxa from the final alignment. Unwanted taxa my be provided in a text file containing 1 column with a species name in each line or they may be specified in the command line and separated by whitespace.

```
# Example; Provide taxa names
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output.fas
↪ -rm Taxon3 Taxon5 Taxon19
# Example; Provide file
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output.fas
↪ -rm taxa_to_remove.txt
```

-grep *Taxon1 Taxon2 ...*

The inverse of the *-rm* command. It removes all taxa from the alignment except for the ones specified with this option. Taxa names may be specified in a text file containing 1 column with a species name in each line or in the command line separated by whitespace. See the option above for the example.

-outgroup *Taxon1 Taxon2 ...*

Provide taxon names/number for the outgroup (This option is only supported for NEXUS output format files).

```
# Example; Provide taxa names
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output.fas
↪ -rm Taxon3 Taxon5 Taxon19
# Example; Provide file
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output.fas
↪ -rm taxa_to_remove.txt
```

9.1.5 Formatting options

-z *Zorro_suffix*

Use this option if you wish to concatenate auxiliary Zorro files associated with each alignment. Provide the suffix for the concatenated zorro file.s

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output.fas -z
↪ _zorro
```

-model *Model*

Available options: *DAYHOFF*, *DCMUT*, *JTT*, *MTREV*, *WAG*, *RTREV*, *CPREV*, *VT*, *BLOSUM62*, *MTMAM*, *LG*

This option only applies for the concatenation of protein data into phylip format. Specify the model for all partitions defined in the partition file (default is 'LG').

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of fasta -o output.fas -m
↪ WAG
```

-interleave

By default, output files are written in leave format. Specify this option to write output files in interleave format

--ima2-params *Pop_file Pop_tree Mutation_model Inheritance_scalar*

Provide 4 additional arguments needed to write the output in a format compliant with IMa2. The order of the required arguments (separated by whitespace is as follows: [(1) File name of population mapping][(2) Population tree][(3) Mutational model][(4) Inheritance Scalar]. Additional notes: (1) The population mapping file is a simple .csv file containing two columns separated by a semi-colon, in which the first column contains the taxon name and the second column contains the corresponding population name; (2) The order of the population names in the population tree must be the same as the order in the file with the population mapping.

```
# Example
TriSeq -in file1.fas file2.fas file3.fas -of ima2 -o output.fas
↪ --ima2-params population_file.txt (1,2):3,4:5 IS 1
```

9.1.6 Data manipulation

--get-taxa

Writes all taxa names into a text file.

10. Statistcs - Cli (TriStats)

10.1 Options

10.1.1 General options

-in *Input_file1 Input_file2 ...*

Provide the input file name. If multiple files are provided, please separated the names with spaces.

```
# Example
TriStats -in file1.fas file2.fas file3.fas -o my_project -cfg
↪ my_options.ini
```

-o *Project_name*

Choose a name for the output of the **Statistics** project. A directory with this name will be created in the current working directory and all generated plots will be stored there. See the option above for the example.

-cfg *Configuration_file*

Provide the configuration file where the **Statistics** analyses to be performed are specified. An example configuration file can be generated with the *--generate-cfg* option below, and more details on how to set your analyses can be seen in section 10.1.2.

--generate.cfg

Run with this option alone to generate a sample configuration file that contains all available analyses selected for **TriStats**.

```
TriStats --generate-cfg
```

10.1.2 Configuration file

The configuration file that must be passed to **TriStats** provides a convenient and flexible way to select the analyses you wish to perform. An example configuration file can be automatically generated using the *--generate-cfg* option (see 10.1.1) and you can modify this file as you see fit. Here are the rules.

> **Note:** All of the following fields are case-insensitive, but always check for typos that may cause options to be ignored. ■

The same categories of the GUI version of **TriStats** are enclosed in "[]".

```
[General Information]
```

For each category, the individual options follow below with a pattern of *key: value*, where *key* is the name of the analysis and *value* contains the desired plot types separated by spaces. Note that individual analyses have to be below their corresponding category, otherwise they will be ignored.

```
[General Information]
# Options available: species average
distribution_sequence_size: species average
# Options available: species average
proportion_nucleotides_residues: species average
# Options available: average
distribution_taxa_frequency: average
```

In the example above, the *distribution_sequence_size* analysis will be performed for plot types *species* and *average*. To prevent certain plot types to be generated, simply remove them. Entire analysis can also be removed or commented out (with a "#" at the beginning of the line).

The generated sample configuration file already includes the available plot type options in a command above each analysis. These are the same as described in section 7.4.1:

- **gene**: The *single gene* plot type that focus on a single specific gene. **Can only be used with single input files**.
- **species**: The *per species* plot type that focus on taxa
- **average**: The *average* plot type that focus on whole dataset averages

III Appendix

.2 Hash lookup table to speed sequence similarity calculations

The computation of pairwise sequence similarities (or number of segregating sites) can be very computationally intensive due to the sheer number of pairwise combinations that need to be calculated, even in small data sets. Several analyses of the **Statistics** module make use of this calculation, which means that the same calculations may be required multiple times during a single TriFusion session. To alleviate the computational strain generated by the replication of these calculations, we use an hash table lookup method to gain considerable speed ups.

Here's how it works. During the execution of analyses that require the calculation of pairwise sequence similarities, an hash table in a SQLite database is populated with an hash of the two sequence strings as keys and the corresponding sequence similarities as values. Before the calculation of similarity values between two sequences, TriFusion will always look if a value is already available for those two sequences in the hash table - it performs a table look up. If not, the similarity value is calculated and stored in the hash table. If the hash of the two sequence's combination is already present in the table the corresponding value is simply fetched, which is considerably less costly than performing the calculation again. **This means that the similarity value for a particular combination of sequences is calculated only once.** Additionally requests will simply fetch the corresponding value.

This means that subsequent analyses that make use of these calculations will be much faster. For instance, changing the active data set and repeating the ***Sequence similarity per species*** analyses will take only a fraction of the time it took to complete the first analysis. The same happens when analyzing segregating sites after calculating the pairwise sequence similarities, since the metrics of both analyses are related.

However, we note that the hash table stored in the SQLite database only exists during the current TriFusion session. Every time TriFusion is restarted, so will the hash table. This is done to avoid feeding a monstrous table with consecutive usages of TriFusion.

Bibliography

[Edg10] Robert C Edgar. “Search and clustering orders of magnitude faster than BLAST”. In: *Bioinformatics* 26.19 (2010), pages 2460–2461. DOI: 10.1093/bioinformatics/btq461 (cited on page 47).

[LSR03] Li Li, Christian J Stoeckert, and David S Roos. “OrthoMCL: identification of ortholog groups for eukaryotic genomes.” In: *Genome research* 13.9 (2003), pages 2178–2189. ISSN: 1088-9051. DOI: 10.1101/gr.1224503. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=403725{\&}tool=pmcentrez{\&}rendertype=abstract (cited on pages 35, 42, 46).

[LS03] Na Li and Matthew Stephens. “Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data”. In: *Genetics* 165.4 (2003), pages 2213–2233. ISSN: 00166731. DOI: 10.1534/genetics.104.030692 (cited on page 10).

[SO00] M P Simmons and H Ochoterena. “Gaps as characters in sequence-based phylogenetic analyses.” In: *Systematic biology* 49.2 (2000), pages 369–381. ISSN: 1063-5157. URL: http://www.ncbi.nlm.nih.gov/pubmed/12118412 (cited on pages 12, 78).

[WCE12] Martin Wu, Sourav Chatterji, and Jonathan a. Eisen. “Accounting For Alignment Uncertainty in Phylogenomics”. In: *PLoS ONE* 7.1 (2012). Edited by Marco Salemi, e30288.

ISSN: 1932-6203. DOI: 10.1371/journal.pone.0030288. URL: http://dx.plos.org/10.1371/journal.pone.0030288 (cited on page 66).