TOA (Taxonomy-oriented Annotation)

v0.65 (17/06/2021 update)

A software package for automated functional annotation in non-model plant species

GI Sistemas Naturales e Historia Forestal
(formerly known as GI Genética, Fisiología e Historia Forestal)
Dpto. Sistemas y Recursos Naturales
ETSI Montes, Forestal y del Medio Natural
Universidad Politécnica de Madrid

https://github.com/ggfhf/

Table of contents

Disclaimer

The software package TOA (Taxonomy-oriented Annotation) is available for free download from the GitHub software repository (https://github.com/GGFHF/TOA) under GNU General Public License v3.0.

Comentado [U1]: Ponemos un párrafo referente a dónde se puede descargar la versión para el cloud de Amazon?

Introduction

Functional annotation is an essential stage of bioinformatic analysis of next-generation sequencing experiments. The purpose of this analysis is to assess the biochemical and biological functions of the sequences yielded after an assembly, the stage where reads are aligned and merged in longer fragments: contigs or scaffolds. In order to perform the functional annotation, it is necessary to access functional information deposited in genomic databases. Functional annotation is usually a slow task because it involves manual querying to the databases and the subsequent processing of the information obtained.

TOA aims to establish workflows geared towards woody plant species that automate the extraction of information from genomic databases and the annotation of sequences. TOA uses the following databases: Dicots PLAZA 4.0, Monocots PLAZA 4.0, Gymno PLAZA 1.0, NCBI RefSeq Plant and NCBI Nucleotide Database (NT) and NCBI Non-Redundant Protein Sequence Database (NR). Although TOA was primarily designed to work with woody plant species, it can also be used in the analysis of experiments on any type of plant organism. Additionally, NCBI Gene, InterPro and Gene Ontology databases are also used to complete the information.

TOA has a user-friendly front-end where researchers can perform the following tasks simply with a few clicks (Figure 1): i) download biochemical and biological data from genomic databases; ii) design the annotation pipeline that implements the most appropriate workflow for the experiment they are analyzing; iii) view statistics and graphics of results; and iv) track the status of processes during execution.
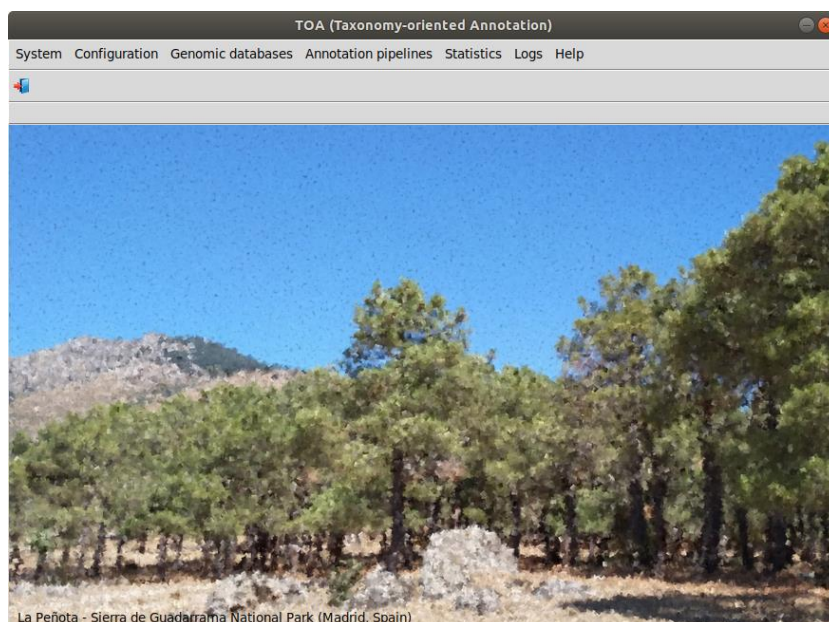
Figure. 1. Aspect of the program interface at startup.

We are going to provide an example to understand the functioning of TOA: We have carried out an RNA-seq experiment of Canary Island pine (*Pinus canariensis*) samples and we want to perform the functional annotation of the transcript sequences obtained in the *de novo* transcriptome assembly. First, we have to determine the genomic databases that we are going to use and the order in which we will carry out the processes. TOA is designed to explore the information in the databases sequentially. In this example, we will first explore the phylogenetically closest databases to the focal organism (in this case Plaza Gymno database), to end with the less specific database (NT). The database order is Gymno, Dicots, Monocots, RefSeq Plant and NT (selecting plant sequences). However, the order of the databases can be defined by the user.

Each time a specific database is queried is considered an iteration. An iteration has conceptually three steps (Figure 2):

(1) extraction of the information from the external genomic database and loading in the TOA database.

(2) performing an initial iteration of the full FASTA file with the nucleotide sequences against the sequences stored in the database selected in the first place.

(3) alignment of the sequences that were not annotated in the previous iteration against the existing sequences in the remaining database.

(4) annotation of the sequences aligned with the biochemical and biological information obtaining four files: (i) the alignments; (ii) the annotation description; (iii) the annotated sequences in FASTA format; and (iv) the non-annotated sequences in FASTA format. The file of non-annotated sequences will be the input for the alignment of the next iteration

If we consider using predicted peptides instead of nucleotides, it is necessary a previous step to extract ORFs and predict coding regions of the transcript sequences. In this case, we will use the NR database instead of the NT database.

Once the iterations have been performed for the selected databases, the output files generated in the different iterations are merged to have files with unified information. Summary statistics and plots are generated with these files. In nucleotide pipelines, a FASTA file with annotated transcripts is output as well.

The NT/NR iteration process yields an additional file with contamination sequences corresponding to non-viridiplantae species, i.e. fungi, bacteria, etc. Also, the last non annotated sequences file can hold chimeric sequences generated during the assembly stage, but they can correspond to novel genes that lack functional annotation.

The TOA output files can be used with other related software, e.g. the unified alignment file by Blast2Go or the frequency distribution per GO term file by REVIGO.

The plant annotation files of two pipelines can be merged in order to yield: a) a annotation files with annotation of both pipelines; or b) annotation of all first pipeline transcripts and annotation of second pipeline transcripts of those transcripts are not found in the first pepeline.
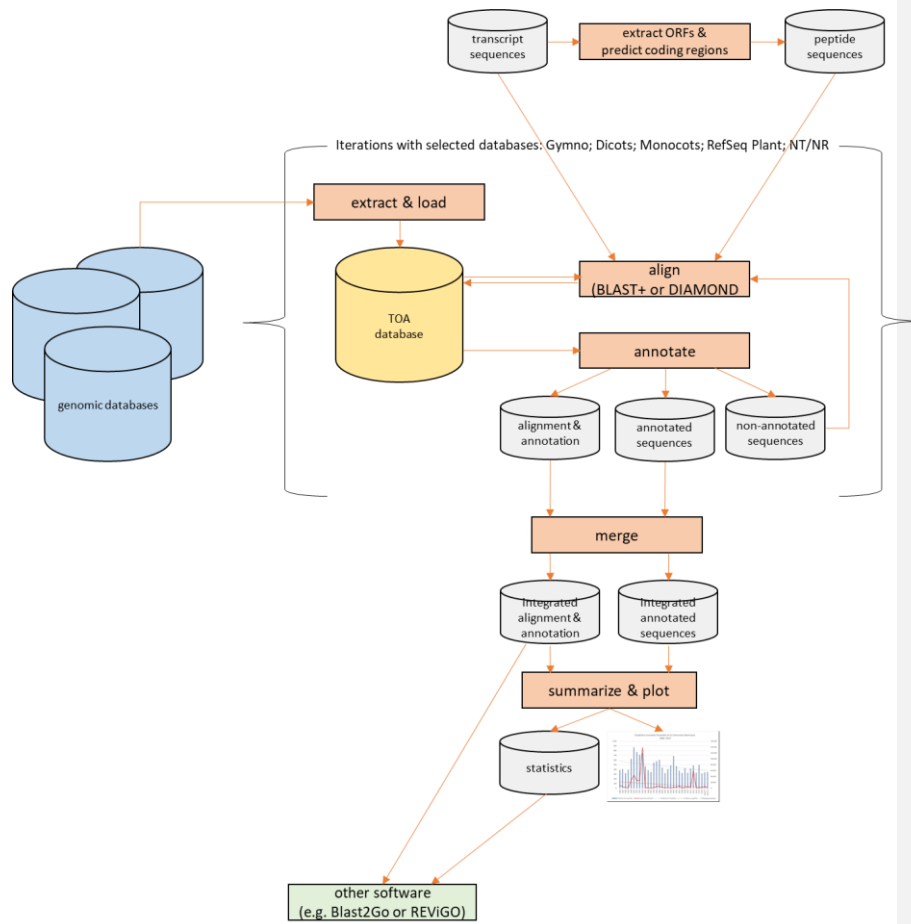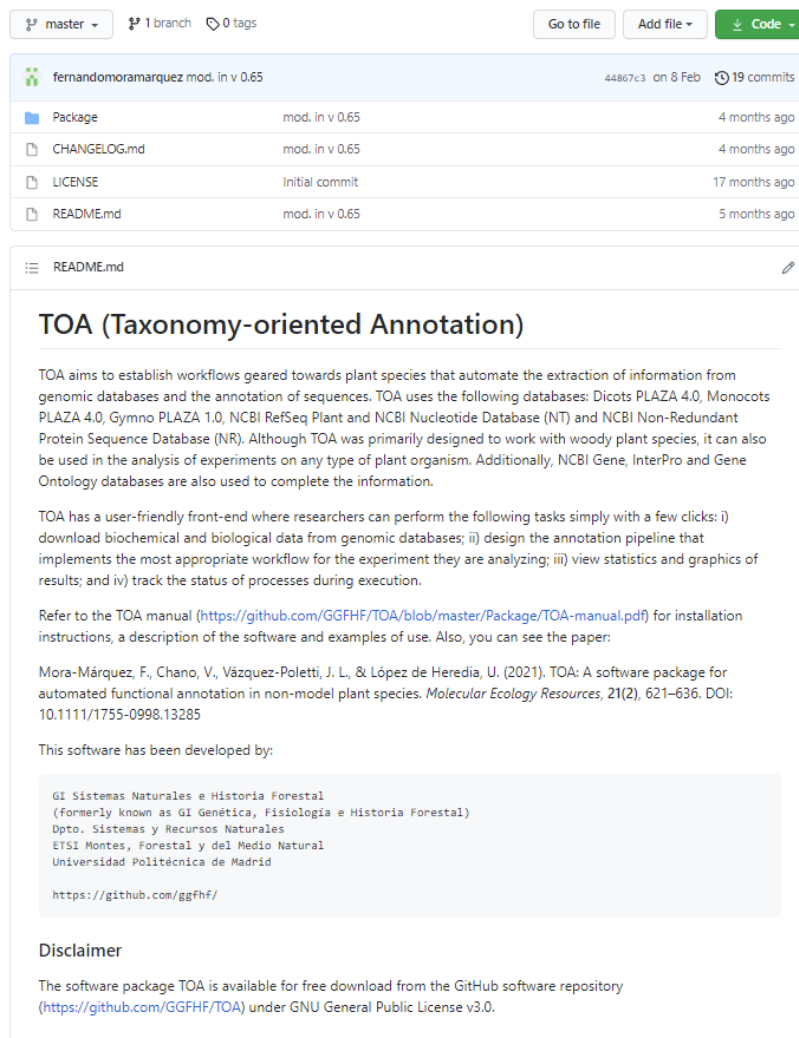
Figure. 2. Flowchart of sequence file annotation in TOA.

Installation

TOA installation

TOA was programmed in Python 3 and it generates dynamic Bash scripts to perform annotation pipelines. TOA runs in any computer with Linux or macOS.

TOA is available from the GitHub software repository of the GI Sistemas Naturales e Historia Forestal (https://github.com/GGFHF/TOA/), and it is distributed under GNU General Public License Version 3 (see Figure 3).

Figure. 3. TOA home at GitHub software repository.

To download TOA, click in *Code* and in the pup-up window click in *Download ZIP*.

To install TOA on Linux and macOS, simply decompress the TOA-master.zip into a directory, typing the following command in a terminal window:

```
$ unzip TOA-master.zip
```

Then, the execution permissions of the programs must be set by using this command:

```
$ chmod u+x *.py
```

Additional infrastructure software installation

Python 3, version 3.6 or higher, is necessary for a correct functioning of TOA. If Python 3 is not installed in your computer, you can download it from the official website (https://www.python.org/), or use one of the several distributions that include Python along with other software packages for standard bioinformatic analysis such as Anaconda (https://www.continuum.io/).

To work properly, TOA needs the following Python modules:

- Tkinter (https://docs.python.org/3.6/library/tkinter.html), the standard Python interface to the Tk GUI toolkit.
- Requests (https://requests.kennethreitz.org/en/master/), an HTTP library.
- Plotnine (https://plotnine.readthedocs.io/en/stable/), an implementation of a grammar of graphics in Python based on ggplot2

Next, we present how to install this additional software in two example environments: a) an Ubuntu Linux 18.04 (Bionic Beaver) where Python3 is pre-installed in the OS; and b) a macOS 10.13 (High Sierra) where Python is installed using Anaconda.

Ubuntu Linux 18.04

First, open a terminal window and type the following commands to install the Python 3 modules Tk, PIL and PIL.ImageTk and pip3, if necessary:

```
$ sudo apt-get install python3-tk python3-pil python3-pil.imagetk
```

```
$ sudo apt-get install python3-pip
```

Then install the Requests library, typing the following command in the terminal window, if necessary:

```
$ sudo pip3 install requests
```

And finally, install the Plotine library, typing the following command in the terminal window, if necessary:

```
$ sudo pip3 install plotnine
```

macOS 13.10

First, install Homebrew and wget command, if necessary, typing the following commands in the terminal window:

```
$ /usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

```
$ brew install wget
```

Now download the Anaconda software file, e.g. the version 2019.09, typing the command:

```
$ wget https://repo.anaconda.com/archive/Anaconda3-2019.07-MacOSX-x86_64.sh
```

And provide execution permission to this file and run it typing the commands:

```
$ chmod u+x Anaconda3-2019.07-MacOSX-x86_64.sh
```

```
$ ./Anaconda3-2019.07-MacOSX-x86_64.sh
```

During the Anaconda installation, read the Anaconda End User License Agreement, accept the license terms and indicate the location where Anaconda will be installed. Later, review that the PATH variable incorporates the directory where Python is. So, if Anaconda installation directory is *Anaconda3_path*, edit *.bash_profile* file in the user's home directory and check the directory *Anaconda3_path/bin*. is added in the PATH variable through a sentence like this:

```
export PATH=Anaconda3_path/bin:$PATH
```

The appearance of the file *.bash_profile* must be similar to the file shown in the Figure 4.

```
# added by Anaconda3 2.3.0 installer
export PATH="/Users/jmoramarquez/bioinfo/anaconda3/bin:$PATH"

# >>> conda initialize >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$('/Users/jmoramarquez/bioinfo/anaconda3/bin/conda' 'shell.bash' 'hook' 2> /dev/null)"
if [ $? -eq 0 ]; then
    eval "$__conda_setup"
else
    if [ -f "/Users/jmoramarquez/bioinfo/anaconda3/etc/profile.d/conda.sh" ]; then
        . "/Users/jmoramarquez/bioinfo/anaconda3/etc/profile.d/conda.sh"
    else
        export PATH="/Users/jmoramarquez/bioinfo/anaconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda initialize <<<
```

Figure. 4. An example of the file *.bash_profile*.

Finally, install Plotnine, because other modules are by default when Anaconda is installed. To install it, type the following command:

```
$ conda install --channel conda-forge plotnine
```

Starting TOA

TOA runs in graphical mode using the graphical user interface (GUI), but it can also be run in console mode on server machines without GUI installed.

Start TOA in GUI mode typing the following command in a terminal window in the directory where the package of TOA is downloaded:

$ ./TOA.py

Alternatively, you can type also:

$ ./TOA.py --mode=gui

To run TOA in console mode:

$ ./TOA.py --mode=console

Initial appearance of TOA at application startup in GUI mode is shown in Figure 1.

TOA Docker

If you are a Docker user, you can run the TOA image typing using a Bash script with the following instructions:

```
        LOCAL_MINICONDA_DIR=/home/$USER/Docker-Miniconda3
        LOCAL_CONFIG_DIR=/home/$USER/Docker-TOA/config
        LOCAL_DATABASE_DIR=/home/$USER/Docker-TOA/databases
        LOCAL_LOGS_DIR=/home/$USER/Docker-TOA/logs
        LOCAL_RESULT_DIR=/home/$USER/Docker-TOA/results
        LOCAL_TEMP_DIR=/home/$USER/Docker-TOA/temp
        LOCAL_SWAP_DIR=/home/$USER/Docker-TOA/swap

        mkdir --parents $LOCAL_MINICONDA_DIR
        mkdir --parents $LOCAL_CONFIG_DIR
        mkdir --parents $LOCAL_DATABASE_DIR
        mkdir --parents $LOCAL_LOGS_DIR
        mkdir --parents $LOCAL_RESULT_DIR
        mkdir --parents $LOCAL_TEMP_DIR
        mkdir --parents $LOCAL_SWAP_DIR

        xhost + local: 2>&1 > /dev/null

        docker run \
            --name toa-container \
            --volume /tmp/.X11-unix:/tmp/.X11-unix \
            --volume "$LOCAL_MINICONDA_DIR":/Docker/TOA/Miniconda3 \
            --volume "$LOCAL_CONFIG_DIR":/Docker/TOA/config \
            --volume "$LOCAL_DATABASE_DIR":/Docker/TOA/TOA-databases \
            --volume "$LOCAL_LOGS_DIR":/Docker/TOA/logs \
            --volume "$LOCAL_RESULT_DIR":/Docker/TOA/TOA-results \
```

```
    --volume "$LOCAL_TEMP_DIR":/Docker/TOA/temp \
    --volume "$LOCAL_SWAP_DIR":/Docker/TOA/TOA-swap \
    --env DISPLAY=unix$DISPLAY \
    --device /dev/snd \
    fernandomoramarquez/toa
```

The variables LOCAL_MINICONDA_DIR, LOCAL_CONFIG_DIR, LOCAL_DATABASE_DIR, LOCAL_LOGS_DIR, LOCAL_RESULT_DIR, LOCAL_TEMP_DIR and LOCAL_SWAP_DIR are used to link the TOA container directories with local directories in order to hold the data persistence if you remove the container.

The variable LOCAL_SWAP_DIRECTORY is used to pass files to be processed by TOA, such as transcriptome files.

The default values of these four variables can be modified to fit to your preferences.

First steps

TOA menus

TOA is structured in several menus:

System

Just to exit the application.

Configuration

This menu contains all the items related to:

- Recreate TOA config file
- View TOA config file
- Recreate TOA database
- Rebuild TOA database
- Install Bioinfo software

Genomic databases

Here, all options related to Genomic databases are shown:

- Basic data
    - Recreate genomic dataset file
    - Edit genomic dataset file
    - Recreate species file
    - Edit species file
    - Download other basic data
    - Load data into TOA database
- Gymno PLAZA 1.0
    - Build proteome
    - Download functional annotations from PLAZA server
    - Load data into TOA database
- Dicots PLAZA 1.0
    - Build proteome
    - Download functional annotations from PLAZA server
    - Load data into TOA database
- Monocots PLAZA 1.0
    - Build proteome
    - Download functional annotations from PLAZA server
    - Load data into TOA database
- NCBI RefSeq Plant
    - Build proteome

- NCBI Plant database NT
  - Build BLAST database
- NCBI Plant database NR
  - Build BLAST database
- NCBI Gene
  - Download functional annotations from NCBI server
  - Load data into TOA database
- InterPro
  - Download functional annotations from InterPro server
  - Load data into TOA database
- Gene Ontology
  - Download functional annotations from Gene Ontology server
  - Load data into TOA database

Annotation pipelines menu

The options included here allow to build and run scripts to perform the annotation process:

- TOA nucleotide pipeline
  - Recreate config file
  - Edit config file
  - Run pipeline
  - Restart pipeline
- TOA amino acid pipeline
  - Recreate config file
  - Edit config file
  - Run pipeline
  - Restart pipeline
- Annotation merger of TOA pipelines
  - Recreate config file
  - Edit config file
  - Run process

Statistics

This menu contains all the items related to statistics corresponding to the annotation pipelines results and their plots:

- Alignment
  - # HITs per # HSPs data
  - # HITs per # HSPs plot
- Annotation datasets
  - Frequency distribution data
  - Frequency distribution plot
- Species
  - Frequency distribution data
  - Frequency distribution plot

- Family
  - Frequency distribution data
  - Frequency distribution plot
- Phylum
  - Frequency distribution data
  - Frequency distribution plot
- EC
  - Frequency distribution data
  - Frequency distribution plot
  - # sequences per # ids data
  - # sequences per # ids plot
- Gene Ontology
  - Frequency distribution per term data
  - Frequency distribution per term plot
  - Frequency distribution per namespace data
  - Frequency distribution per namespace plot
  - # sequences per # terms data
  - # sequences per # terms plot
- InterPro
  - Frequency distribution data
  - Frequency distribution plot
  - # sequences per # ids data
  - # sequences per # ids plot
- KEGG
  - Frequency distribution data
  - Frequency distribution plot
  - # sequences per # ids data
  - # sequences per # ids plot
- MapMan
  - Frequency distribution data
  - Frequency distribution plot
  - # sequences per # ids data
  - # sequences per # ids plot
- MetaCyc
  - Frequency distribution data
  - Frequency distribution plot
  - # sequences per # ids data
  - # sequences per # ids plot

Logs menu

This menu allows the access to the application logs:

- View submission logs
- View result logs

Help menu

It contains the documentation of the application.

Configuring the TOA environment

When TOA starts for the first time it is required to configure the TOA environment. To do so, we select the menu item with the following path:

Main menu > Configuration > Recreate TOA config file

The Figure 5 shows the window corresponding to this menu item. Default values are presented for *Miniconda directory*, *Database directory* and *Result directory*. If necessary, modify them and press the button *[Execute]*.
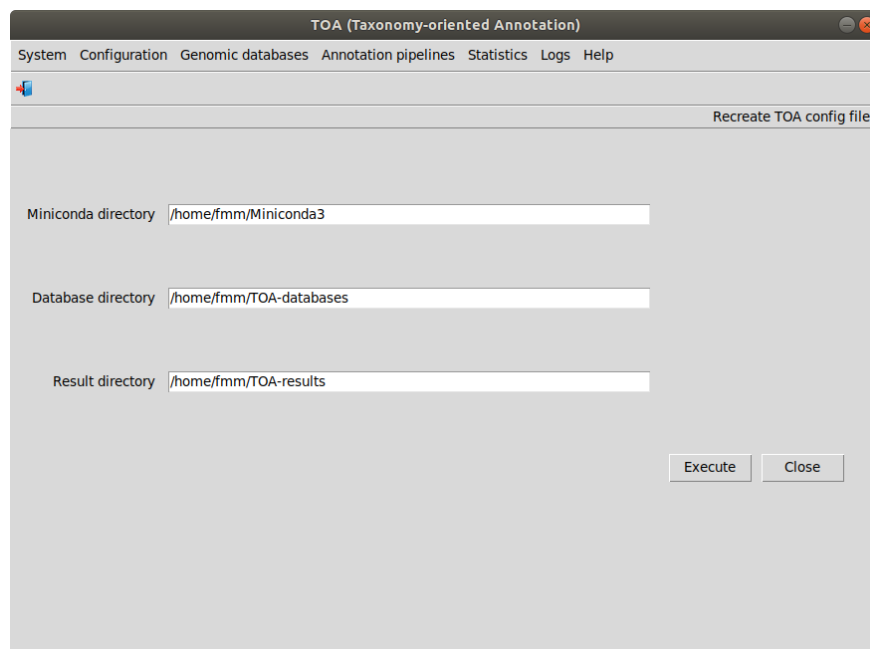

Figure. 5. TOA window corresponding to the menu item *Recreate TOA config file*.

Installing bioinformatic software

TOA needs the following bioinformatic software:

- TransDecoder (https://github.com/TransDecoder/TransDecoder). It is used to extract ORFs and predict coding regions within transcripts.
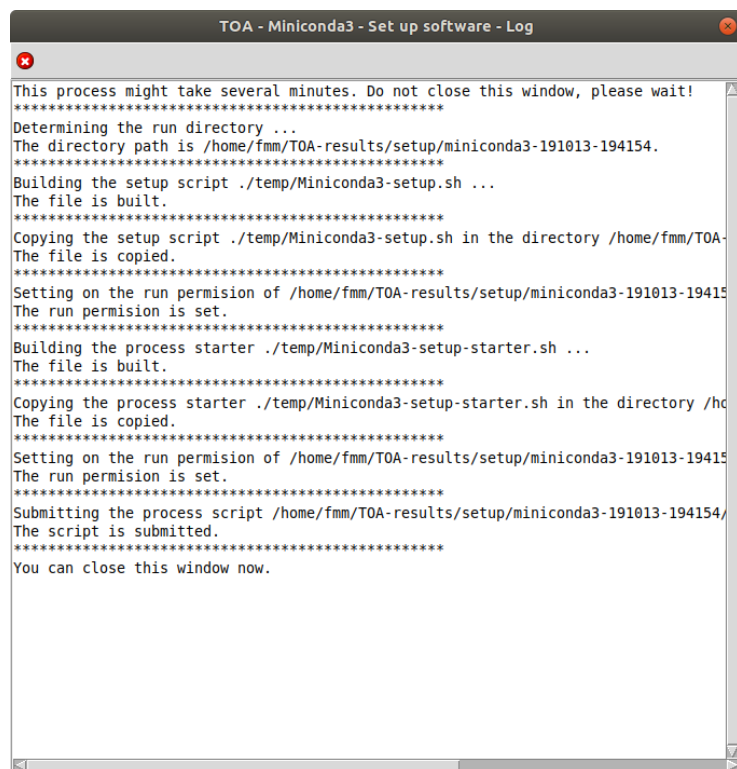
- BLAST+ (https://blast.ncbi.nlm.nih.gov/). It is used to find local similarity between transcripts or predicted peptides and sequences of genomic databases.
- DIAMOND (http://www.diamondsearch.org/) is a faster BLAST alternative (500x-20,000x) to align protein and predicted peptides though it reports fewer matches.
- Entrez Direct (https://www.ncbi.nlm.nih.gov/books/NBK179288/). It is a software package to download NCBI sequence identifiers of nucleotides or proteins.

The installation of these packages is automatic by using Bioconda (https://bioconda.github.io/), a channel of the Conda (https://conda.io/) package manager, which holds a large number of bioinformatics software packages.

First, install Miniconda·(Bioconda infrastructure) selecting the menu item with this path:

Main menu > Configuration > Bioinfo software installation > Miniconda3 [Execute]

Press the bottom *[Execute]*. A pop-up window will display the submission log (Figure 6).


```
TOA - Miniconda3 - Set up software - Log

This process might take several minutes. Do not close this window, please wait!
***************************************************
Determining the run directory ...
The directory path is /home/fmm/TOA-results/setup/miniconda3-191013-194154.
***************************************************
Building the setup script ./temp/Miniconda3-setup.sh ...
The file is built.
***************************************************
Copying the setup script ./temp/Miniconda3-setup.sh in the directory /home/fmm/TOA-
The file is copied.
***************************************************
Setting on the run permision of /home/fmm/TOA-results/setup/miniconda3-191013-19415
The run permision is set.
***************************************************
Building the process starter ./temp/Miniconda3-setup-starter.sh ...
The file is built.
***************************************************
Copying the process starter ./temp/Miniconda3-setup-starter.sh in the directory /ho
The file is copied.
***************************************************
Setting on the run permision of /home/fmm/TOA-results/setup/miniconda3-191013-19415
The run permision is set.
***************************************************
Submitting the process script /home/fmm/TOA-results/setup/miniconda3-191013-194154/
The script is submitted.
***************************************************
You can close this window now.
```

Figure.6. Submission log of a Miniconda3 installation process.

To view the process log during and after the run, select the menu item with this path:

Main menu > Logs > View result logs

In the raised window (Figure 7), select **installations** in the *Process type* combo-box, and then press the button *[Execute]*.
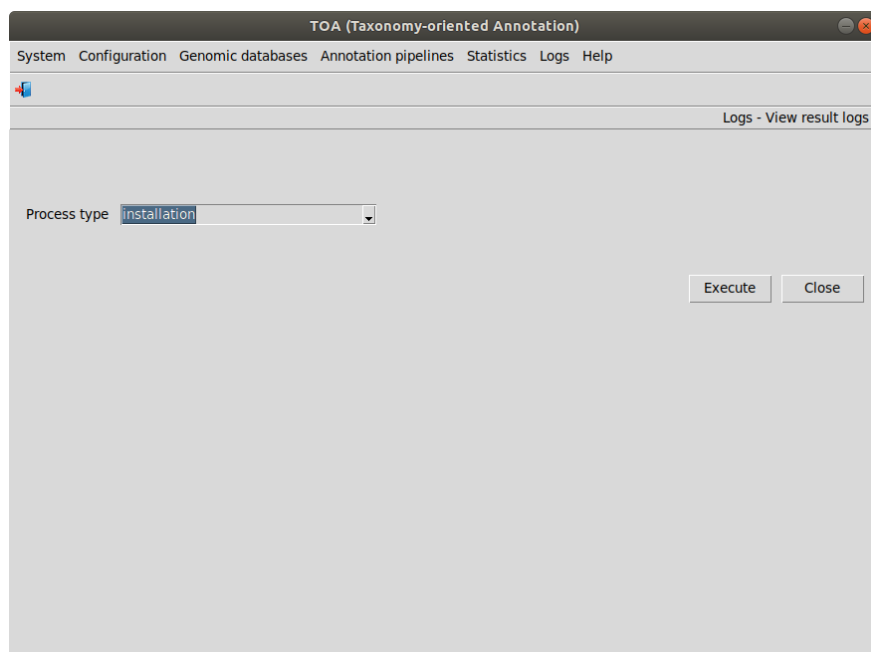

Figure. 7. TOA window corresponding to the menu item *Logs - View result logs*.

Now, a pop-up window with all installation processes is shown (Figure 8).

Figure. 8. List of installation processes.

So far, we have only performed a single installation process: the process **miniconda3-20200219-163343** corresponding to the last (and unique) Miniconda3 installation. Clicking on it, a new pop-up window appears with its corresponding log (Figure 9).

Figure. 9. Begin and end of a Miniconda3 installation process.

In the toolbar, there is a button to refresh the run status. Clicking it, the log will be updated.

All the process logs have:

- A header with the time when it started.
- At the bottom, a summary with the status (OK, if all the programs have ended without errors; WRONG, otherwise), the end time, and the duration of the script run.

When Miniconda3 is installed, we install BLAST+, Entrez Direct and TransDecoder selecting the menu items with these paths:

Main menu > Configuration > Bioinfo software installation > BLAST+ [Execute]

Main menu > Configuration > Bioinfo software installation > DIAMOND [Execute]

Main menu > Configuration > Bioinfo software installation > Entrez Direct [Execute]

Main menu > Configuration > Bioinfo software installation > TransDecoder [Execute]

The installation steps of each software are like for Miniconda3.

Consulting submitted processes and troubleshooting

The correct operability of the submitted processes is controlled by logs similar to the Miniconda3 installation (see Figure 9). So, the user can monitor the performance of the process at any time and detect problems. Please, confirm that each process is ended before submitting other one.

A step by step example

We have sequencing data corresponding to an RNA-seq Illumina library of an experiment about the process of cicatrization after wounding the xylem of the stem of the Canary Island pine (*Pinus canariensis*) and in this example, we are going to annotate the candidate peptides extracted from the transcriptome build in the assembly step using the following databases Gymno PLAZA 1.0, Dicots PLAZA 4.0, Monocots PLAZA 4.0, NCBI RefSeq Plant and NCBI Non-Redundant Protein Sequence Database (NR), in this order. In addition to these databases, we have to complete the biological data with other databases: NCBI Gene, InterPro and Gene Ontology.

The steps that we are going to do are:

a) Load genomic data into TOA database of:
   - Basic data
   - Gymno PLAZA 1.0
   - Dicots PLAZA 4.0
   - Monocots PLAZA 4.0
   - NCBI RefSeq Plant
   - NCBI BLAST database NR
   - NCBI Gene
   - InterPro
   - Gene Ontology
b) Create and run an annotation pipeline (amino acid)
c) Consult statistics data and show plots

The data loading of a genomic database into TOA database has to be performed once, and we only have to repeat the loading of a database when it is necessary to update its data.

The submit logs of genomic data loading can be reviewed selecting **TOA-databases** in the combo-box *Experiment/process* in the window *Logs - View result logs* (see Figure 7). And the logs of annotation pipelines can be consulted selection **TOA-pipelines** in this combo-box.

Load genomic data into TOA database

Basic data

The basic data loading requires performing several tasks. First, we create the genomic dataset file selecting the menu item with this path:

Main menu > Genomic databases > basic data > Recreate genomic dataset file [Execute]

Then, we create the species file:

Main menu > Genomic databases > basic data > Recreate species file [Execute]

Now, we download other basic data:

Main menu > Genomic databases > basic data > Download other basic data [Execute]

Finally, we load the dataset file, species file and other basic data into TOA database:

Main menu > Genomic databases > basic data > Load basic data into TOA database [Execute]

Gymno PLAZA 1.0

To Gymno PLAZA 1.0, first, we build the proteome selecting the menu item with this path:

Main menu > Genomic databases > Gymno PLAZA 1.0 > Build proteome [Execute]

Then, we download the functional annotations:

Main menu > Genomic databases > Gymno PLAZA 1.0 > Download functional annotation [Execute]

Finally, we load into TOA database:

Main menu > Genomic databases > Gymno PLAZA 1.0 > Load data into TOA database [Execute]

Dicots PLAZA 4.0

For Dicots PLAZA 4.0, how to proceed is similar to Gymno PLAZA 1.0.

Monocots PLAZA 4.0

For Monocots PLAZA 4.0, how to proceed is similar to Gymno PLAZA 1.0.

NCBI RefSeq Plant

In the case of this database, a task is only necessary, build its proteome:

Main menu > Genomic databases > NCBI RefSeq Plant > Build proteome [Execute]

NCBI BLAST database NR

To NCBI BLAST database NR, we only carry out the task of building the BLAST database:

Main menu > Genomic databases > NCBI BLAST database NR > Build BLAST database [Execute]

NCBI Gene

To NCBI Gene, first, we download the functional annotations:

Main menu > Genomic databases > NCBI Gene > Download functional annotation [Execute]

And then, we load into TOA database:

Main menu > Genomic databases > NCBI Gene > Load data into TOA database [Execute]

InterPro

For InterPro, the procedure is similar to the case of NCBI Gene.

Gene Ontology

For Gene Ontology, the procedure is similar to the case of NCBI Gene.

Create and run an annotation pipeline (amino acid)

Create the config file

The first task to perform an annotation process with TOA is to build a config file that contains data about the transcriptome directory and file, selected databases and their order, and BLAST parameters. Taking into account these data, TOA will be able to build dynamically a Bash script corresponding to the annotation pipeline.

We create a new config file selecting the menu item with this path:

Main menu > Annotation pipelines > TOA amino acid pipeline > Recreate config file [Execute]

Then, TOA presents a window (see Figure 10) where we indicate the *Transcriptome directory* selecting it with the corresponding button. Once this is done, we select the file in the combo-box *Transcriptome file*.
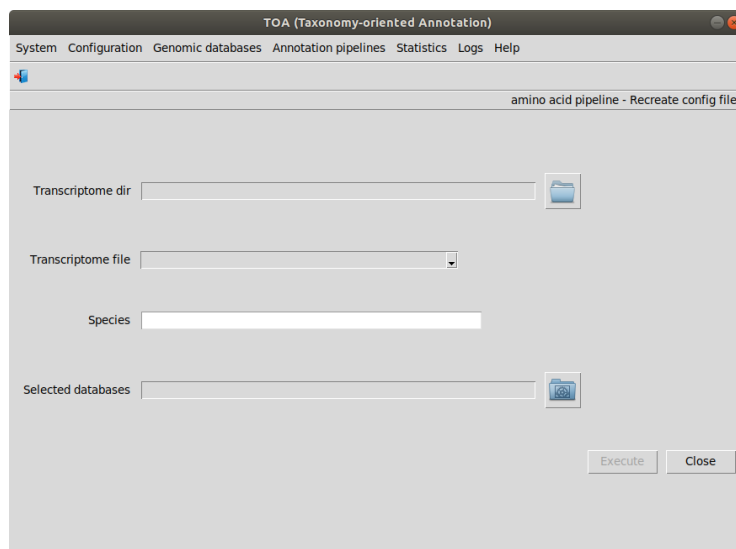
Figure. 10. TOA window corresponding to the menu item *amino acid pipeline - Recreate config file*.

Next, we type the *Pinus canariensis* in the combo-box *Species*. Then TOA load a default value in the entry field *Selected databases* (See Figure 11).
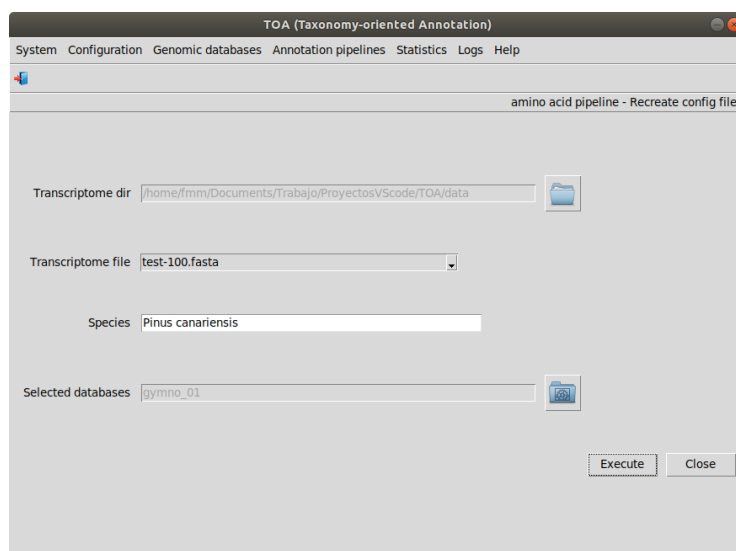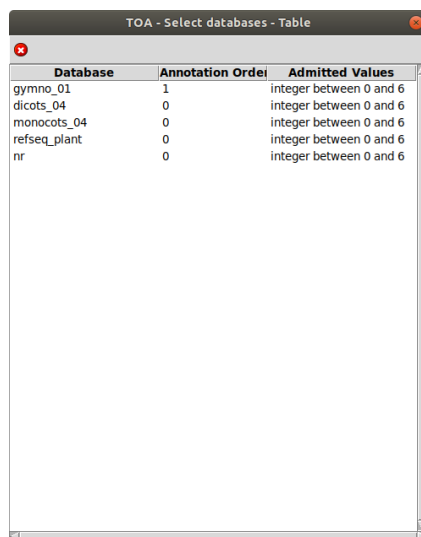

Figure. 11. TOA window corresponding to the menu item *amino acid pipeline - Recreate config file* once the species has been typed.
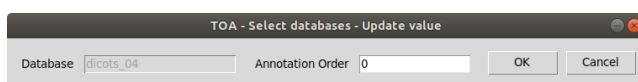
Last, if we consider that the analysis must be done with other databases, we click the corresponding button and a pop-up window is displayed (see Figure 12).

**Figure. 12**. TOA window where the databases the bases involved in the annotation are selected

We need the following databases in our example: Gymno PLAZA 01, Dicots PLAZA 04, Monocots PLAZA 04, RefSeq Plant and NR (in this order). *gymno_01* (Gymno PLAZA 01) is the default value. Then we click on the second database, *dicots_04* (Dicots PLAZA 04). A new pop-up window is shown (see Figure 13). In the entry field *Annotation order*, we type 2 (0 means not selected; 1, the first database; 2, the second one; etc.) and press button [OK].

Figure. 13. TOA window where the annotation order of a selected database is set

We will proceed with the remaining databases in a similar way as with Dicots PLAZA 04. When we are done with all the databases involved in the annotation, we will close the window displayed in Figure 10. Now, the appearance of the window *amino acid pipeline - Recreate config file* will be similar to the one in the Figure 14.
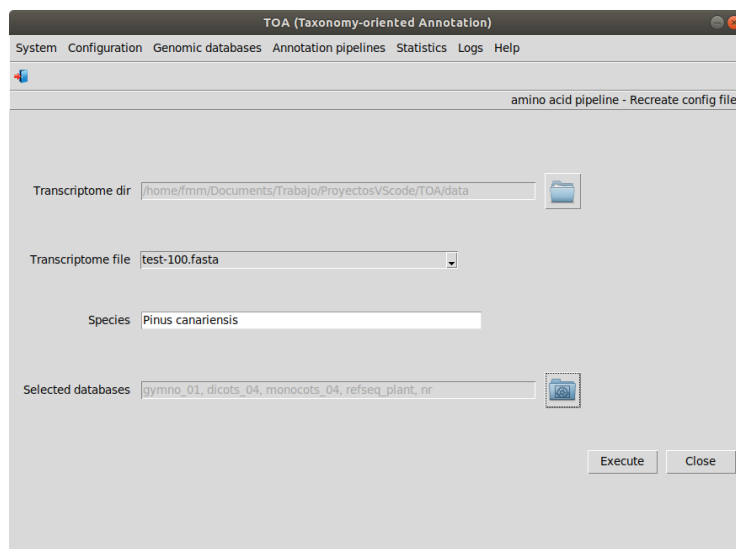
Figure. 14. TOA window corresponding to the menu item *amino acid pipeline - Recreate config file* once the data has been filled.

Then we press the button [Execute]. In the pop-up window (see Figure 15), we can examine the config file. In this example, it has five sections: *identification*, with the directory and file of the transcriptome; *database parameters*, where the order of the databases is set; *pipeline parameters*, where the aligner software (BLAST+ o DIAMOND) and the threat number is set; and *BLAST parameters* where we can modify the thread number, e_value, the maximum number of aligned sequences, the maximum number of HSPs and the minimum coverage of alignments when we use BLAST+; and *DIAMOND parameters* where we can modify the thread number, e_value, the maximum number of aligned sequences and the maximum number of HSPs `when we use DIAMOND. These last section have the possibility of pass additional parameters to the aligner
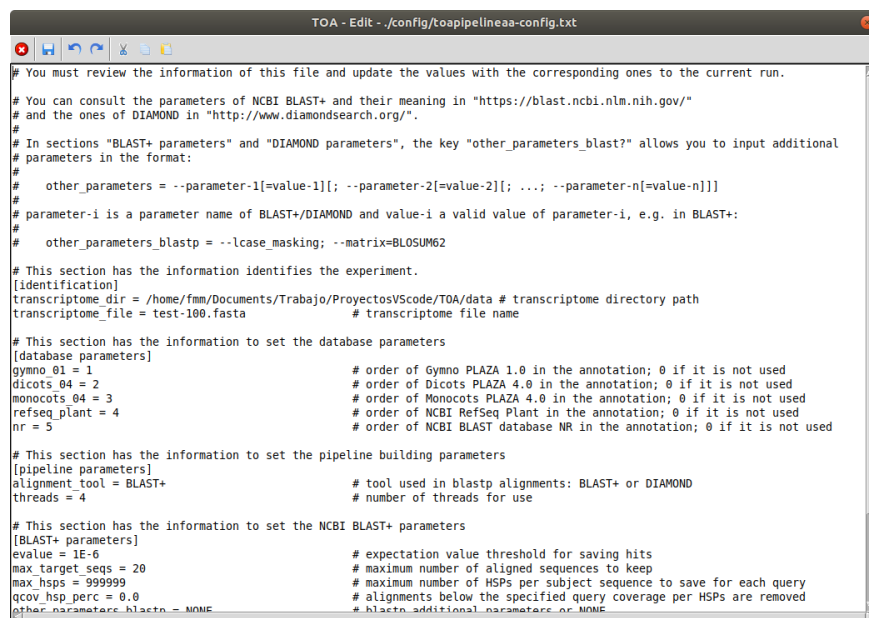
```
TOA - Edit - ./config/toapipelineaa-config.txt

# You must review the information of this file and update the values with the corresponding ones to the current run.

# You can consult the parameters of NCBI BLAST+ and their meaning in "https://blast.ncbi.nlm.nih.gov/"
# and the ones of DIAMOND in "http://www.diamondsearch.org/".
#
# In sections "BLAST+ parameters" and "DIAMOND parameters", the key "other_parameters_blast?" allows you to input additional
# parameters in the format:
#
#    other_parameters = --parameter-1[=value-1][; --parameter-2[=value-2][; ...; --parameter-n[=value-n]]]
#
# parameter-i is a parameter name of BLAST+/DIAMOND and value-i a valid value of parameter-i, e.g. in BLAST+:
#
#    other_parameters_blastp = --lcase_masking; --matrix=BLOSUM62

# This section has the information identifies the experiment.
[identification]
transcriptome_dir = /home/fmm/Documents/Trabajo/ProyectosVScode/TOA/data # transcriptome directory path
transcriptome_file = test-100.fasta                  # transcriptome file name

# This section has the information to set the database parameters
[database parameters]
gymno_01 = 1                                # order of Gymno PLAZA 1.0 in the annotation; 0 if it is not used
dicots_04 = 2                               # order of Dicots PLAZA 4.0 in the annotation; 0 if it is not used
monocots_04 = 3                             # order of Monocots PLAZA 4.0 in the annotation; 0 if it is not used
refseq_plant = 4                            # order of NCBI RefSeq Plant in the annotation; 0 if it is not used
nr = 5                                      # order of NCBI BLAST database NR in the annotation; 0 if it is not used

# This section has the information to set the pipeline building parameters
[pipeline parameters]
alignment_tool = BLAST+                     # tool used in blastp alignments: BLAST+ or DIAMOND
threads = 4                                 # number of threads for use

# This section has the information to set the NCBI BLAST+ parameters
[BLAST+ parameters]
evalue = 1E-6                               # expectation value threshold for saving hits
max_target_seqs = 20                        # maximum number of aligned sequences to keep
max_hsps = 999999                           # maximum number of HSPs per subject sequence to save for each query
qcov_hsp_perc = 0.0                         # alignments below the specified query coverage per HSPs are removed
other_parameters_blastp = NONE              # blastp additional parameters or NONE
```

Figure. 15. Editable TOA window where the config file of an amino acid pipeline can be modified.

Edit the config file

We can review and modify the current config file selecting:

Main menu > Annotation pipelines > TOA amino acid pipeline > Edit config file [Execute]
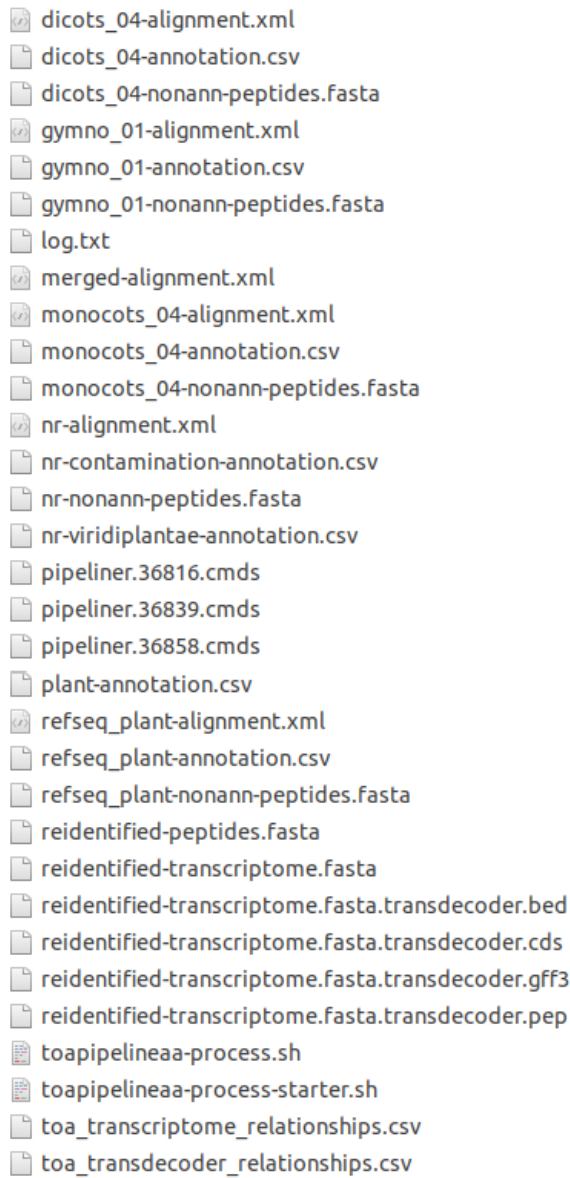
Run a pipeline

We run the pipeline corresponding to the current config file selecting:

Main menu > Annotation pipelines > TOA amino acid pipeline > Run pipeline [Execute]

Then, a Bash script is dynamically built and run.

When the process end, we could go the subdirectory of TOA-results\TOA-pipelines corresponding to the script run and consult the generated files (see Figure 16). There are 3 files per database. E.g. gymno_01-alignment.xml (the BLAST alignment with the Gymno PLAZA 01 proteome), gymno_01-annotation.csv (the annotation file of aligned candidate peptides) and gymno_01-nonann-peptides.fasta (a FASTA file with the non-annotated candidate peptides). Also, there are other two files that join files of plant alignment, plant-alignment.xml, and files of plant alignment, plant-annotation.csv. In the subdirectory stats, there are the CSV files of statistics of alignment, annotation datasets, phylogenetic information and function.

dicots_04-alignment.xml
dicots_04-annotation.csv
dicots_04-nonann-peptides.fasta
gymno_01-alignment.xml
gymno_01-annotation.csv
gymno_01-nonann-peptides.fasta
log.txt
merged-alignment.xml
monocots_04-alignment.xml
monocots_04-annotation.csv
monocots_04-nonann-peptides.fasta
nr-alignment.xml
nr-contamination-annotation.csv
nr-nonann-peptides.fasta
nr-viridiplantae-annotation.csv
pipeliner.36816.cmds
pipeliner.36839.cmds
pipeliner.36858.cmds
plant-annotation.csv
refseq_plant-alignment.xml
refseq_plant-annotation.csv
refseq_plant-nonann-peptides.fasta
reidentified-peptides.fasta
reidentified-transcriptome.fasta
reidentified-transcriptome.fasta.transdecoder.bed
reidentified-transcriptome.fasta.transdecoder.cds
reidentified-transcriptome.fasta.transdecoder.gff3
reidentified-transcriptome.fasta.transdecoder.pep
toapipelineaa-process.sh
toapipelineaa-process-starter.sh
toa_transcriptome_relationships.csv
toa_transdecoder_relationships.csv

Figure. 16. List of directories and files generated by the example annotation pipeline.

Consult statistics data and show plots

TOA pipelines generate statistics of alignment, annotation datasets, phylogenetic information and function. In this tutorial, we are only going to consult some of these statistics.

Alignment - # HITs per HSPs

We consult # HITs per # HSPs data of the alignment selecting the menu item with this path:
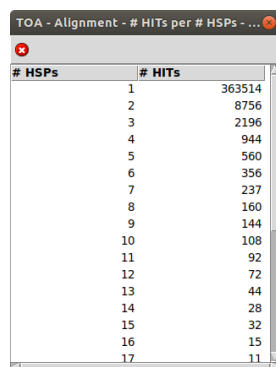
Main menu > Statistics > Alignment > # HITs per # HSPs data [Execute]

In the raised window (see Figure 17), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. Then we press the button *[Execute].*
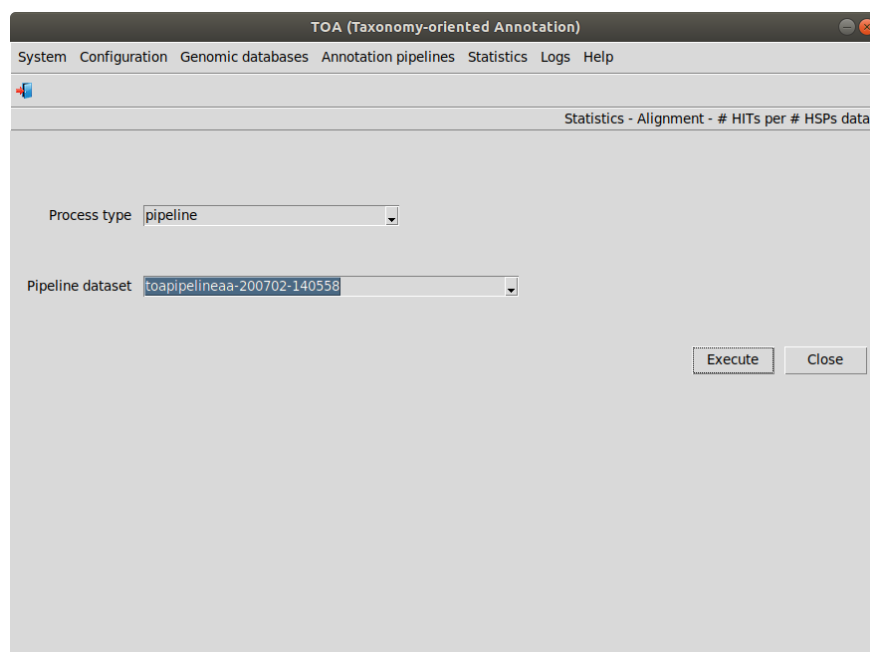
Figure. 17. TOA window corresponding to the menu item *Statistics - Alignment - # HITS per # HSPs data*.

Data are listed in a pop-up window (see Figure 18).
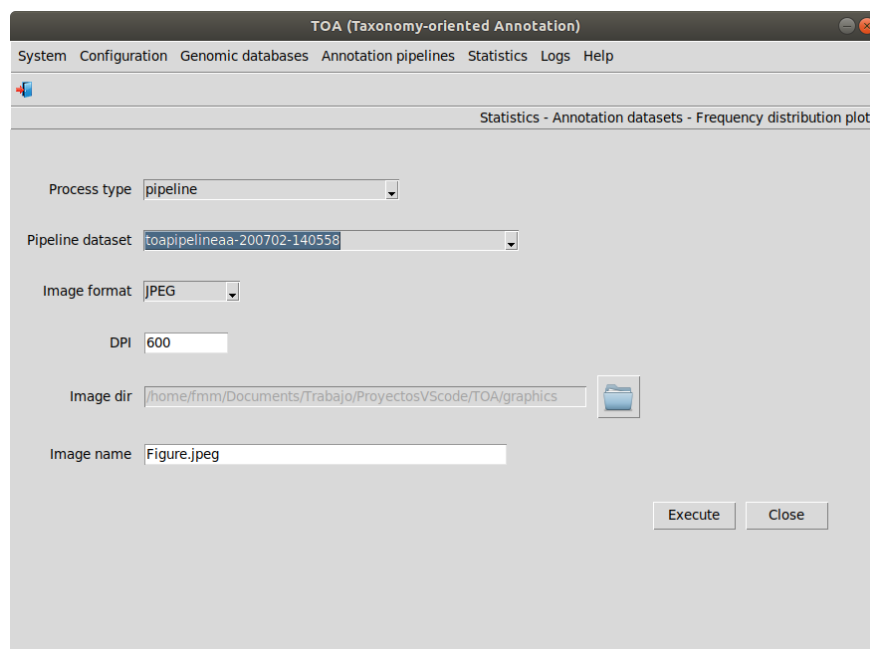
Figure. 18. # HITs per # HSPs data of the alignment generated by the example annotation pipeline.

We view the graphic corresponding to # HITs per # HSPs data of the alignment selecting:

Main menu > Statistics > Alignment > # HITs per # HSPs plot [Execute]

In the raised window (see Figure 19), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. The default value of the remaining fields (*Image format*, *DPI*, *Image dir* and *Image name*) can be modified if necessary. Then we press the button *[Execute]*.

Figure. 19. TOA window corresponding to the menu item *Statistics - Alignment - # HITS per # HSPs plot*.

The plot is saved in the selected directory in *Image dir* and is displayed in a pop-up window (see Figure 20).

Alignment - # HITs per # HSPs
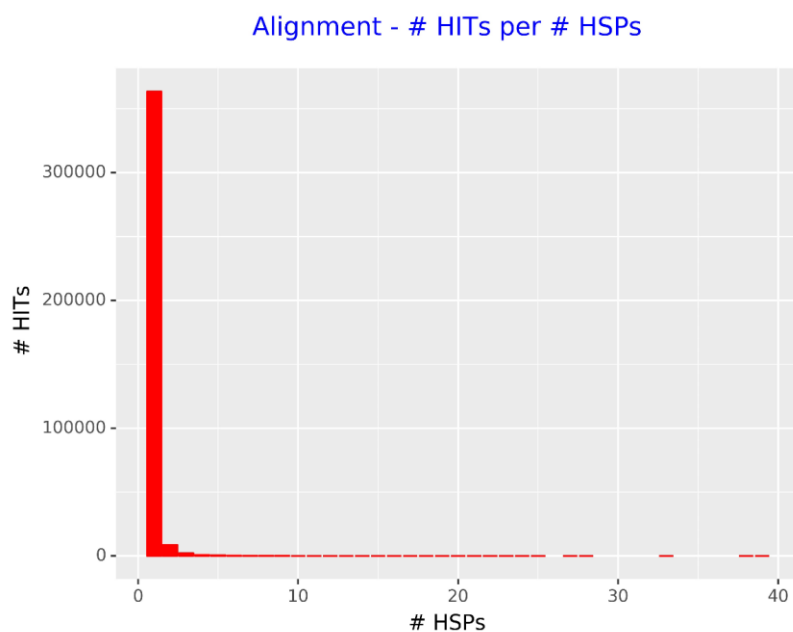
Figure. 20. Plot of # HITs per # HSPs data of the alignment generated by the example annotation pipeline.

Annotation datasets - Frequency distribution

We can consult the frequency distribution of annotation datasets selecting the menu item with this path:

Main menu > Statistics > Annotation datasets > Frequency distribution data [Execute]

In the raised window (see Figure 21), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. Then, we press the button *[Execute]*.
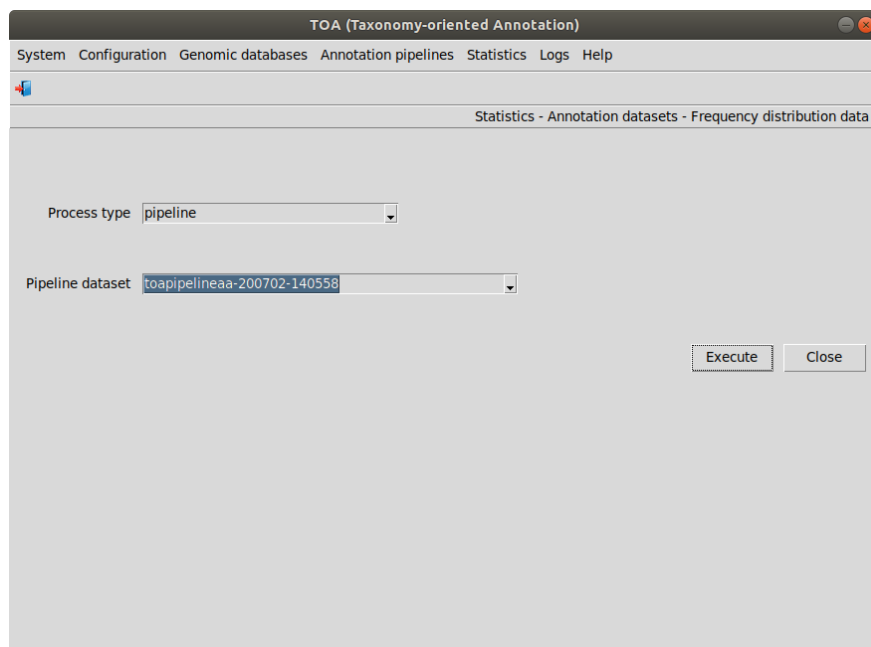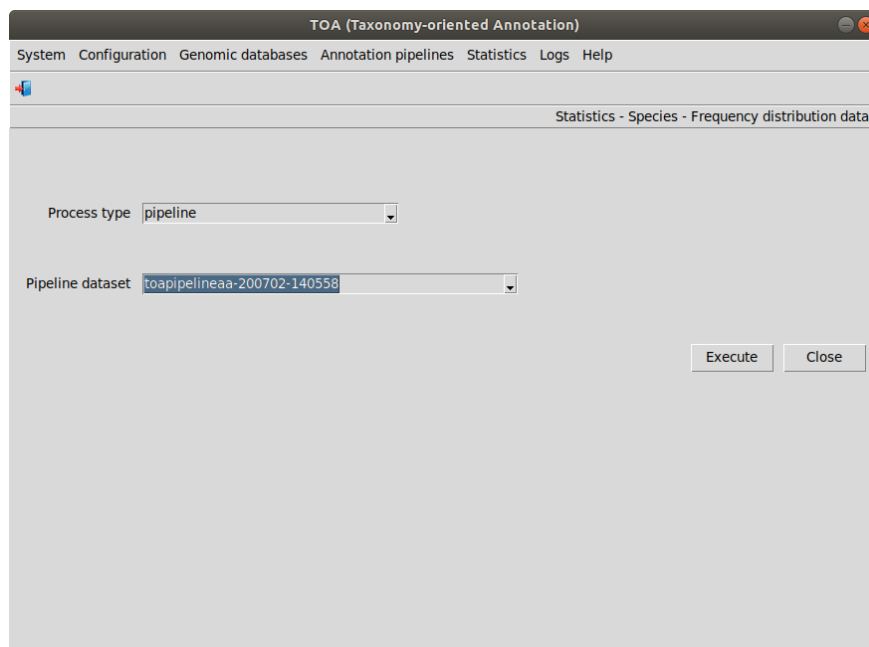
Figure. 21. TOA window corresponding to the menu item *Statistics - Annotation datasets - Frequency distribution data*.

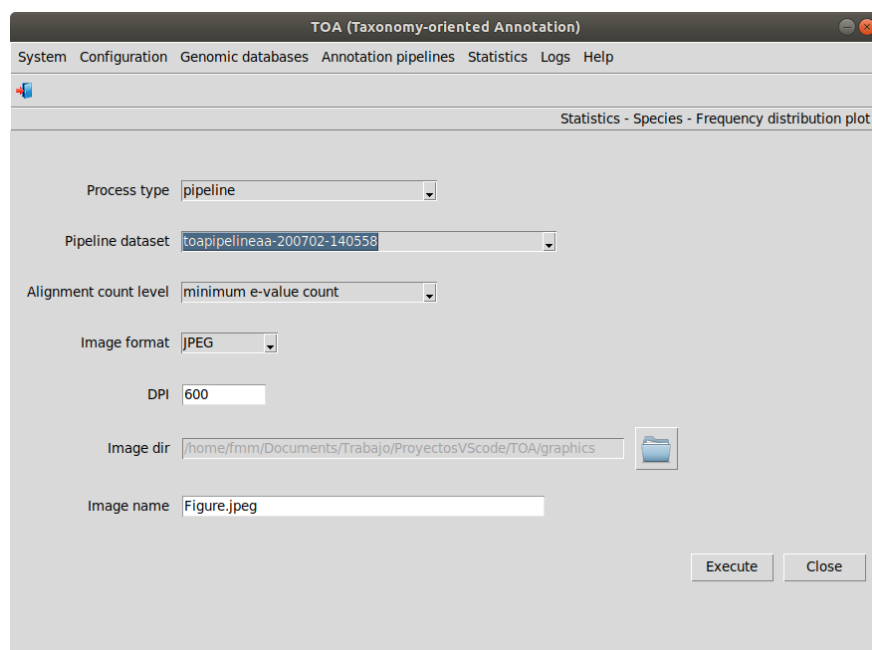Data are listed in a pop-up window (see Figure 22).

Figure. 22. Frequency distribution of annotation datasets generated by the example annotation pipeline.

We view the graphic corresponding to the frequency distribution of annotation datasets selecting:

Main menu > Statistics > Annotation datasets > Frequency distribution plot [Execute]

In the raised window (see Figure 23), we select **TOA-pipeline** in the *Experiment/process* combo-box and the example pipeline in the *Pipeline dataset* combo-box. The default value of the remaining fields (*Image format*, *DPI*, *Image dir* and *Image name*) can be modified if necessary. Then, we press the button *[Execute]*.
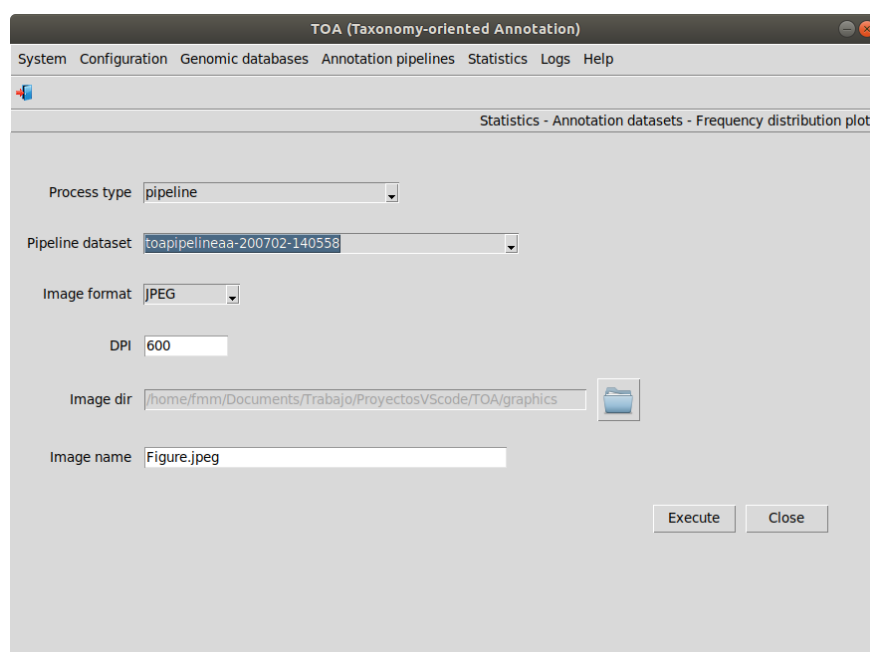
Figure. 23. TOA window corresponding to the menu item *Statistics - Annotation datasets - Frequency distribution plot*.

The plot is saved in the directory selected in *Image dir* and is displayed in a pop-up window (see Figure 24).
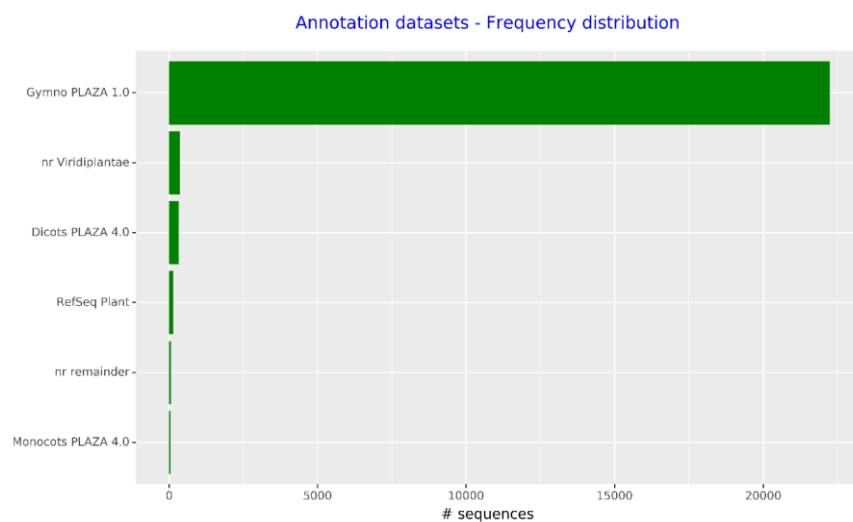
Figure. 24. Plot of frequency distribution of annotation datasets generated by the example annotation pipeline.

Species - Frequency distribution

We consult the frequency distribution of species selecting the menu item with this path:
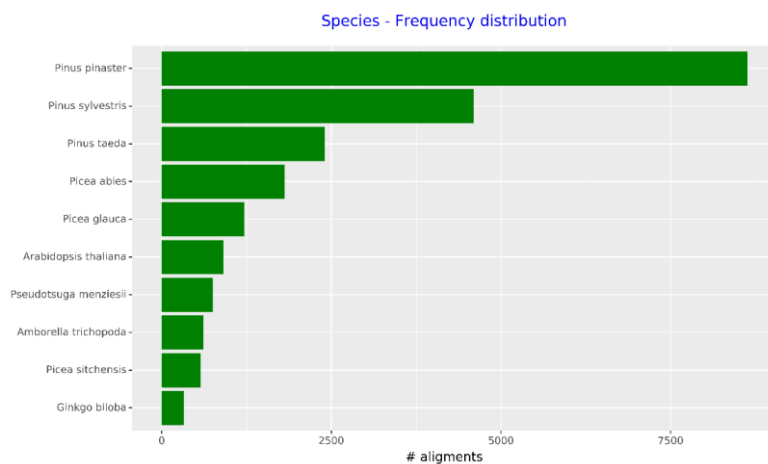
Main menu > Statistics > Species > Frequency distribution data [Execute]

In the raised window (see Figure 25), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. Then, we press the button *[Execute]*.

Figure. 25. TOA window corresponding to the menu item *Statistics - Species - Frequency distribution data*.

Data are listed in a pop-up window (see Figure 26).

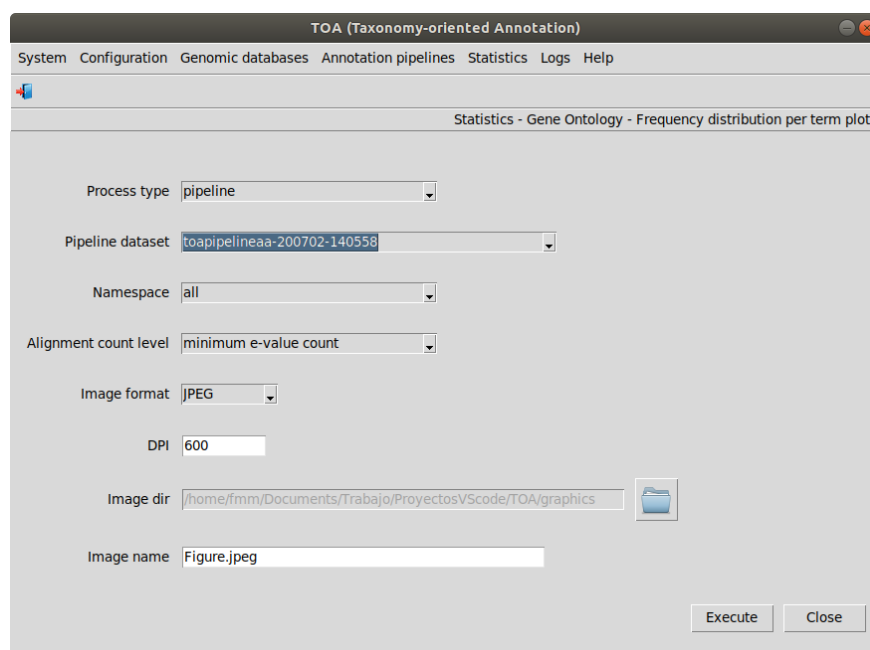**Figure. 26**. Frequency distribution of species generated by the example annotation pipeline.

We view the graphic corresponding to the frequency distribution of species selecting:

Main menu > Statistics > Species > Frequency distribution plot [Execute]

In the raised window (see Figure 27), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. The default value of the remaining fields (*Alignment count level*, *Image format*, *DPI*, *Image dir* and *Image name*) can be modified if necessary. The possible values of *Alignment count level* are **all count** (all alignments are considered), **first HSP count** (alignments with HSP 1 are considered) and **minimum e-value count** (alignments with minimum e-value per sequence are considered). Then we press the button *[Execute].*

Figure. 27. TOA window corresponding to the menu item *Statistics - Species - Frequency distribution plot*.

The plot is saved in the directory selected in *Image dir* and is displayed in a pop-up window (see Figure 28).

Figure. 28. Plot of frequency distribution of species generated by the example annotation pipeline.

Gene Ontology - Frequency distribution per term

We consult the frequency distribution per Gene Ontology term selecting the menu item with this path:

Main menu > Statistics > Gene Ontology > Frequency distribution per term data[Execute]

In the raised window (see Figure 29), we select **TOA-pipeline** in the *Experiment/process* combo-box and the example pipeline in the *Pipeline dataset* combo-box. Then we press the button *[Execute].*
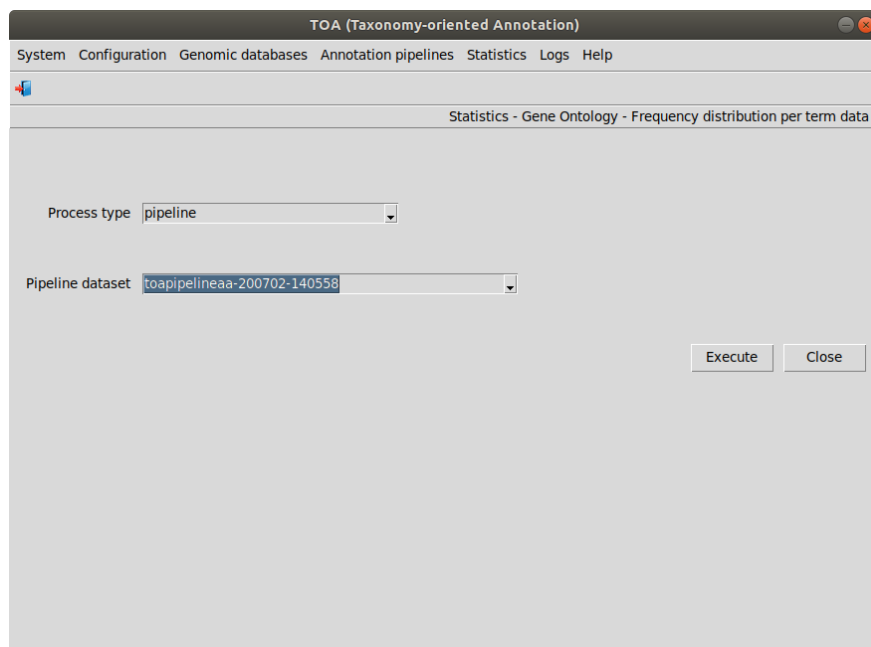
Figure. 29. TOA window corresponding to the menu item *Statistics - Gene Ontology - Frequency distribution per term data*.

Data are listed in a pop-up window (see Figure 30).

Figure. 30. Frequency distribution per Gene Ontology term generated by the example annotation pipeline.

We view the graphic corresponding to the frequency distribution per Gene Ontology term selecting:
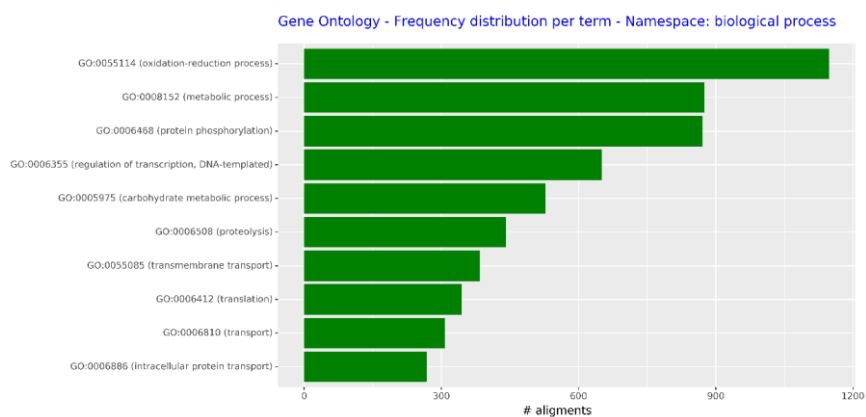
Main menu > Statistics > Gene Ontology > Frequency distribution per term plot [Execute]

In the raised window (see Figure 31), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. The default value of the remaining fields (*Namespace*, *Alignment count level*, *Image format*, *DPI*, *Image dir* and *Image name*) can be modified if necessary. The Namespace combo-box has the values **all**, **biological process**, **cellular component** and **molecular function**. The possible values of *Alignment count level* are **all count** (all alignments are taken into account), **first HSP count** (alignments with HSP 1 are taken into account) and **minimum e-value count** (alignments with minimum e-value per sequence are taken into account). Then we press the button *[Execute]*.

Figure. 31. TOA window corresponding to the menu item *Statistics - Gene Ontology - Frequency distribution per term plot*.

The plot is saved in the directory selected in *Image dir* and is displayed in a pop-up window (see Figure 32).

Figure. 32. Plot of frequency distribution per Gene Ontology term generated by the example annotation pipeline.
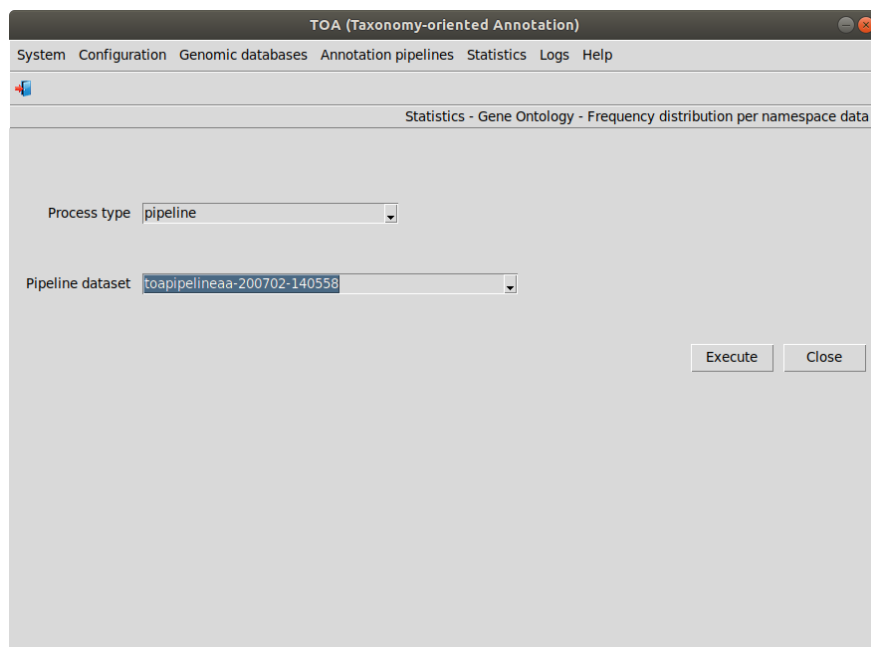
Gene Ontology - Frequency distribution per namespace

We consult the frequency distribution per Gene Ontology namespace selecting the menu item with this path:

Main menu > Statistics > Gene Ontology > Frequency distribution per namespace data[Execute]

In the raised window (see Figure 33), we select **TOA-pipeline** in the *Experiment/process* combo-box and the example pipeline in the *Pipeline dataset* combo-box. Then we press the button *[Execute].*

Figure. 33. TOA window corresponding to the menu item *Statistics - Gene Ontology - Frequency distribution per namespace data*.
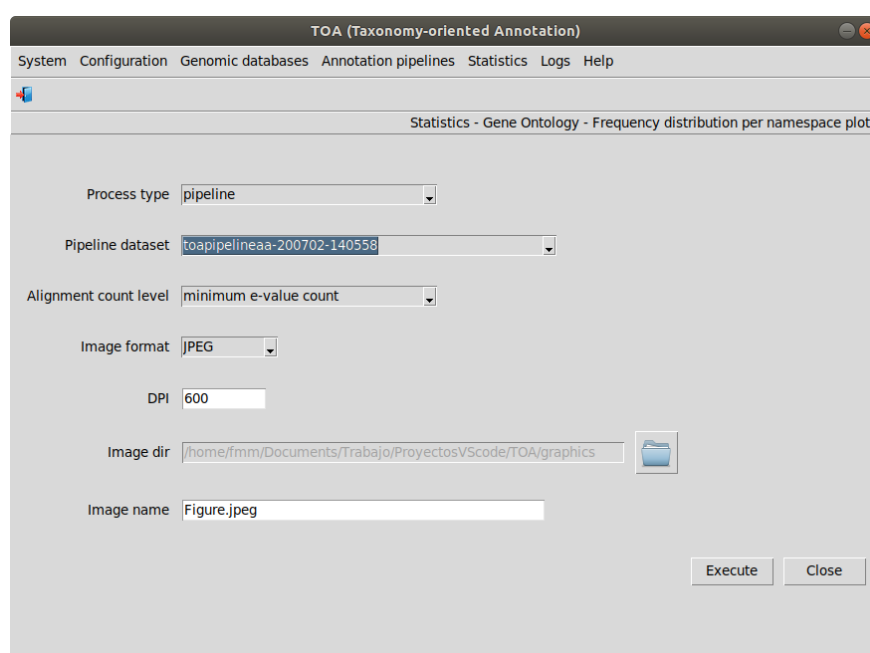
Data are listed in a pop-up window (see Figure 34).

Figure. 34. Frequency distribution per Gene Ontology namespace generated by the example annotation pipeline.

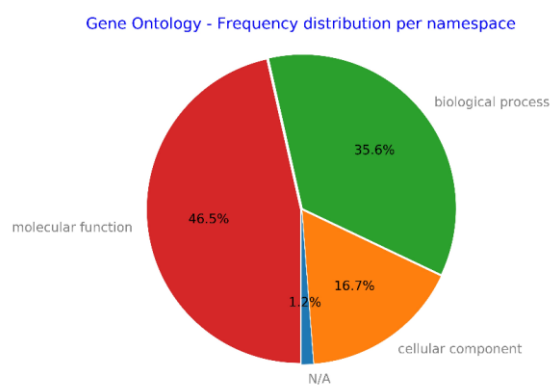We view the graphic corresponding to the frequency distribution per Gene Ontology namespace selecting:

Main menu > Statistics > Gene Ontology > Frequency distribution per namespace plot [Execute]

In the raised window (see Figure 35), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. The default value of the remaining fields (*Alignment count level*, *Image format*, *DPI*, *Image dir* and *Image name*) can be modified if necessary. The possible values of *Alignment count level* are: **all count** (all alignments are taken into account), **first HSP count** (alignments with HSP 1 are taken into account) and **minimum e-value count** (alignments with minimum e-value per sequence are taken into account). Then we press the button *[Execute]*.

Figure. 35. TOA window corresponding to the menu item *Statistics - Gene Ontology - Frequency distribution per namespace plot*.

The plot is saved in the directory selected in *Image dir* and is displayed in a pop-up window (see Figure 36).

Gene Ontology - Frequency distribution per namespace

Figure. 36. Plot of frequency distribution per Gene Ontology namespace generated by the example annotation pipeline.

Gene Ontology - # sequences per #terms

We consult #sequences per # Gene Ontology terms the menu item with this path:

Main menu > Statistics > Species > # sequences per #terms data [Execute]

In the raised window (see Figure 37), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. Then we press the button *[Execute].*
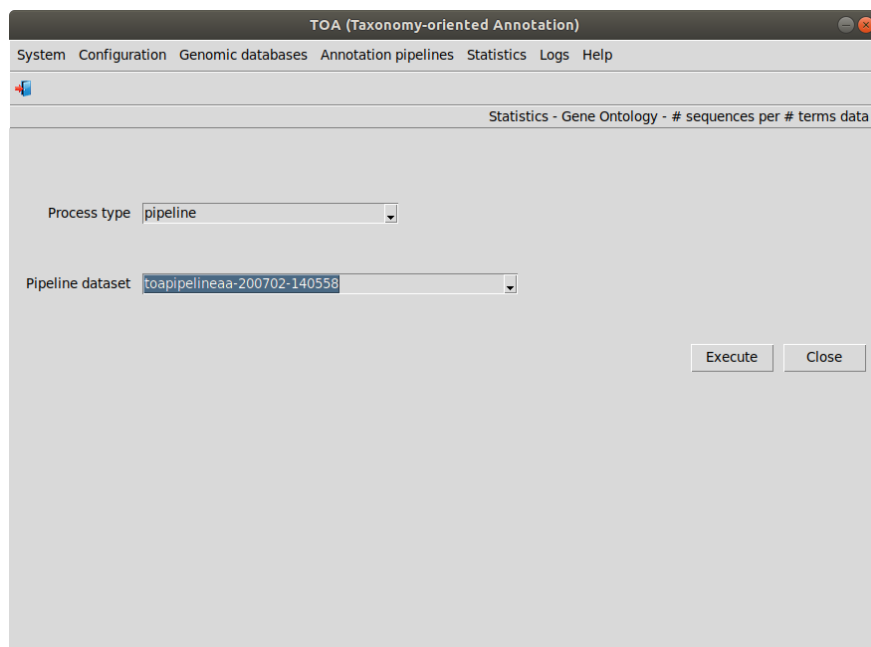
Figure. 37. TOA window corresponding to the menu item *Statistics - Gene Ontology - # sequences per #terms data*.

Data are listed in a pop-up window (see Figure 38).

Figure. 38. #sequences per # Gene Ontology terms generated by the example annotation pipeline.

We view the graphic corresponding to #sequences per # Gene Ontology terms selecting:

Main menu > Statistics > Species > # sequences per #terms plot [Execute]

In the raised window (see Figure 39), we select **pipeline** in the *Process type* combo-box and the example pipeline in the *Pipeline dataset* combo-box. The default value of the remaining fields (*Image format*, *DPI*, *Image dir* and *Image name*) can be modified if necessary. Then we press the button *[Execute]*.
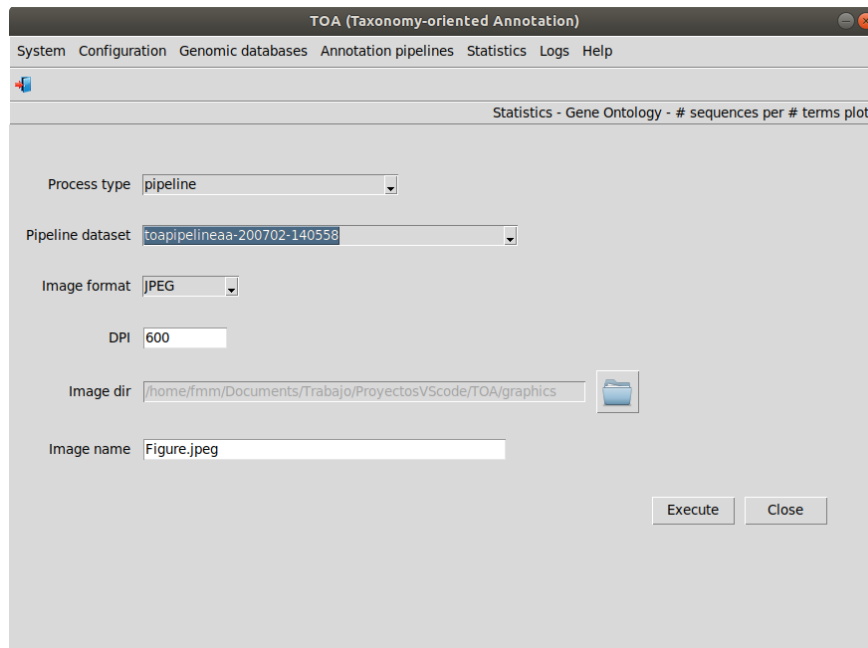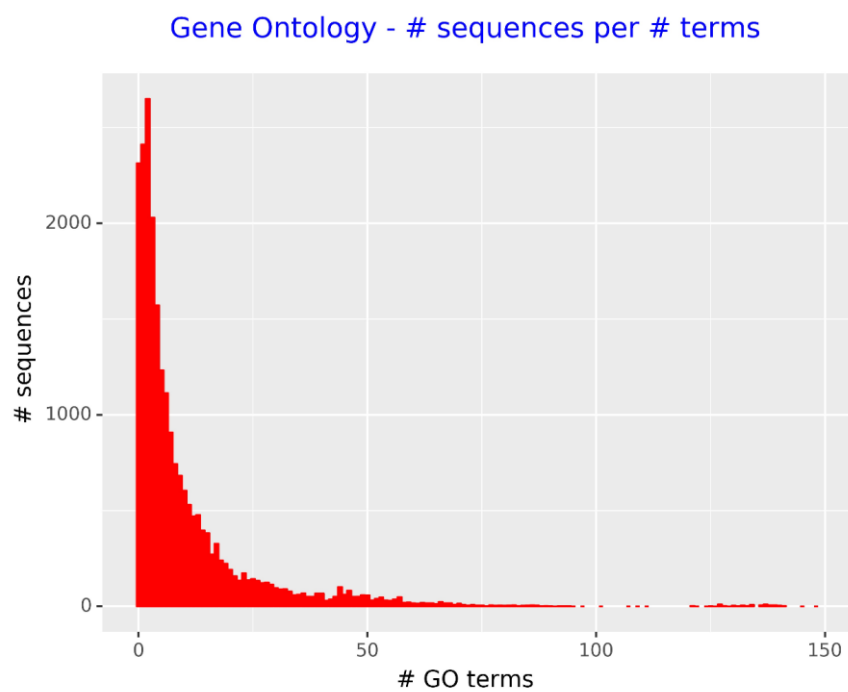

Figure. 39. TOA window corresponding to the menu item *Statistics - Gene Ontology - # sequences per #terms plot*.

The plot is saved in the directory selected in *Image dir* and is displayed in a pop-up window (see Figure 40).

Figure. 40. Plot of #sequences per # Gene Ontology terms generated by the example annotation pipeline.

How to cite

If you are using TOA, you should cite the following paper:

Mora-Márquez, F., Chano, V., Vázquez-Poletti, J. L., & López de Heredia, U. (2021). TOA: A software package for automated functional annotation in non-model plant species. *Molecular Ecology Resources*, *21*(2), 621–636. DOI: 10.1111/1755-0998.13285