GYMNOTOA-APP

(Gymnosperms Taxonomy-oriented Annotation)

v0.20

A software package for automated functional annotation in Gymnosperms

GI en Desarrollo de Especies y Comunidades Leñosas (WooSp)
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 GYMNOTOA-APP (Gymnosperms Taxonomy-oriented Annotation) is available for free download from the GitHub repository:

https://github.com/GGFHF/gymnoTOA-app

under GNU General Public License v3.0.

Introduction

Functional annotation is the task of bioinformatics analysis that aims to determine the biochemical and biological functions of nucleotide sequences obtained in the assembly of massive high-throughput sequencing experiments. A common way to perform this annotation is to perform a search for homologous sequences and access related functional information deposited in genomic databases.

In order to structure functional information, several annotation ontology systems have been developed located in accessible databases that contain formally defined, normalized, and consistent words and identifiers of classes and relationships that represent the biochemical and biological phenomena of a domain. Among these systems are:

- Gene Ontology (GO): It is a database of gene functions, called GO terms, which have an associated unique alphanumeric identifier, name, and definition. They describe gene products through three separate domains (namespaces): molecular function (the elementary activities of a gene product at the molecular level), biological process (operations or sets of molecular events with a defined beginning and end relevant to the functioning of cells, tissues, organs, and organisms), and cellular component (the parts of the cell or its extracellular environment).
- Kyoto Encyclopedia of Genes and Genomes (KEGG): It stores molecular functions represented in terms of functional orthologs of genes and proteins. KEGG modules and pathway maps represent higher level functions through networks of molecular interactions, reactions and relationships.
- MetaCyc Metabolic Pathway Database: It contains pathways involved in both primary and secondary metabolism, as well as associated metabolites, reactions, enzymes, and genes.
- EC (Enzyme Commission) number: It is a system used to classify enzymes according to the reaction they catalyze. Each identifier consists of four numbers: the first number defines the highest level of classification, and the following numbers indicate increasingly specific sub-classifications.

In the functional enrichment analysis, functional annotations that are significantly over-expressed and under-expressed in experiments are detected using statistical methods.

Gymnosperms are a non-flowering seed plant clade of about 1,000 living species. Most gymnosperms are long-lived woody plant species of great economic and ecological importance widely distributed around the globe. However, their large genome sizes and lack of genomic resources if compared to other plant model species, limit the initiatives to address biological questions and the functional evolution of gymnosperms. Obtaining high quality genomes and annotations of gymnosperms remains a major challenge because specific gene functional annotation information is scattered, incomplete or with low curation level. It is very frequent to utilize genomic resources from other land plants, but this approach may not be adequate, due to the far evolutionary distance between the major lineages, especially between angiosperms and gymnosperms.

GYMNOTOA-APP is a taxonomy-aware functional annotation tool and is intended to be a public reference database to help to produce accurate functional annotation reports in genomics and transcriptomics experiments on gymnosperm species.

The whole gymnoTOA framework is designed in two separate parts with different functions:

- GYMNOTOA-DB is a database that includes curated records of gymnosperm proteins and related taxonomic, functional and structural information.
- GYMNOTOA-APP is a desktop application that allows us to perform the functional annotation and enrichment analysis of transcriptomes yielded in gymnosperms experiments exploiting the cross-references uploaded in GYMNOTOA-DB.

The inputs to build GYMNOTOA-DB (Figure 1) are three sources of sequences:

- the NCBI protein database (https://www.ncbi.nlm.nih.gov/protein)
- TAIR10 database (https://www.arabidopsis.org/)
- CANTATA lncRNA database (http://cantata.amu.edu.pl/).

First, all amino acid sequences of gymnosperms available in the NCBI protein database are loaded by the utility *esearch* of **Entrez Direct** software (https://www.ncbi.nlm.nih.gov/books/NBK179288/). Redundant terms are removed using **MMseqs2** (https://github.com/soedinglab/MMseqs2) to produce clusters of sequences that are aligned with **MAFFT** (https://mafft.cbrc.jp/alignment/software/) to produce a consensus clustered sequence with **EMBOSS** (http://emboss.open-bio.org/). The overall quality of the set consensus sequences performed by **BUSCO** (https://busco.ezlab.org/).

Then, annotation records corresponding to the consensus sequences are searched using **InterProScan** and **eggnog-mapper**. Both consensus sequences and associated records are uploaded into an SQLite database (see appendix A) with **load-interproscan-annotations.py** and **load-interproscan-annotations.py** of **NGShelper** software package (https://github.com/GGFHF/NGShelper). BLAST+ and DIAMOND databases are built using **blastp** and **diamond blastp** respectively.

TAIR10 protein sequences (downloaded from TAIR10 database) are processed by the program **makeblastdb** to get a BLAST+database used by the program **blastx**, to find the *Arabidopsis thaliana* orthologs of consensus sequences, loaded into the SQLite database by the program **load-tair10-orthologs.py** of **NGShelper**.

On the other side, lncRNA sequences (downloaded from CANTATA lncRNA database) are processed by **makeblastdb** to build a BLAST+ database.

The SQLite database, BLAST+ and DIAMOND consensus databases, and BLAST+ lncRNA database (shapes filled in yellow color in Figure 1) constitute the gymnoTOA database: GYMNOTOA-DB. It is hosted at the CESVIMA server of the Universidad Politécnica de Madrid and can be downloaded directly (https://blogs.upm.es/gymnoTOA-db/gymnoTOA-db/) or (preferably) using GYMNOTOA-APP (https://github.com/GGFHF/gymnoTOA-app)

GYMNOTOA-DB is periodically updated by the development team, and statistics of the database are provided at the web page.

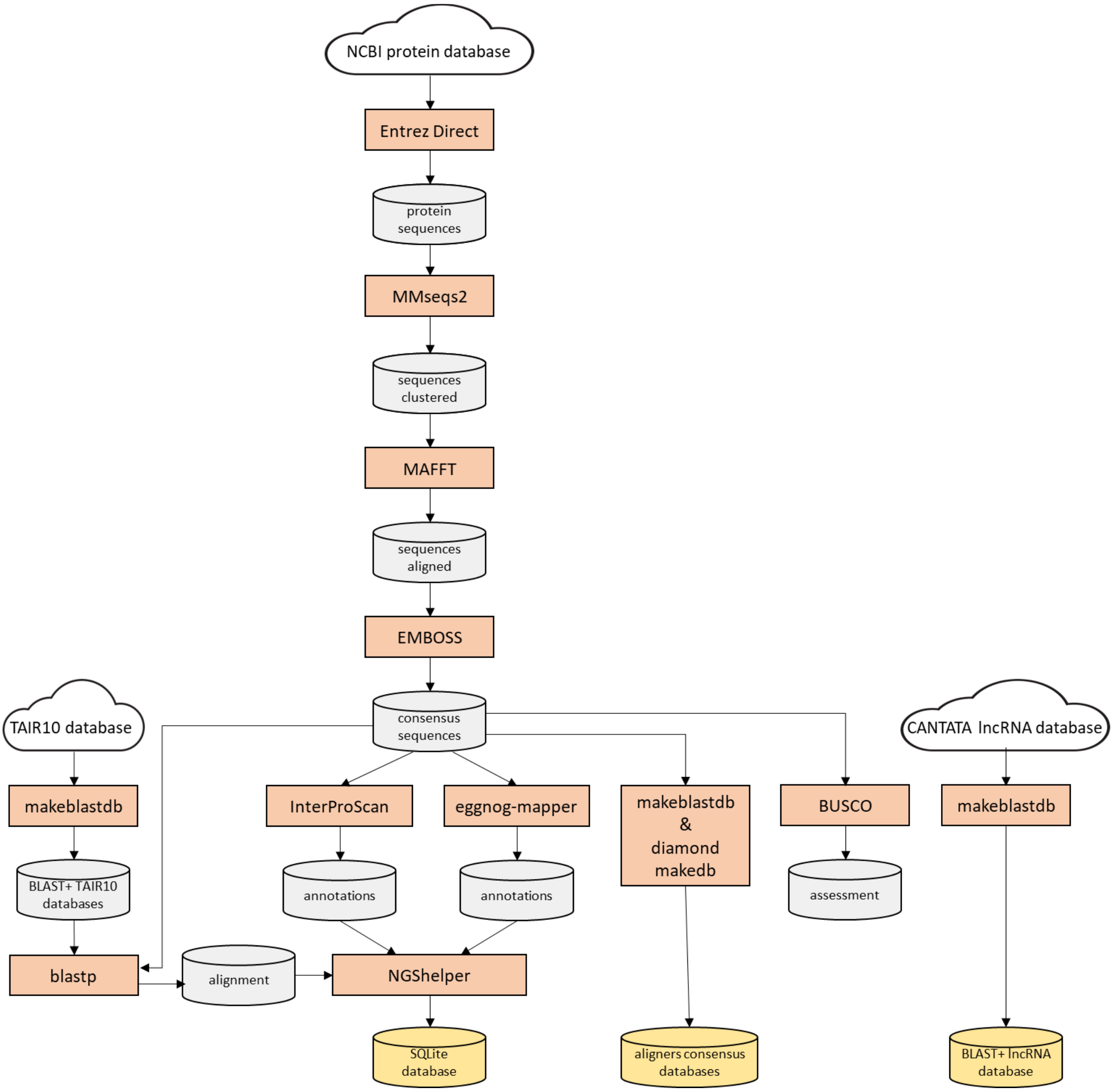


Figure 1. Flow-chart of the creation of the GYMNOTOA-DB.

Once GYMNOTOA-DB is downloaded by GYMNOTOA-APP, transcripts can be annotated (Figure 2). The transcript sequences of a genomics or transcriptomics experiment are processed by **CodAn** (https://github.com/pedronachtigall/CodAn) in order to get predict ORFs. These sequences are aligned to BLAST+ consensus database by **blastp** if BLAST+ aligner is used or **diamond blastp** for alignments performed with DIAMOND. Then, transcripts sequences are aligned to BLAST+ or DIAMOND consensus database by **blastx** or **diamond blastx**, respectively. And finally, transcript sequences are aligned to BLAST+ lncRNA datatabase using **blastn**. All alignments are concatenated to their functional annotations by the program **concat-functional-annotations.py** included in GYMNOTOA-APP. This program yields two annotations files: a file containing all annotations yielded by blast programs for every transcript sequence, and the other one contains the annotations of the subject sequence identification with best hit yielded by blast programs for every transcript sequence.

For a transcript sequence, the **blastp/diamond blastp** annotations are chosen. If these annotations do not exist, the **blastx/diamond blastx** annotations are taken. And finally, if they do not exist either, the blastn annotation is selected.

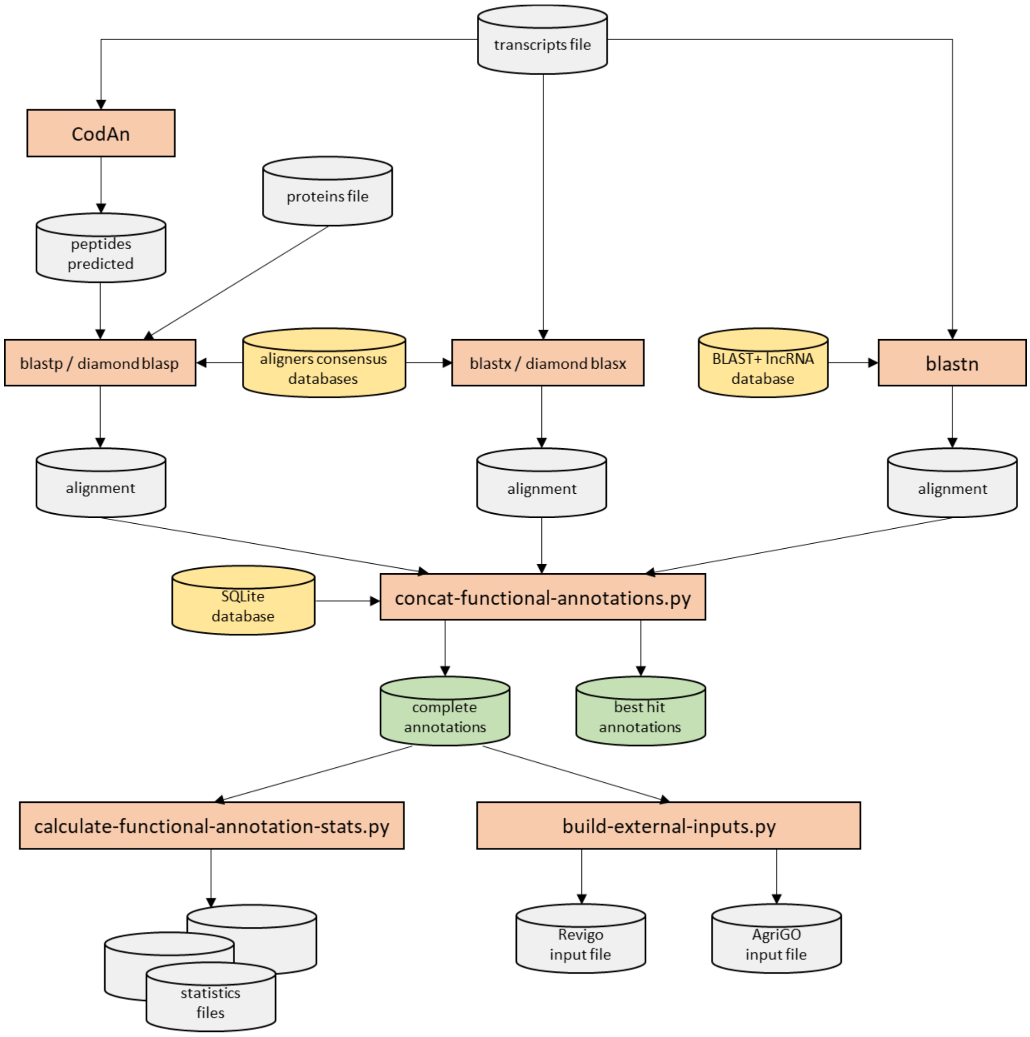


Figure 2. Flow-chart of the process of functional annotation in the GYMNOTOA-APP desktop application. The input file can be a transcripts file or a file with protein sequences.

The input file of a functional annotation process can also be a file with protein sequences (Figure 2). In this case the corresponding steps with **blastx**/**diamond blastx** and **blastn** are not run.

The annotations files contain the following data:

- qseqid: query (transcriptome) sequence identification
- sseqid: subject (BLAST+ databases of GYMNOTOA-DB) sequence identification
- pident: percentage of identical positions
- length: alignment length
- mismatch: number of mismatches
- gapopen: number of gap openings
- qstart: start of alignment in query
- qend: end of alignment in query
- sstart: start of alignment in subject
- send: end of alignment in subject
- evalue: expect value
- bitscore: bit score
- algorithm: alignment algorithm that yielded the annotation (blastp, blastx or blastn)
- ncbi_description: description from the NCBI protein sequence
- ncbi_species: species from the NCBI protein sequence
- tair10_ortholog_seq_id: ortholog sequence identification from TAR10
- interpro_goterms: concatenated list of GO terms from InterPro
- panther_goterms: concatenated list of GO terms from Panther
- metacyc_pathways: concatenated list of pathway identifications from MetaCyc
- eggnog_ortholog_seq_id: ortholog sequence identification from eggNOG
- eggnog_ortholog_species: species from eggNOG
- eggnog_ogs: OGs (Orthologous Groups) of proteins from eggNOG
- cog_category: COG (Cluster of Orthologous Genes) from eggNOG
- eggnog_description: description from eggNOG
- eggnog_goterms: concatenated list of GO terms from eggNOG
- ec: concatenated list of EC (Enzyme Commission) numbers
- kegg_kos: concatenated list of KO from KEGG
- kegg_pathways: concatenated list of pathway identifications from KEGG
- kegg_modules: concatenated list of module identifications from KEGG
- kegg_reactions: concatenated list of chemical reactions identifications from KEGG
- kegg_rclasses: concatenated list of reactions classification identifications from KEGG
- brite: functional hierarchy of OGs assigned to the sequence
- kegg_tc: T cell receptor (TCR) signaling pathway
- cazy: concatenated list of Carbohydrate-Active Enzymes (CAZymes)
- pfams: concatenated list of protein families from Pfam

The program **calculate-functional-stats.py** yields functional annotations statistics and **build-external-inputs.py** buids specific inputs for AgriGO and REVIGO servers.

From the annotation files, both complete annotations as the best hit annotations, GYMNOTOA-APP can run an enrichment analysis using as background the registries from GYMNOTOA-DB (Figure 3).

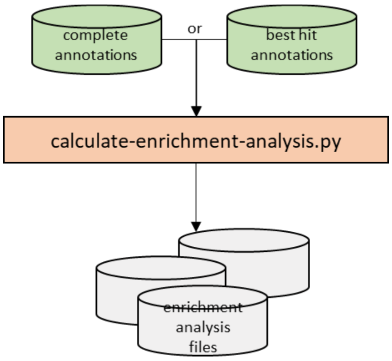


Figure 3. Flow-chart of the process of enrichement analysis in the GYMNOTOA-APP desktop application.

The program **calculate-enrichment-analysis.py** included in GYMNOTOA-APP software package yields files with enrichment analysis of GO terms, MetaCyc pathways, KEGG KOs, and KEGG pathways.

These files contain the following data:

- Identification: GO term, MetaCyc pathway, KEGG KO or KEGG pathway
- Description (only for GO terms)
- Namespace (only for GO terms)
- Sequence number with the identification in transcript annotations
- Sequence number with identifications in transcript annotations
- Sequence number with the identification in species or gymnosperms annotations
- Sequence number with identifications in species or gymnosperms annotations
- Enrichment
- p-value
- FDR (False Discovery Rate)

Programs **makeblastdb, blastn**, **blastx** and **blastp** are included in BLAST+, a NCBI suite of command-line tools (https://blast.ncbi.nlm.nih.gov/doc/blast-help/downloadblastdata.html).

GYMNOTOA-APP is a desktop application designed with a user-friendly interface where researchers can perform functional annotation and enrichment analysis processes with just a few clicks. Nevertheless, GYMNOTOA-APP functional annotation and enrichment analysis processes can also be run on Linus servers using the Bash scripts included in this software (see the chapter “Functional annotation and enrichment analysis on Linux servers”).

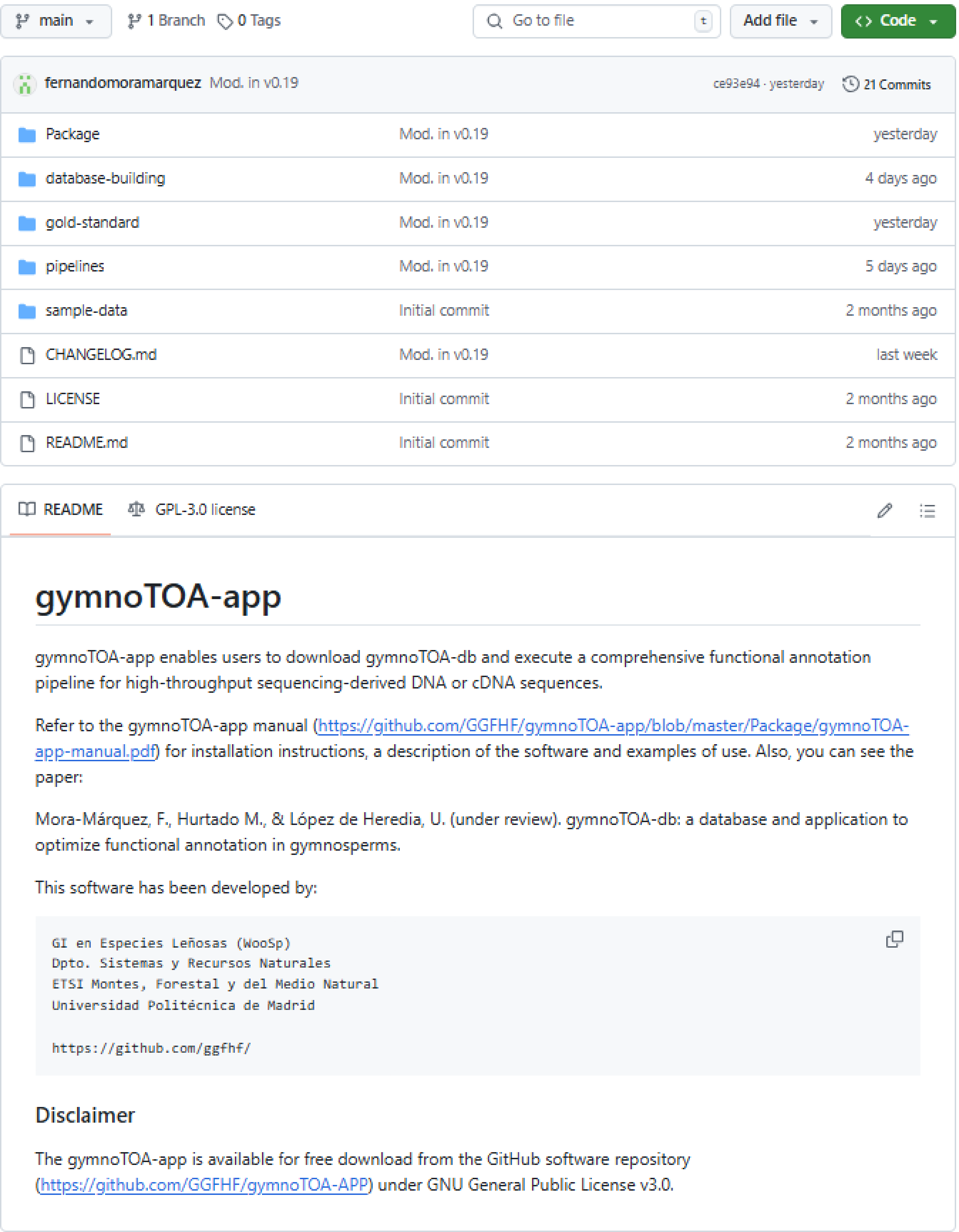
For further information refer to the manuscript:

Fernando Mora-Márquez, Mikel Hurtado & Unai López de Heredia (under review). GYMNOTOA-DB: a database and application to optimize functional annotation in gymnosperms. DOI: https://doi.org/x

Installation in GUI environments

GYMNOTOA-APP installation

GYMNOTOA-APP was programmed in Python 3 and it generates dynamic Bash scripts to perform annotation pipelines. gymnoTOA runs on any computer with Linux or macOS. This software is available from the GitHub repository https://github.com/GGFHF/gymnoTOA-app/ (Figure 4), and it is distributed under GNU General Public License Version 3.



fernandomoraramarquez Mod. in v0.19 ce93e94 · yesterday 21 Commits

Package	Mod. in v0.19	yesterday
database-building	Mod. in v0.19	4 days ago
gold-standard	Mod. in v0.19	yesterday
pipelines	Mod. in v0.19	5 days ago
sample-data	Initial commit	2 months ago
CHANGELOG.md	Mod. in v0.19	last week
LICENSE	Initial commit	2 months ago
README.md	Initial commit	2 months ago

gymnoTOA-app

gymnoTOA-app enables users to download gymnoTOA-db and execute a comprehensive functional annotation pipeline for high-throughput sequencing-derived DNA or cDNA sequences.

Refer to the gymnoTOA-app manual (https://github.com/GGFHF/gymnoTOA-app/blob/master/Package/gymnoTOA-app-manual.pdf) for installation instructions, a description of the software and examples of use. Also, you can see the paper:

Mora-Márquez, F., Hurtado M., & López de Heredia, U. (under review). gymnoTOA-db: a database and application to optimize functional annotation in gymnosperms.

This software has been developed by:

```
GI en Especies Leñosas (WooSp)
Dpto. Sistemas y Recursos Naturales
ETSI Montes, Forestal y del Medio Natural
Universidad Politécnica de Madrid

https://github.com/ggfhf/
```

Disclaimer

The gymnoTOA-app is available for free download from the GitHub software repository (https://github.com/GGFHF/gymnoTOA-APP) under GNU General Public License v3.0.

Figure 4. GYMNOTOA-APP home at GitHub software repository.

In this manual it is clearly delineated using red typeface those file paths or environment names requiring user modification if necessary.

If the OS of your computer is Linux or macOS, we will download the GYMNOTOA-APP software in the directory, e.g., $HOME/Apps/gymnoTOA-app, and decompress the ZIP file using the following command:

```
$ mkdir -p $HOME/Apps

$ cd $HOME/Apps

$ wget --output-document main.zip https://github.com/GGFHF/gymnoTOA-
app/archive/refs/heads/main.zip

$ unzip main.zip

$ mv gymnoTOA-app-main gymnoTOA-app

$ rm main.zip
```

We will have the directory gymnoTOA-app in $HOME/Apps. Then, the execution permissions of the programs must be set by using these commands:

```
$ find $HOME/Apps/gymnoTOA-app -type f \( -name "*.py" -o -name "*.sh" \)
-exec chmod +x {} \;
```

To install GYMNOTOA-APP in a computer with Windows, go to the GitHub repository https://github.com/GGFHF/gymnoTOA-app/ (Figure 4), click *Code* and in the pup-up window click in *Download ZIP*. Decompress the downloaded ZIP file using “Extract All..” of the File Explorer on gymnoTOA-app-master.zip extracting the files in the directory, e.g., %USERPROFILE%\Apps\gymnoTOA-app.

In this manual we will refer to the GYMNOTOA-APP directory as $HOME/Apps/gymnoTOA-app (Linux or macOS) or %USERPROFILE%\Apps\gymnoTOA-app, but you can install GYMNOTOA-APP in the directory you think is appropriate.

Conda and additional infrastructure software installation

Python 3 (https://www.python.org/), version 3.12 or higher, is required by GYMNOTOA-APP. Also this application needs the following Python modules:

- PyQt5 (https://www.riverbankcomputing.com/static/Docs/PyQt5/), a Python interface for QT software package.
- Pandas (https://pandas.pydata.org/), a Python library for data analysis and manipulation tool

- Matplotlib (https://matplotlib.org/), a software for creating static, animated, and interactive visualizations in Python.
- NumPy (https://numpy.org/), a Python library that supports for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- SciPy (https://scipy.org/), a Python library used for scientific computing and technical computing.
- Plotnine (https://plotnine.readthedocs.io/en/stable/), an implementation of a grammar of graphics in Python based on ggplot2

Conda is an open-source package management and environment management tool that allows you to install, manage, and configure software packages and dependencies, particularly for Python and data science projects. We will use Miniforge3 (https://github.com/conda-forge/miniforge) that is a minimal installer for Conda that uses the conda-forge channel by default (https://conda-forge.org/), but you could use other Conda installer.

Next, we present examples of how to install Python, Conda software and this additional software in Linux Ubuntu, macOS and Windows using WSL.

Installation on Ubuntu 22.04 LTS or Ubuntu 24.04 LTS

The following are the commands for the installation of Miniforge3 in the directory, e.g. $HOME/Apps/Miniforge3, if there is no previous installation:

```
$ mkdir -p $HOME/Apps

$ cd $HOME/Apps

$ wget "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-$(uname)-$(uname -m).sh"

$ bash Miniforge3-$(uname)-$(uname -m).sh -b -p $HOME/Apps/Miniforge3

$ rm Miniforge3-$(uname)-$(uname -m).sh

$ $HOME/Apps/Miniforge3/condabin/conda init bash
```

Close the terminal and open a new terminal. The Conda environment base is now active. Then type the following commands to configure the download environment:

```
$ conda config --add channels bioconda

$ conda config --add channels conda-forge

$ conda config --set channel_priority strict
```

Finally, we will create the gymnotoa environment used to run gymnoTOA-app programs typing the following command:

```
$ conda env create -f $HOME/Apps/gymnoTOA-app/yml/gymnotoa.yml
```

Installation on macOS 15.0.1

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

```
$ /bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

$ brew install wget
```

Now install the Miniforge3 in the directory, e.g. $HOME/Apps/Miniforge3, if there is no previous installation:

```
$ mkdir -p $HOME/Apps

$ cd $HOME/Apps

$ wget "https://github.com/conda-
forge/miniforge/releases/latest/download/Miniforge3-$(uname)-$(uname -
m).sh"

$ bash Miniforge3-$(uname)-$(uname -m).sh -b -p $HOME/Apps/Miniforge3

$ rm Miniforge3-$(uname)-$(uname -m).sh

$ $HOME/Apps/Miniforge3/condabin/conda init zsh
```

Then close the terminal and open a new terminal and type the following commands to configure the download environment:

```
$ conda config --add channels bioconda

$ conda config --add channels conda-forge

$ conda config --set channel_priority strict
```

Finally, we will create the gymnotoa environment used to run gymnoTOA-app programs typing the following command:

```
$ conda env create -f $HOME/Apps/gymnoTOA-app/yml/gymnotoa.yml
```

Installation on Microsoft Windows 10 (64 bits) and Windows 11 using WSL

GYMNOTOA-APP uses the Windows Subsystem for Linux (WSL) and Ubuntu to run some scripts coded in Bash. WSL has to be installed before using GYMNOTOA-APP. For further clarification about WSL, you can see the URL https://learn.microsoft.com/en-us/windows/wsl. In order to install WSL and Ubuntu 24.04, open a command prompt as administrator and type the following command:

```
> wsl --install --distribution Ubuntu-24.04
```

Restart Windows. Then a window will appear installing Ubuntu. This process may take a few minutes. You will have to enter a username and its password. When the process ends, close this window.

In order to install Miniforge3, e.g. in the user root directory (%USERPROFILE%), open a command prompt and type the following commands:

```
> cd %USERPROFILE%

> curl -O “https://github.com/conda-
forge/miniforge/releases/latest/download/Miniforge3-Windows-
x86_64.exe”

> Miniforge3-Windows-x86_64.exe /InstallationType=JustMe /AddToPath=1
/RegisterPython=1 /S /D=%USERPROFILE%\Miniforge3

> del Miniforge3-Windows-x86_64.exe
```

Then close the terminal and open a new terminal and type the following commands to configure the download environment:

```
> conda config --add channels conda-forge

> conda config --set channel_priority strict
```

Finally, we will create the gymnotoa environment used to run gymnoTOA-app programs typing the following command:

```
$ conda env create -f %USERPROFILE%\Apps\gymnoTOA-app/yml/gymnotoa-
win.yml
```

Starting GYMNOTOA-APP

GYMNOTOA-APP runs in graphical mode using the graphical user interface (GUI).

If your O.S. is Ubuntu or macOS, start GYMNOTOA-APP typing the following commands in a terminal window:

```
$ cd $HOME/Apps/gymnoTOA-app/Package

$ conda activate gymnotoa

$ ./gymnoTOA.py
```

And if your O.S. is Windows, type the command:

```
> cd %USERPROFILE%\Apps\gymnoTOA-app\Package

> conda activate gymnotoa

> python gymnoTOA.py
```

The initial appearance of GYMNOTOA-APP at application startup in GUI mode is shown in Figure 5.

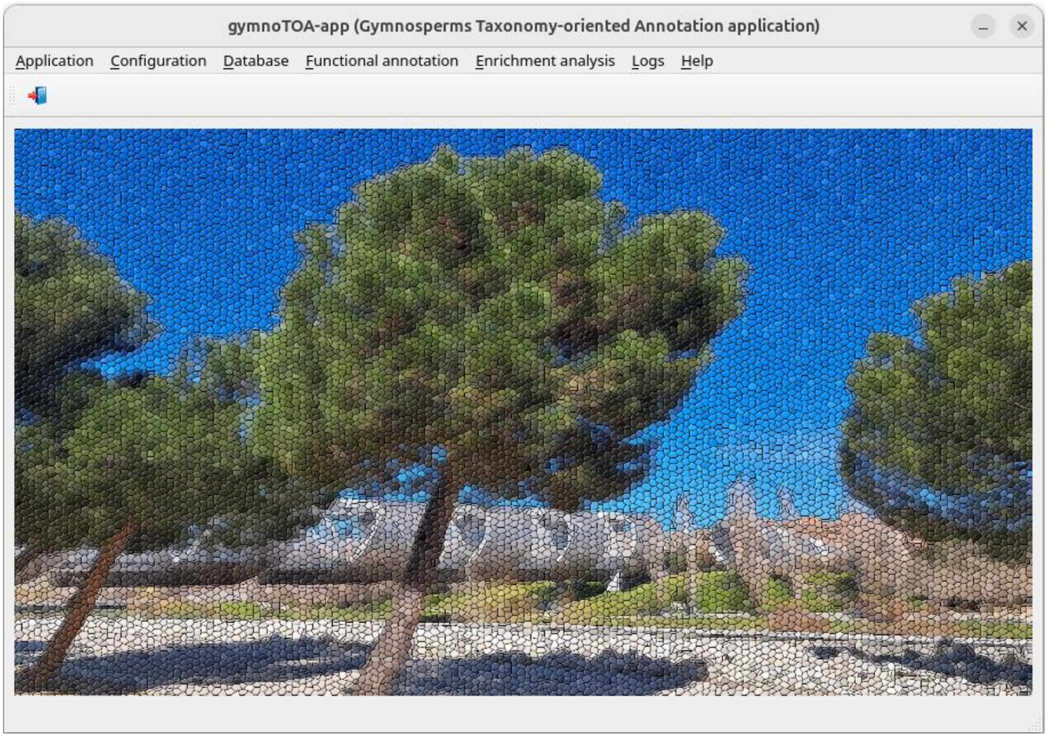


Figure 5. Front-end of the GYMNOTOA-APP application interface at startup.

First steps in GUI environments

GYMNOTOA-APP menus

GYMNOTOA-APP is structured in several menus:

Application

Just to exit the application.

Configuration

This menu contains all the items related to:

- Recreate gymnoTOA-app config file
- View gymnoTOA-app config file
- Install Bioinfo software
    - Miniforge3 and infrastructure package in WSL (only on Windows)
    - BLAST+
    - CodAn
    - DIAMOND

Database

This menu contains all the items related to:

- Download gymnoTOA-db
- Statistics

Functional annotation

This menu contains all the items related to:

- Run pipeline
- Restart pipeline
- Browse results
- Statistics
    - Summary report
    - Species
        - Frequency distribution data
        - Frequency distribution plot
    - Gene Ontology
        - Frequency distribution per GO term data
        - Frequency distribution per GO term plot
        - Frequency distribution per namespace data

        - Frequency distribution per namespace plot
        - Sequences # per GO terms # data
        - Sequences # per GO terms # plot

Enrichment analysis

This menu contains all the items related to:

- Run analysis
- Restart analysis
- Browse results
    - GO enrichment analysis
    - Metacyc pathway enrichment analysis
    - KEGG KO enrichment analysis
    - KEGG pathway enrichment analysis

Logs menu

This menu allows the access to the application logs:

- Submitting logs
- Result logs

Help menu

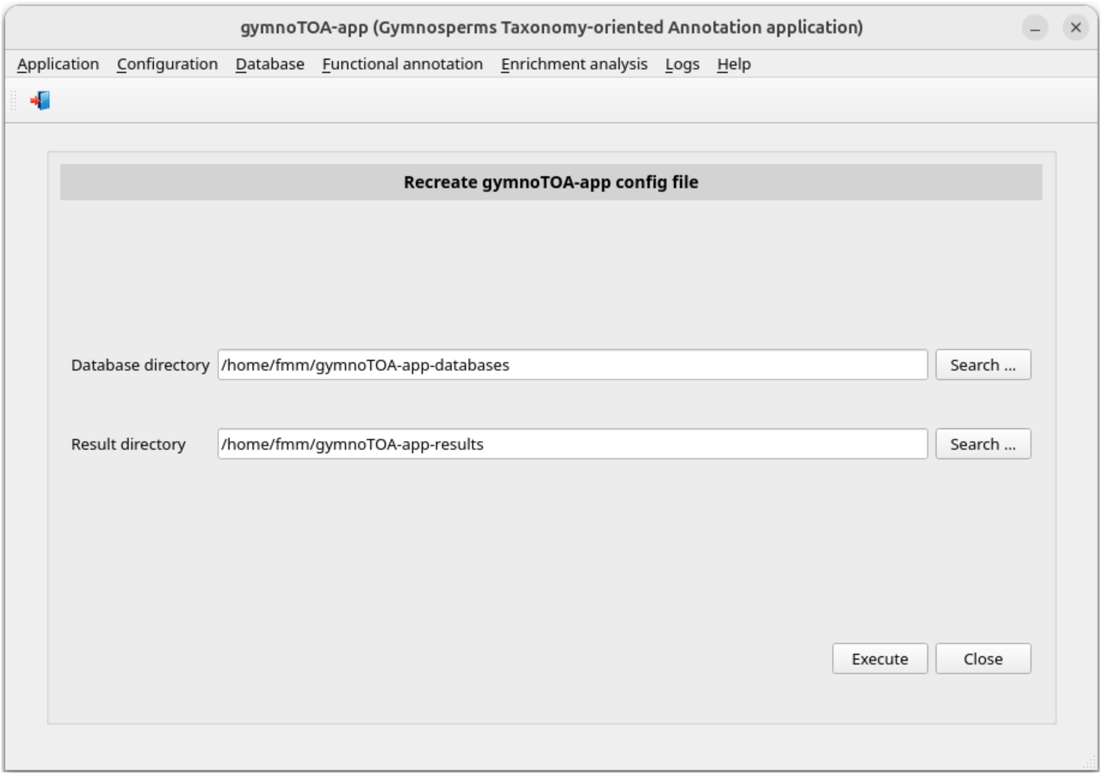
It contains the documentation of the application.

Configuring the GYMNOTOA-APP environment

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

> *Main menu > Configuration > Recreate gymnoTOA-app config file*

Figure 6 shows the window corresponding to this menu item. Default values are presented for *Database directory* and *Result directory*. If necessary, modify them and press the button *[Execute]*.



Database directory: /home/fmm/gymnoTOA-app-databases

Result directory: /home/fmm/gymnoTOA-app-results

Figure 6. Window *Recreate gymnoTOA-app config file*.

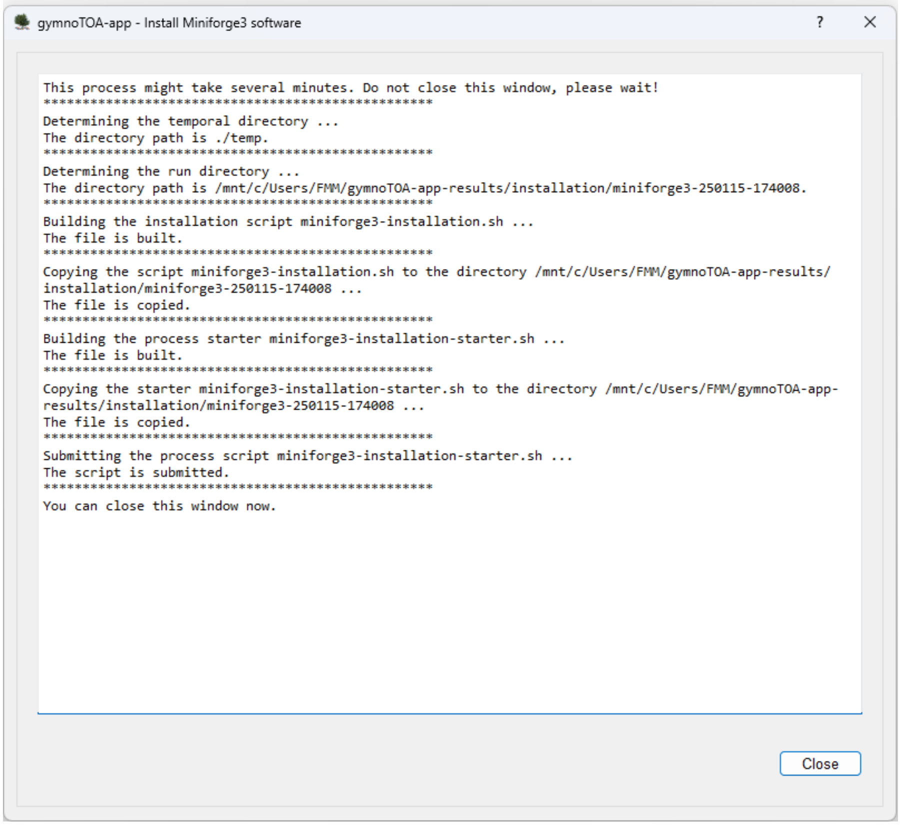
Installing Miniforge3 and infrastructure packages in WSL (only on Windows)

As stated in chapter “Installation in GUI environments”, GYMNOTOA-APP uses the Windows Subsystem for Linux (WSL) and Ubuntu to run some scripts coded in Bash. Therefore, it is necessary to install Miniforge3 and the infrastructure packages in WSL.

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

> *Main menu > Configuration > Bioinfo software installation > Miniforge3 and infrastructure packahes in WSL (only on Windows) [Execute]*

Press the bottom *[Execute]*. A pop-up window will display the submission log (Figure 7).



```
This process might take several minutes. Do not close this window, please wait!
**************************************************
Determining the temporal directory ...
The directory path is ./temp.
**************************************************
Determining the run directory ...
The directory path is /mnt/c/Users/FMM/gymnoTOA-app-results/installation/miniforge3-250115-174008.
**************************************************
Building the installation script miniforge3-installation.sh ...
The file is built.
**************************************************
Copying the script miniforge3-installation.sh to the directory /mnt/c/Users/FMM/gymnoTOA-app-results/
installation/miniforge3-250115-174008 ...
The file is copied.
**************************************************
Building the process starter miniforge3-installation-starter.sh ...
The file is built.
**************************************************
Copying the starter miniforge3-installation-starter.sh to the directory /mnt/c/Users/FMM/gymnoTOA-app-
results/installation/miniforge3-250115-174008 ...
The file is copied.
**************************************************
Submitting the process script miniforge3-installation-starter.sh ...
The script is submitted.
**************************************************
You can close this window now.
```

Figure 7. Submitting log of a Miniforge3 installation process on a computer with Windows.

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

Main menu > Logs > Result logs

Figure 8 shows the window *Browse results logs*.

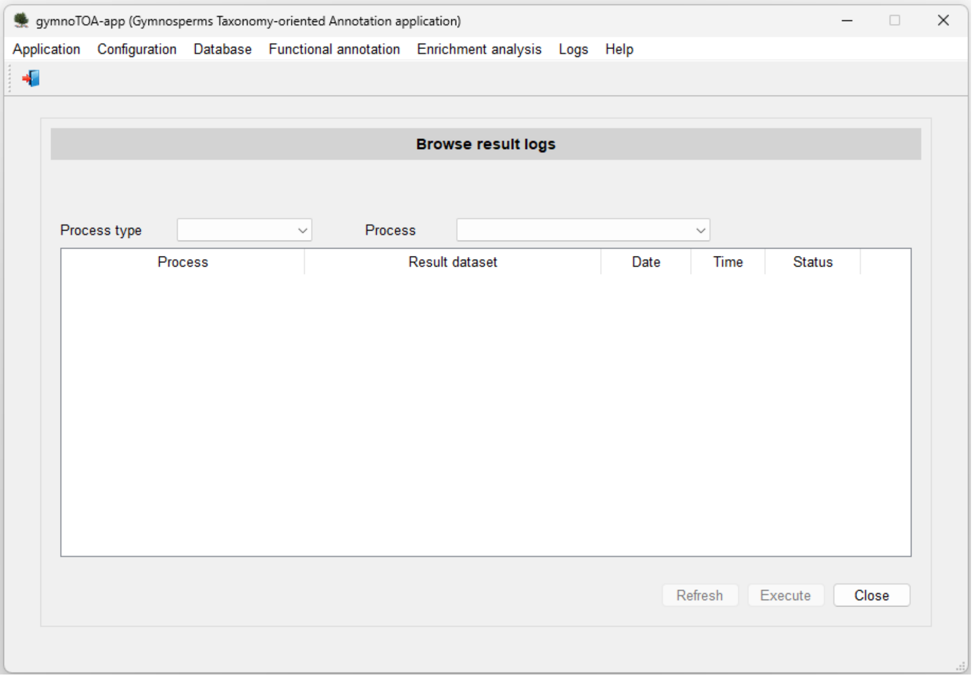
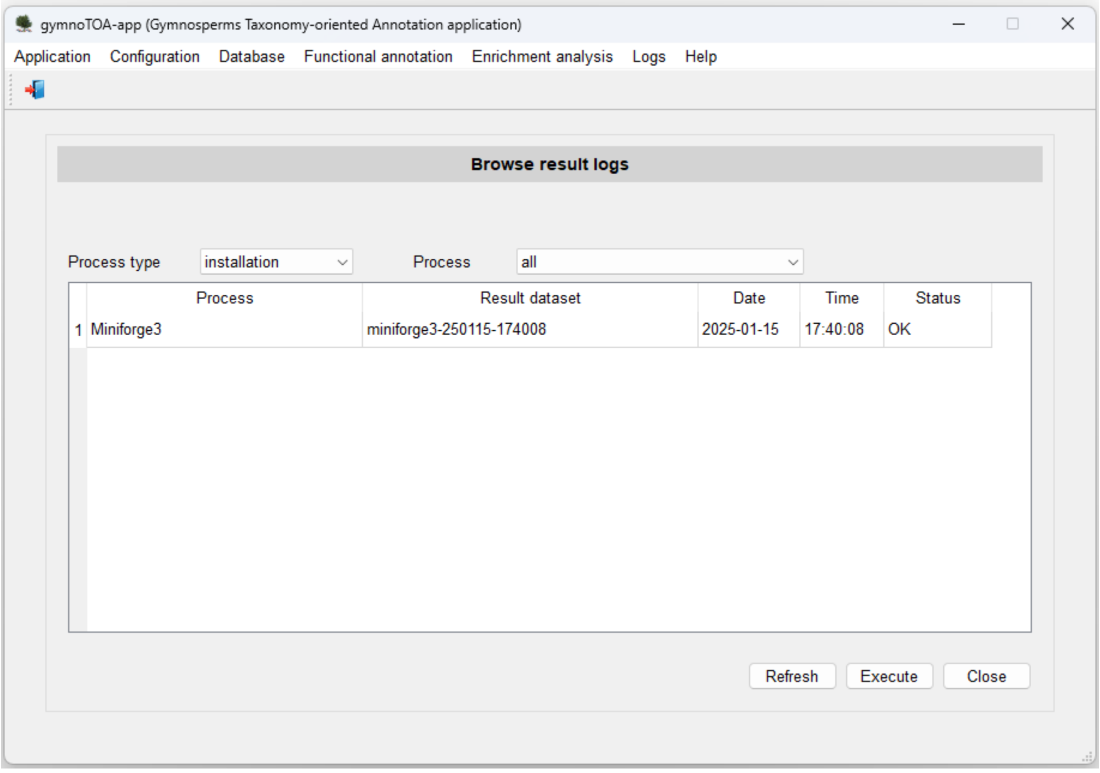


Figure 8. Initial appearance of the window *Browse results logs* on a computer with Windows.

Select **installation** in the *Process type* combo-box. Then the window is updated (Figure 9).

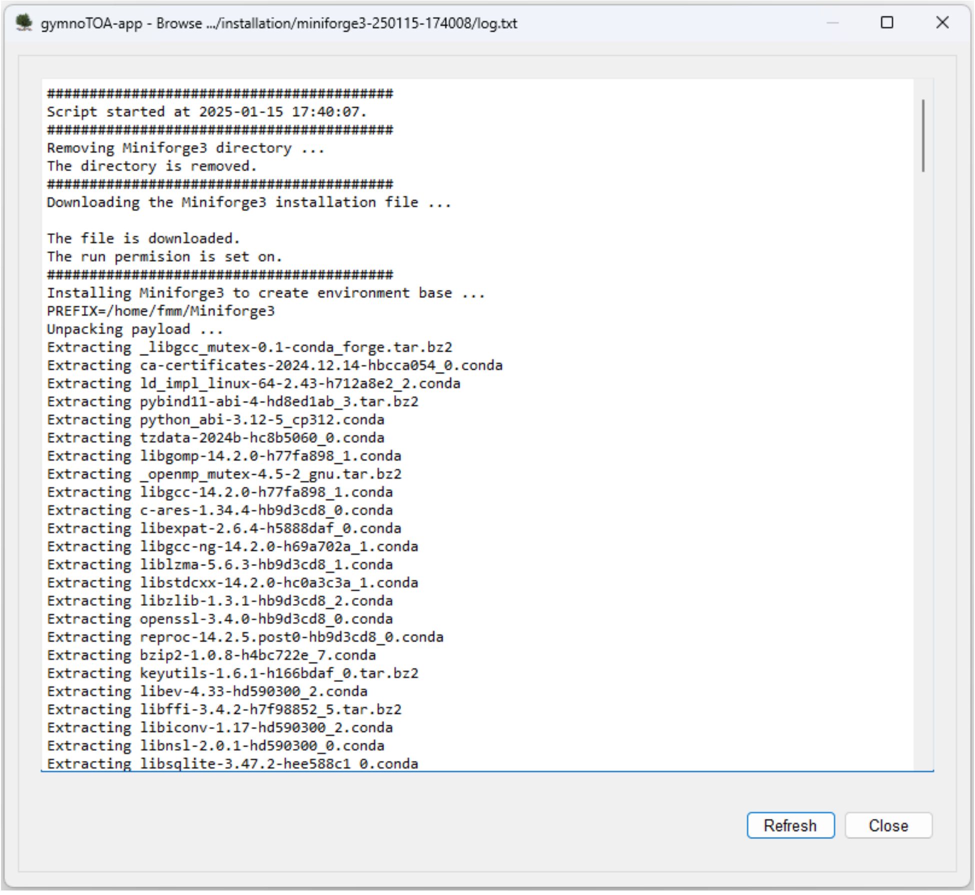


gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application Configuration Database Functional annotation Enrichment analysis Logs Help
Browse result logs
Process type installation
Process all
Refresh
Execute
Close

	Process	Result dataset	Date	Time	Status
1	Miniforge3	miniforge3-250115-174008	2025-01-15	17:40:08	OK

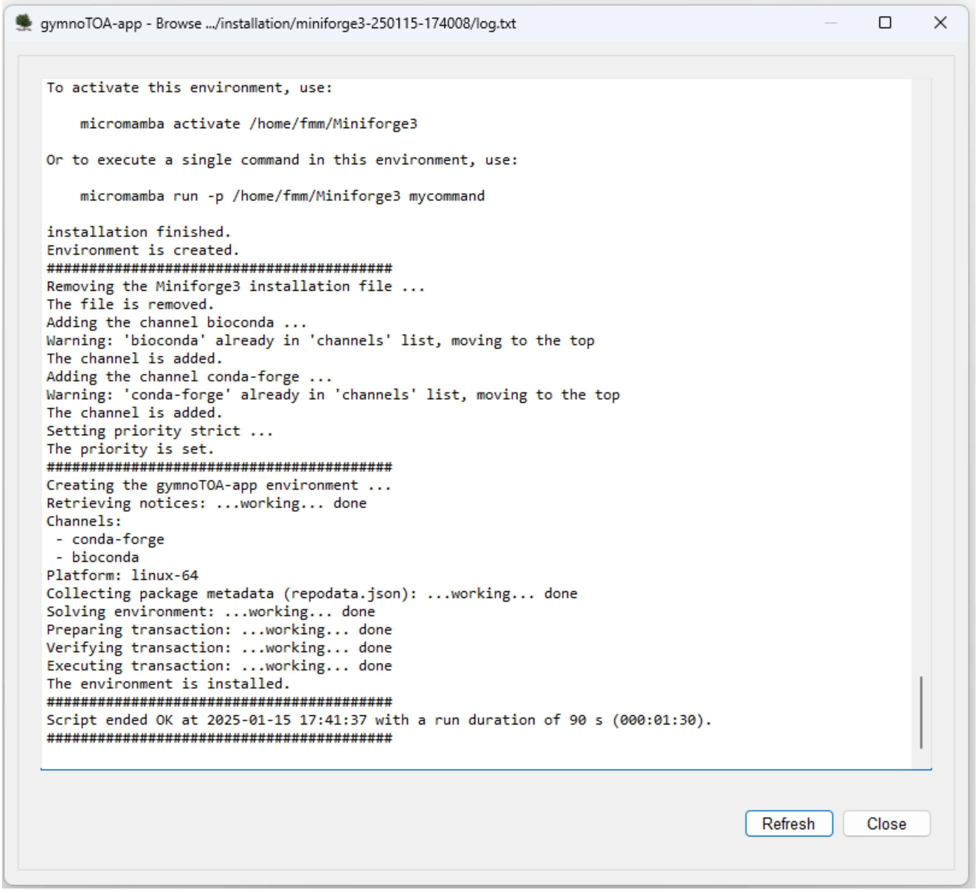
Figure 9. Window *Browse results logs* showing the Miniforge3 installation process finished on a computer with Windows.

So far, we have only performed a single installation process: the process corresponding to the last (and unique) Miniforge3 installation. By double-clicking on it or selecting its row with a click and pressing the Execute button, a new pop-up window appears with its corresponding log (Figure 10).


gymnoTOA-app - Browse .../installation/miniforge3-250115-174008/log.txt
Refresh
Close

```
##########################################
Script started at 2025-01-15 17:40:07.
##########################################
Removing Miniforge3 directory ...
The directory is removed.
##########################################
Downloading the Miniforge3 installation file ...

The file is downloaded.
The run permision is set on.
##########################################
Installing Miniforge3 to create environment base ...
PREFIX=/home/fmm/Miniforge3
Unpacking payload ...
Extracting _libgcc_mutex-0.1-conda_forge.tar.bz2
Extracting ca-certificates-2024.12.14-hbcca054_0.conda
Extracting ld_impl_linux-64-2.43-h712a8e2_2.conda
Extracting pybind11-abi-4-hd8ed1ab_3.tar.bz2
Extracting python_abi-3.12-5_cp312.conda
Extracting tzdata-2024b-hc8b5060_0.conda
Extracting libgomp-14.2.0-h77fa898_1.conda
Extracting _openmp_mutex-4.5-2_gnu.tar.bz2
Extracting libgcc-14.2.0-h77fa898_1.conda
Extracting c-ares-1.34.4-hb9d3cd8_0.conda
Extracting libexpat-2.6.4-h5888daf_0.conda
Extracting libgcc-ng-14.2.0-h69a702a_1.conda
Extracting liblzma-5.6.3-hb9d3cd8_1.conda
Extracting libstdcxx-14.2.0-hc0a3c3a_1.conda
Extracting libzlib-1.3.1-hb9d3cd8_2.conda
Extracting openssl-3.4.0-hb9d3cd8_0.conda
Extracting reproc-14.2.5.post0-hb9d3cd8_0.conda
Extracting bzip2-1.0.8-h4bc722e_7.conda
Extracting keyutils-1.6.1-h166bdaf_0.tar.bz2
Extracting libev-4.33-hd590300_2.conda
Extracting libffi-3.4.2-h7f98852_5.tar.bz2
Extracting libiconv-1.17-hd590300_2.conda
Extracting libnsl-2.0.1-hd590300_0.conda
Extracting libsqlite-3.47.2-hee588c1_0.conda
```


gymnoTOA-app - Browse .../installation/miniforge3-250115-174008/log.txt
Refresh
Close

```
To activate this environment, use:

    micromamba activate /home/fmm/Miniforge3

Or to execute a single command in this environment, use:

    micromamba run -p /home/fmm/Miniforge3 mycommand

installation finished.
Environment is created.
##########################################
Removing the Miniforge3 installation file ...
The file is removed.
Adding the channel bioconda ...
Warning: 'bioconda' already in 'channels' list, moving to the top
The channel is added.
Adding the channel conda-forge ...
Warning: 'conda-forge' already in 'channels' list, moving to the top
The channel is added.
Setting priority strict ...
The priority is set.
##########################################
Creating the gymnoTOA-app environment ...
Retrieving notices: ...working... done
Channels:
 - conda-forge
 - bioconda
Platform: linux-64
Collecting package metadata (repodata.json): ...working... done
Solving environment: ...working... done
Preparing transaction: ...working... done
Verifying transaction: ...working... done
Executing transaction: ...working... done
The environment is installed.
##########################################
Script ended OK at 2025-01-15 17:41:37 with a run duration of 90 s (000:01:30).
##########################################
```

Figure 10. Begin and end of a log of Miniforge3 installation process on a computer with Windows.

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.

Installing bioinformatic software

GYMNOTOA-APP's dependencies are the following:

- BLAST+ (https://blast.ncbi.nlm.nih.gov/). It is used to find local similarity between transcripts or predicted peptides and sequences of genomic databases.
- CodAn (https://github.com/pedronachtigall/CodAn/). It is a software package to characterize the CDS and UTR regions on transcripts from any Eukaryote species.
- DIAMOND (https://github.com/bbuchfink/diamond/) is a faster BLAST alternative (100x-10,000x) to align protein and predicted peptides though it reports fewer matches.

The installation of these packages is automatic by using Bioconda (https://bioconda.github.io/) which holds a large number of bioinformatics software packages.

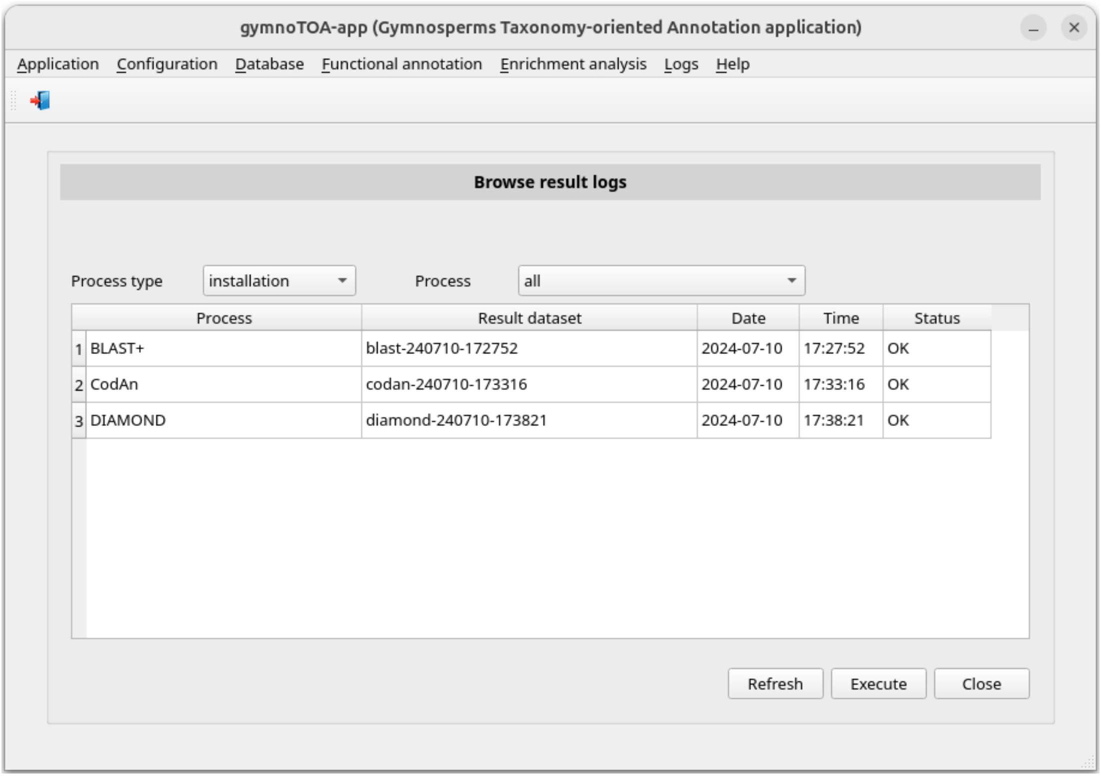
We install BLAST+, CodAn and DIAMOND selecting the menu items with these paths:

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

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

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

The installation steps of each software are like Miniconda3. Finally, the window corresponding to the installation processes will be similar to Figure 11.



Process type: installation

Process: all

	Process	Result dataset	Date	Time	Status
1	BLAST+	blast-240710-172752	2024-07-10	17:27:52	OK
2	CodAn	codan-240710-173316	2024-07-10	17:33:16	OK
3	DIAMOND	diamond-240710-173821	2024-07-10	17:38:21	OK

Figure 11. Window *Browse results logs* once all installation processes are finished on a computer with Ubuntu.

Consulting submitted processes and troubleshooting

The correct operability of the submitted processes is controlled by logs similar to the one described in the section “Installing Miniforge3 and infrastructure packages in WSL (only on Windows)” (Figure 10). Doing 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 another one.

A step by step example in GUI environments

We have sequencing data corresponding to an RNA-seq Illumina library of an experiment about the process of healing after wounding the xylem of the stem of the Canary Island pine (*Pinus canariensis*). We are going to annotate the transcriptome yielded by the assemble phase and perform an enrichment analysis. In this example, we will use a subset of 1000 sequences included in the subdirectory sample-data of GYMNOTOA-APP.

The steps that we are going to take are:

- GYMNOTOA-DB
  - Download GYMNOTOA-DB
  - View statistics of GYMNOTOA-DB
  - Run a functional annotation pipeline
- Funtional annotation
  - Browse result of the functional annotation
  - View statistics of the functional annotation
- Enrichment analysis
  - Run an enrichment analysis
  - Browse results of the enrichment analysis

GYMNOTOA-DB has to be downloaded at the beginning of the use of GYMNOTOA-APP. We should only repeat the downloading when the database has been updated on the server.

GYMNOTOA-DB

Download GYMNOTOA-DB

We download the GYMNOTOA-DB dataset file selecting the menu item with this path:

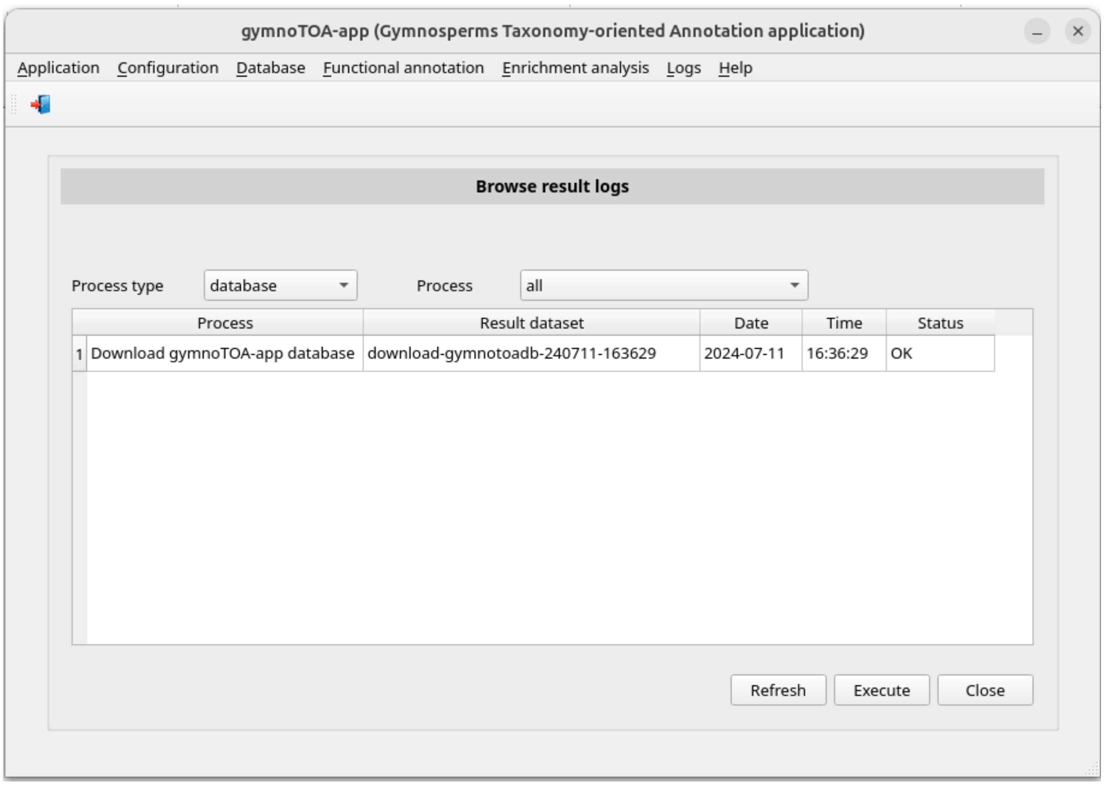
> *Main menu > Database > Download gymnoTOA-db [Execute]*

Then a process is submitted which will access the server, download the latest version of the database and decompress it.

You can check when the process has been completed reviewing its log. So, click the following menu item:

> *Main menu > Logs > Result logs*

Select **database** in the *Process type* combo-box. Then the window is updated (Figure 12).



Process type: database

Process: all

	Process	Result dataset	Date	Time	Status
1	Download gymnoTOA-app database	download-gymnotoadb-240711-163629	2024-07-11	16:36:29	OK

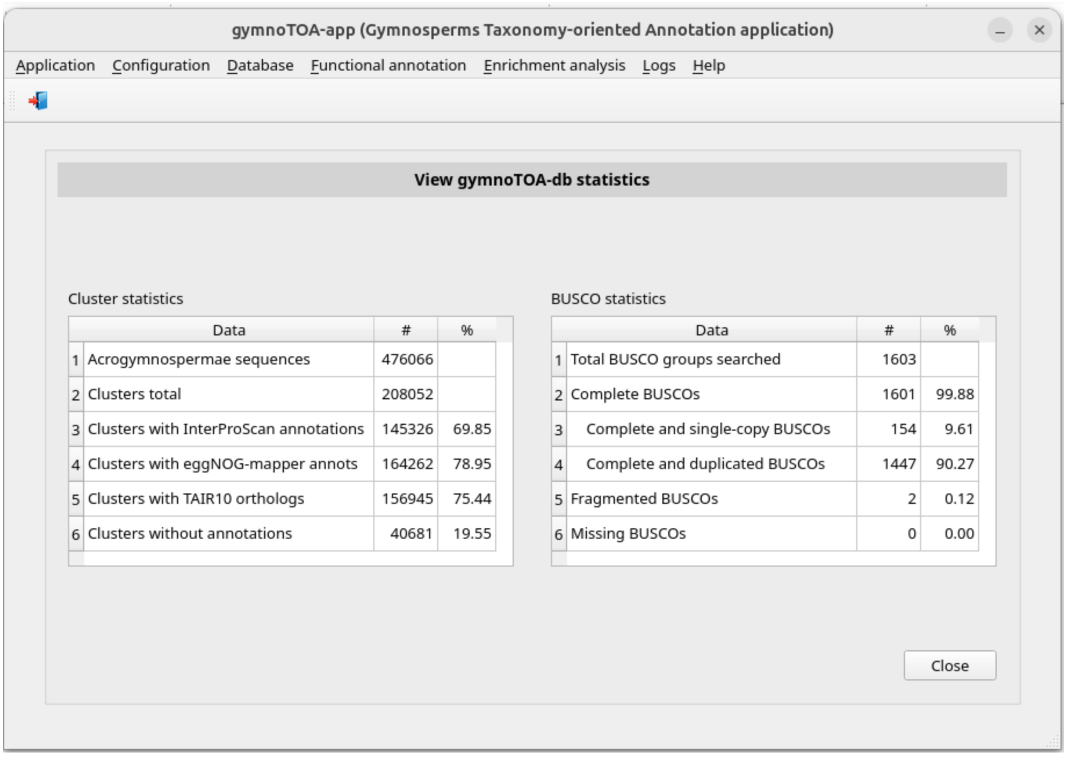
Figure 12. Window *Browse results logs* showing the database process finished.

View statistics of GYMNOTOA-DB

Then, the statistics of GYMNOTOA-DB can be consulted in the menu item with this path:

Main menu > Database > Statistics

The window *View gymnoTOA-db statistics* will be shown (Figure 13).



View gymnoTOA-db statistics

Cluster statistics

	Data	#	%
1	Acrogymnospermae sequences	476066	
2	Clusters total	208052	
3	Clusters with InterProScan annotations	145326	69.85
4	Clusters with eggNOG-mapper annots	164262	78.95
5	Clusters with TAIR10 orthologs	156945	75.44
6	Clusters without annotations	40681	19.55

BUSCO statistics

	Data	#	%
1	Total BUSCO groups searched	1603	
2	Complete BUSCOs	1601	99.88
3	Complete and single-copy BUSCOs	154	9.61
4	Complete and duplicated BUSCOs	1447	90.27
5	Fragmented BUSCOs	2	0.12
6	Missing BUSCOs	0	0.00

Figure 13. Window *View gymnoTOA-db statistics* with data of July 2024 version data.

Functional annotation

Run a functional annotation pipeline

To perform an annotation process (Figure 2) we are going to select the menu item with this path:

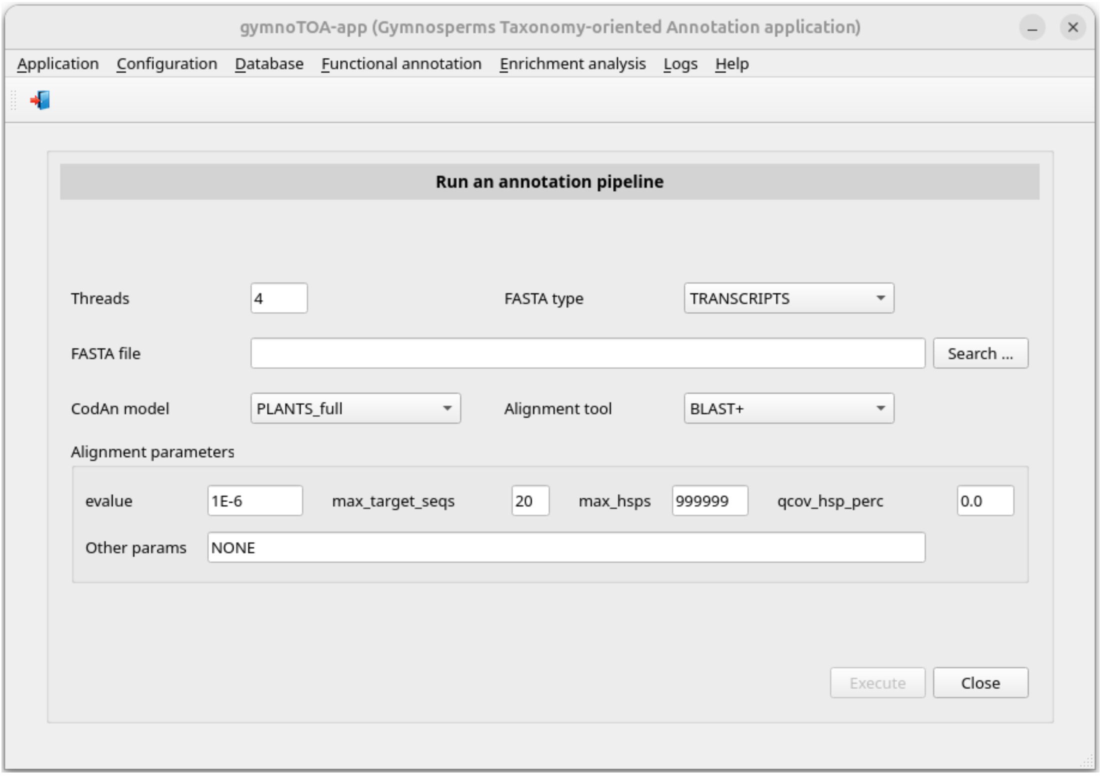
Main menu > Functional annotation > Run pipeline

The window *Run an annotation pipeline* appears (Figure 14). Select the FASTA type option (TRANSCRIPTS is the type of our example). Then type the path of the FASTA file or select it using the button *Search...* (in this example we use the file transcripts file PinusCanariensisXilogenesisGESU01.1-1000seqs.fsa located in the subdirectory sample-data of the gymnoTOA-app software). Then the *Execute* button will be available (Figure 15). Before running the process, you can also modify the default value of the threads number, CodAn model (PLANTS_full or PLANTS_partial), alignment software (BLAST+ or DIAMOND) and its parameters : e_value (number of expected hits of similar quality -score- that could be found just by chance), the number of aligned sequences to keep, the number of HSPs (high-scoring segment pairs) to keep and the query coverage per HSP (0.0 if DIAMOND). In Other parameters you can indicate additional parameters to the previous ones and their corresponding values. Once parameters are updated, click the *Execute* button.

You can check the process status and when it has been completed reviewing its log. So, click the following menu item:

Main menu > Logs > Result logs

Select **run** in the *Process type* combo-box. Then the window is updated (Figure 16).



gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application Configuration Database Functional annotation Enrichment analysis Logs Help

Run an annotation pipeline

Threads 4

FASTA type TRANSCRIPTS

FASTA file

CodAn model PLANTS_full

Alignment tool BLAST+

Alignment parameters

evalue 1E-6

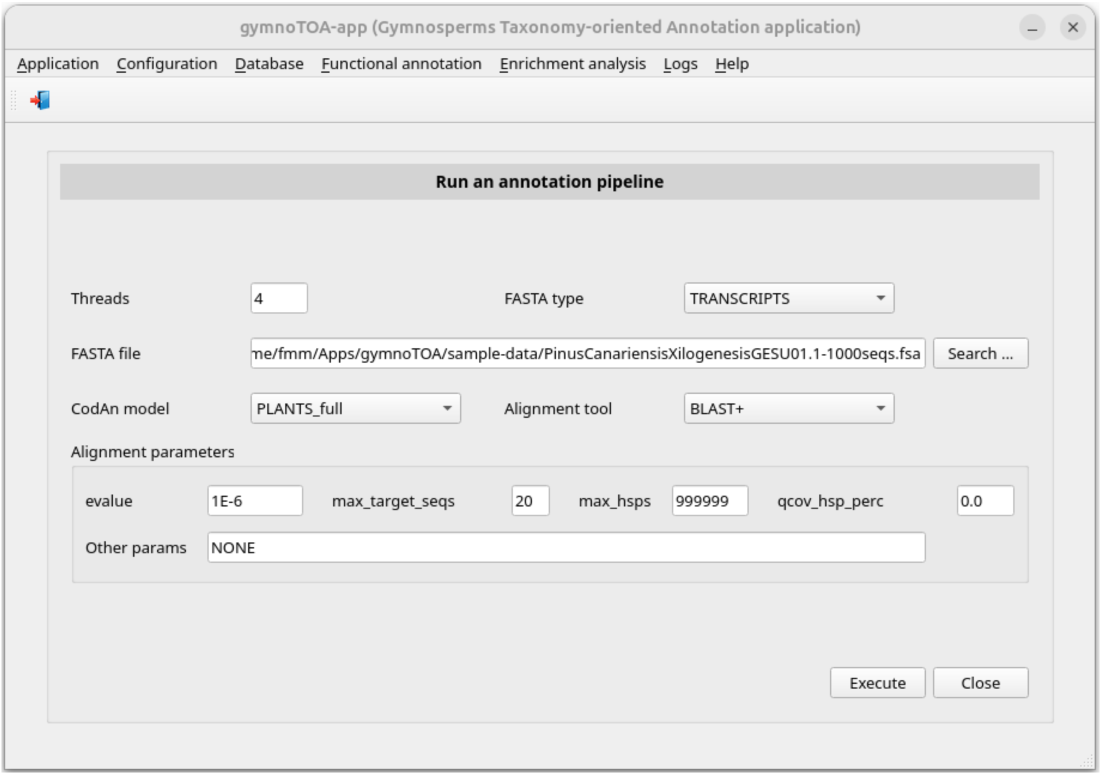
max_target_seqs 20

max_hsps 999999

qcov_hsp_perc 0.0

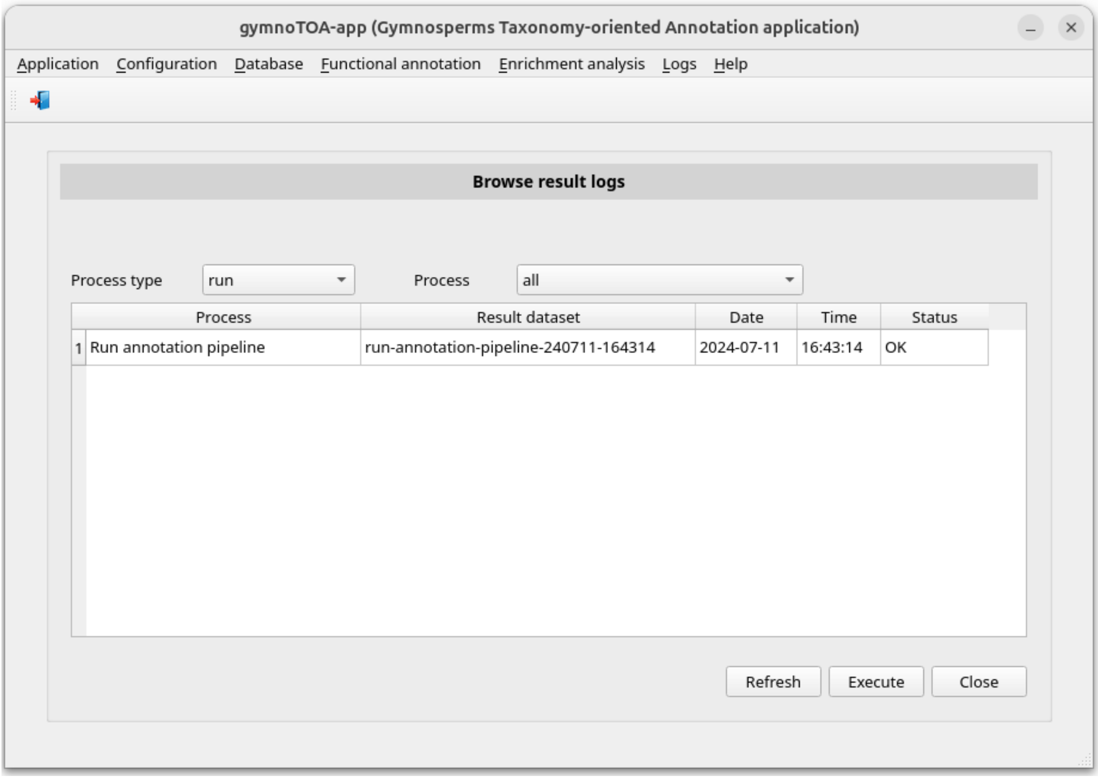
Other params NONE

Figure 14. Initial appearance of the window *Run annotation pipeline*.



Threads: 4
FASTA type: TRANSCRIPTS
FASTA file: me/fmm/Apps/gymnoTOA/sample-data/PinusCanariensisXilogenesisGESU01.1-1000seqs.fsa
CodAn model: PLANTS_full
Alignment tool: BLAST+
Alignment parameters
evalue: 1E-6
max_target_seqs: 20
max_hsps: 999999
qcov_hsp_perc: 0.0
Other params: NONE

Figure 15. Window *Run annotation pipeline* once the transcriptome file has been typed.



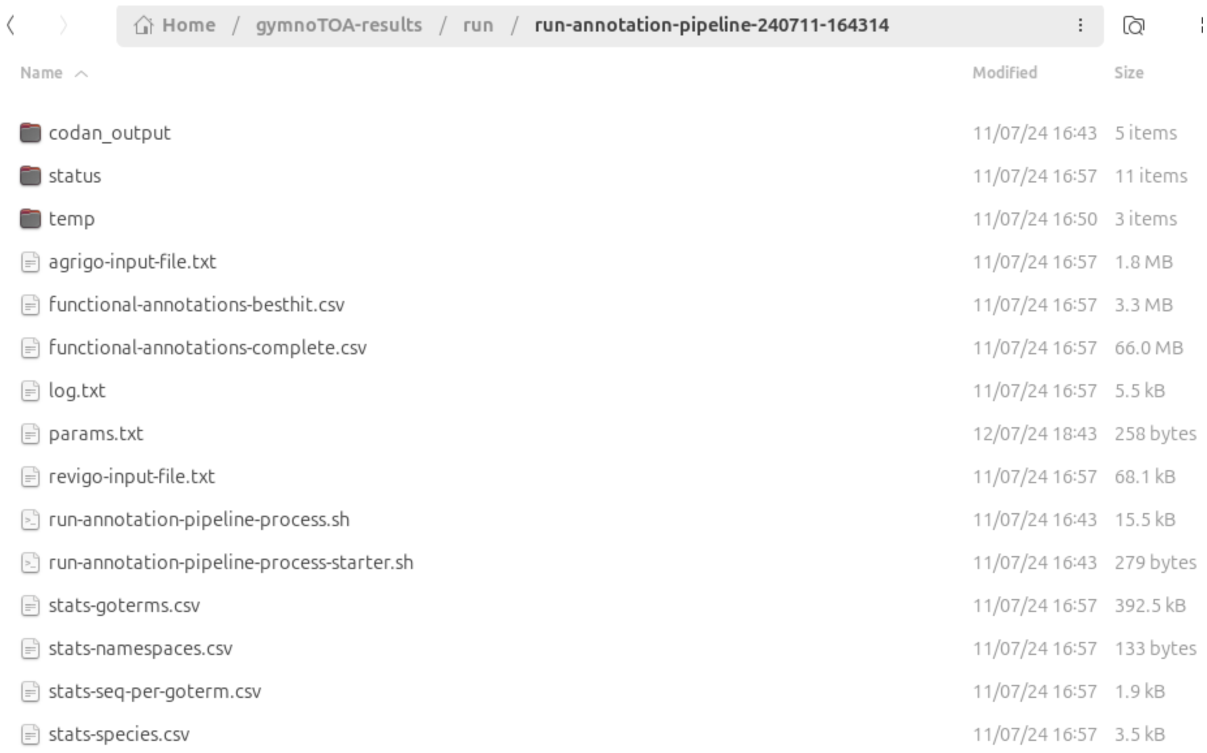
Process type: run
Process: all

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Figure 16. Window *Browse results logs* showing annotation pipelines finished.

When the process has finished, we could go the subdirectory of *…\gymnoTOA-app-results\run* corresponding to the script run, e.g. *run-annotation-pipeline-240711-164314*, and consult the generated files (Figure 17). Some of these files are:

- Annotation:
  - *functional-annotations-complete.csv*: It contains all annotations yielded by blast programs for every query sequence identification.
  - *functional-annotations-besthit.csv*: It contains the annotations of the subject sequence identification with best hit yielded by blast programs for every query sequence identification.
- Statistics:
  - *stats-goterms.csv*: It contains the frequency distribution per GO term.
  - *stats-namespaces.csv*: It contains the frequency distribution per namespace.
  - *stats-seq-per-goterm.csv*: It contains the sequence number per GO term number.
  - *stats-species.csv*: It contains the frequency distribution of species.
- Other applications inputs:
  - *agrigo-input-file.txt*: It contains data to be used as agriGO input.
  - *revigo-input-file.txt*: It contains data to be used as REVIGO input.



Home / gymnoTOA-results / run / run-annotation-pipeline-240711-164314

Name	Modified	Size
codan_output	11/07/24 16:43	5 items
status	11/07/24 16:57	11 items
temp	11/07/24 16:50	3 items
agrigo-input-file.txt	11/07/24 16:57	1.8 MB
functional-annotations-besthit.csv	11/07/24 16:57	3.3 MB
functional-annotations-complete.csv	11/07/24 16:57	66.0 MB
log.txt	11/07/24 16:57	5.5 kB
params.txt	12/07/24 18:43	258 bytes
revigo-input-file.txt	11/07/24 16:57	68.1 kB
run-annotation-pipeline-process.sh	11/07/24 16:43	15.5 kB
run-annotation-pipeline-process-starter.sh	11/07/24 16:43	279 bytes
stats-goterms.csv	11/07/24 16:57	392.5 kB
stats-namespaces.csv	11/07/24 16:57	133 bytes
stats-seq-per-goterm.csv	11/07/24 16:57	1.9 kB
stats-species.csv	11/07/24 16:57	3.5 kB

Figure 17. Folders and files in *…/gymnoTOA-app-results/run/run-annotation-pipeline-240711-164314* yielded by the annotation pipeline.

Browse result of the functional annotation

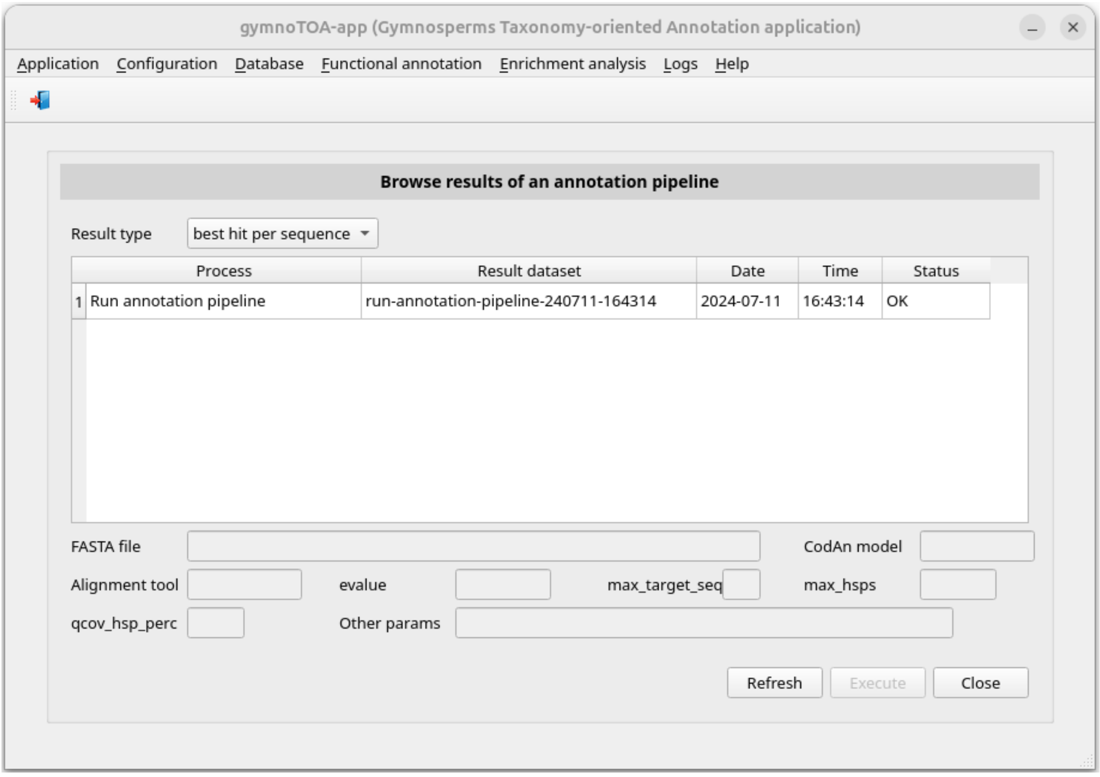
If you edit the files *functional-annotations-complete.csv* or *functional-annotations-besthit.csv*, you will see the result of the function annotation You can browse it using GYMNOTOA-APP in the following menu item:

Main menu > Functional annotation > Browse results of an annotation Pipeline

The window (Figure 18) allows you to choose **best hit per sequence** (*functional-annotations-besthit.csv*) or **all hits per sequence** (*functional-annotations-complete.csv*) in the combo-box *Result type*.

Click on the annotation pipeline that you are interested in consulting, in our example *run-annotation-pipeline-240711-164314*. The window will be updated showing the parameters of the process (Figure 19).

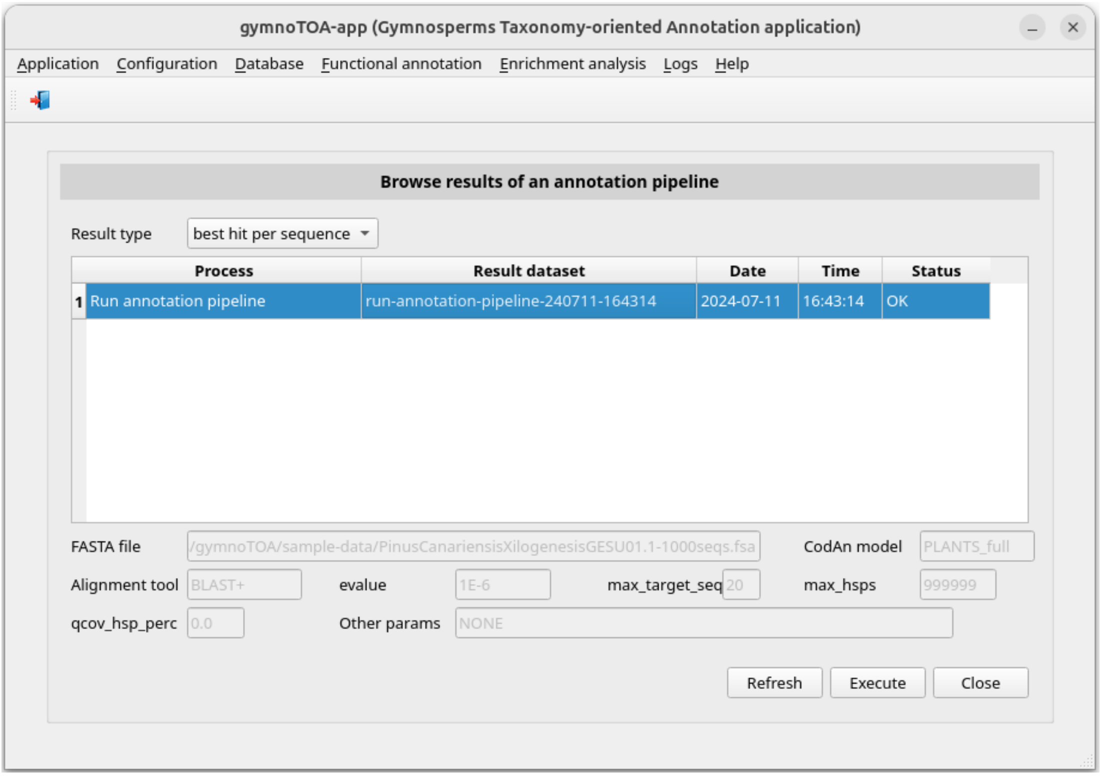
Then click push button Execute and a popup window will appear showing the functional annotation data (Figure 20).



Result type: best hit per sequence

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Figure 18. Window *Browse results of an annotation pipelines* showing annotation pipelines finished.



gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application Configuration Database Functional annotation Enrichment analysis Logs Help
Refresh
Execute
Close

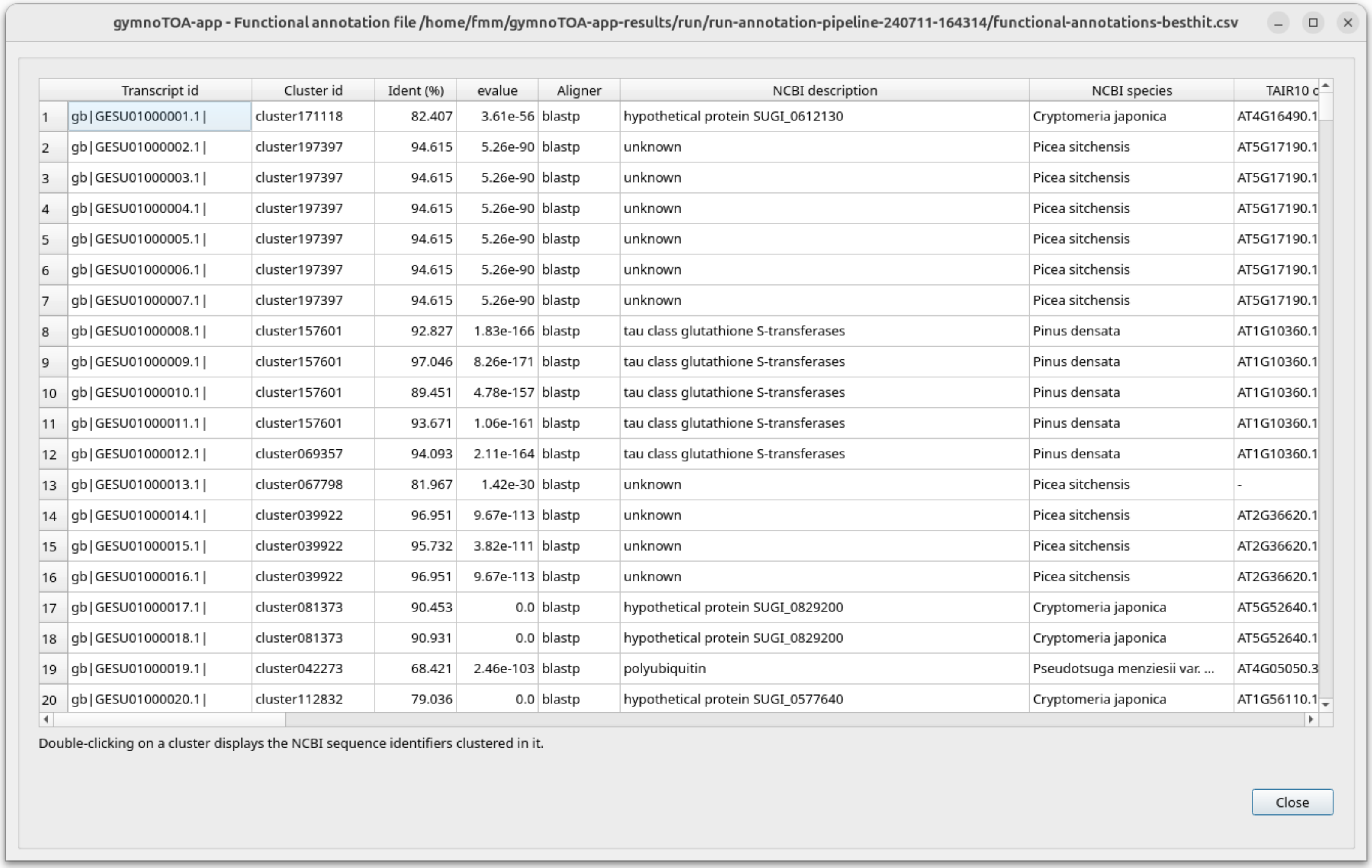
Browse results of an annotation pipeline

Result type: best hit per sequence

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

FASTA file: /gymnoTOA/sample-data/PinusCanariensisXilogenesisGESU01.1-1000seqs.fsa
CodAn model: PLANTS_full
Alignment tool: BLAST+
evalue: 1E-6
max_target_seq: 20
max_hsps: 999999
qcov_hsp_perc: 0.0
Other params: NONE

Figure 19. Window *Browse results of an annotation pipelines* after *run-annotation-pipeline-240711-164314* was selected.



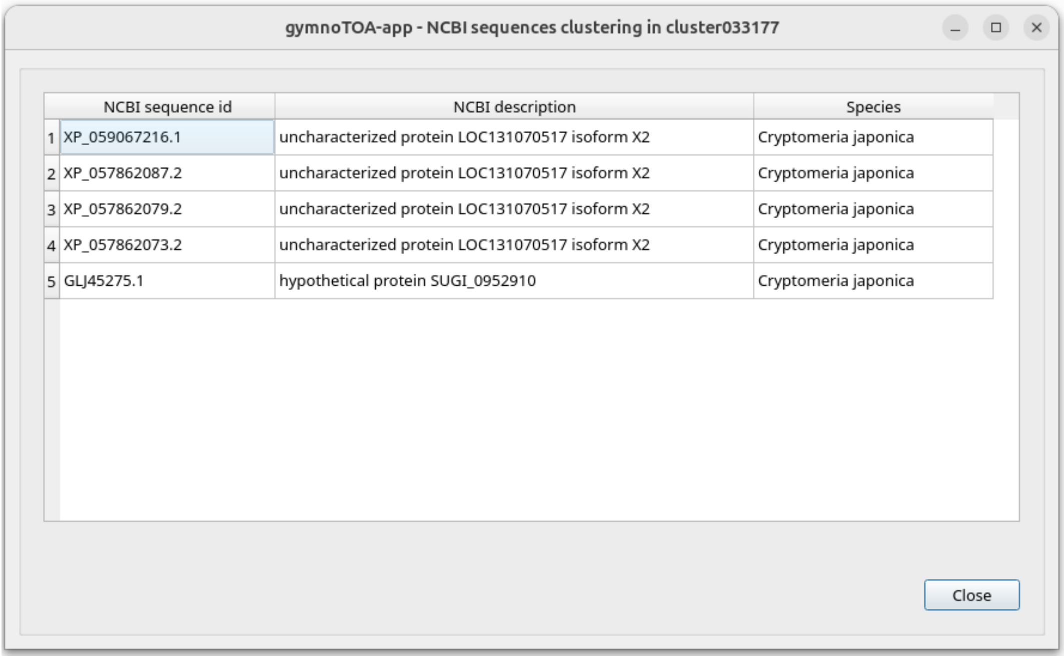
gymnoTOA-app - Functional annotation file /home/fmm/gymnoTOA-app-results/run/run-annotation-pipeline-240711-164314/functional-annotations-besthit.csv

	Transcript id	Cluster id	Ident (%)	evalue	Aligner	NCBI description	NCBI species	TAIR10 c
1	gb|GESU01000001.1|	cluster171118	82.407	3.61e-56	blastp	hypothetical protein SUGI_0612130	Cryptomeria japonica	AT4G16490.1
2	gb|GESU01000002.1|	cluster197397	94.615	5.26e-90	blastp	unknown	Picea sitchensis	AT5G17190.1
3	gb|GESU01000003.1|	cluster197397	94.615	5.26e-90	blastp	unknown	Picea sitchensis	AT5G17190.1
4	gb|GESU01000004.1|	cluster197397	94.615	5.26e-90	blastp	unknown	Picea sitchensis	AT5G17190.1
5	gb|GESU01000005.1|	cluster197397	94.615	5.26e-90	blastp	unknown	Picea sitchensis	AT5G17190.1
6	gb|GESU01000006.1|	cluster197397	94.615	5.26e-90	blastp	unknown	Picea sitchensis	AT5G17190.1
7	gb|GESU01000007.1|	cluster197397	94.615	5.26e-90	blastp	unknown	Picea sitchensis	AT5G17190.1
8	gb|GESU01000008.1|	cluster157601	92.827	1.83e-166	blastp	tau class glutathione S-transferases	Pinus densata	AT1G10360.1
9	gb|GESU01000009.1|	cluster157601	97.046	8.26e-171	blastp	tau class glutathione S-transferases	Pinus densata	AT1G10360.1
10	gb|GESU01000010.1|	cluster157601	89.451	4.78e-157	blastp	tau class glutathione S-transferases	Pinus densata	AT1G10360.1
11	gb|GESU01000011.1|	cluster157601	93.671	1.06e-161	blastp	tau class glutathione S-transferases	Pinus densata	AT1G10360.1
12	gb|GESU01000012.1|	cluster069357	94.093	2.11e-164	blastp	tau class glutathione S-transferases	Pinus densata	AT1G10360.1
13	gb|GESU01000013.1|	cluster067798	81.967	1.42e-30	blastp	unknown	Picea sitchensis	-
14	gb|GESU01000014.1|	cluster039922	96.951	9.67e-113	blastp	unknown	Picea sitchensis	AT2G36620.1
15	gb|GESU01000015.1|	cluster039922	95.732	3.82e-111	blastp	unknown	Picea sitchensis	AT2G36620.1
16	gb|GESU01000016.1|	cluster039922	96.951	9.67e-113	blastp	unknown	Picea sitchensis	AT2G36620.1
17	gb|GESU01000017.1|	cluster081373	90.453	0.0	blastp	hypothetical protein SUGI_0829200	Cryptomeria japonica	AT5G52640.1
18	gb|GESU01000018.1|	cluster081373	90.931	0.0	blastp	hypothetical protein SUGI_0829200	Cryptomeria japonica	AT5G52640.1
19	gb|GESU01000019.1|	cluster042273	68.421	2.46e-103	blastp	polyubiquitin	Pseudotsuga menziesii var. ...	AT4G05050.3
20	gb|GESU01000020.1|	cluster112832	79.036	0.0	blastp	hypothetical protein SUGI_0577640	Cryptomeria japonica	AT1G56110.1

Double-clicking on a cluster displays the NCBI sequence identifiers clustered in it.

Figure 20. Popup window showing the annotation performed by the process *run-annotation-pipeline-240711-164314*.

If you double-click on a cluster identification, e.g. *cluster033177*, other popup will appear with the NCBI sequences that make up the cluster (Figure 21).



gymnoTOA-app - NCBI sequences clustering in cluster033177

	NCBI sequence id	NCBI description	Species
1	XP_059067216.1	uncharacterized protein LOC131070517 isoform X2	Cryptomeria japonica
2	XP_057862087.2	uncharacterized protein LOC131070517 isoform X2	Cryptomeria japonica
3	XP_057862079.2	uncharacterized protein LOC131070517 isoform X2	Cryptomeria japonica
4	XP_057862073.2	uncharacterized protein LOC131070517 isoform X2	Cryptomeria japonica
5	GLJ45275.1	hypothetical protein SUGI_0952910	Cryptomeria japonica

Figure 21. Popup window showing the sequence composition of the cluster *cluster033177*.

View statistics of the functional annotation

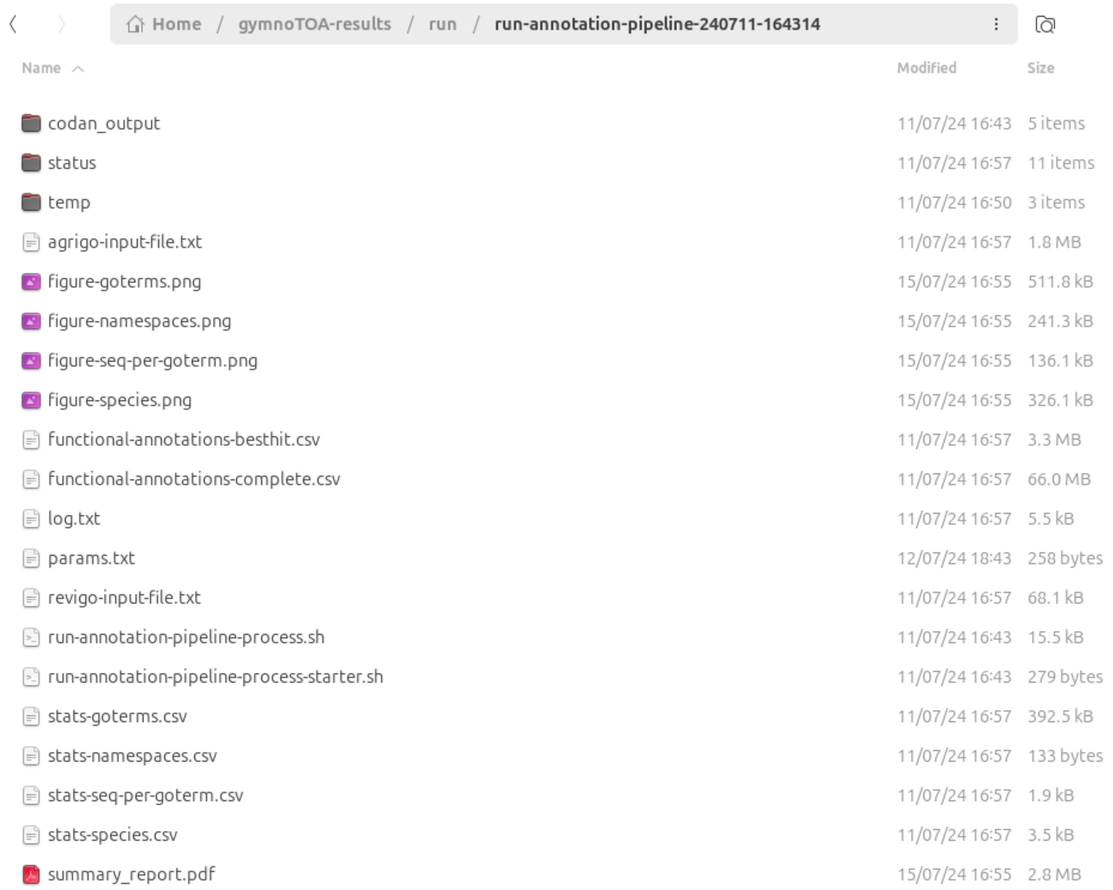
Annotation pipelines generate some statistics: frequency distribution of species, frequency distribution per GO term, frequency distribution per namespace and sequence number per GO term number.

First, we build a summary report using the menu item with this path:

Main menu > Functional annotation > Statistics > Summary report [Execute]

A PDF document will be shown with plots of the above statistics. If you go the subdirectory of *…\gymnoTOA-app-results\run* corresponding to the script run and consult the files (Figure 22), you will see these new files:

- *summary_report.pdf*: report including the following PNG files
- *figure-species.png*: frequency distribution of species
- *figure-goterms.png*: frequency distribution per GO term
- *figure-namespaces.png*: frequency distribution per namespace
- *figure-seq-per-goterm.png*: sequence number per GO term number



Home / gymnoTOA-results / run / run-annotation-pipeline-240711-164314

Name	Modified	Size
codan_output	11/07/24 16:43	5 items
status	11/07/24 16:57	11 items
temp	11/07/24 16:50	3 items
agrigo-input-file.txt	11/07/24 16:57	1.8 MB
figure-goterms.png	15/07/24 16:55	511.8 kB
figure-namespaces.png	15/07/24 16:55	241.3 kB
figure-seq-per-goterm.png	15/07/24 16:55	136.1 kB
figure-species.png	15/07/24 16:55	326.1 kB
functional-annotations-besthit.csv	11/07/24 16:57	3.3 MB
functional-annotations-complete.csv	11/07/24 16:57	66.0 MB
log.txt	11/07/24 16:57	5.5 kB
params.txt	12/07/24 18:43	258 bytes
revigo-input-file.txt	11/07/24 16:57	68.1 kB
run-annotation-pipeline-process.sh	11/07/24 16:43	15.5 kB
run-annotation-pipeline-process-starter.sh	11/07/24 16:43	279 bytes
stats-goterms.csv	11/07/24 16:57	392.5 kB
stats-namespaces.csv	11/07/24 16:57	133 bytes
stats-seq-per-goterm.csv	11/07/24 16:57	1.9 kB
stats-species.csv	11/07/24 16:57	3.5 kB
summary_report.pdf	15/07/24 16:55	2.8 MB

Figure 22. Folders and files in *.../gymnoTOA-app-results/run/run-annotation-pipeline-240711-164314* after summary report was built.

You also can browse statistical data or make plots with different formats and resolutions using GYMNOTOA-APP. First, we consult data of frequency distribution of species in the following menu item:

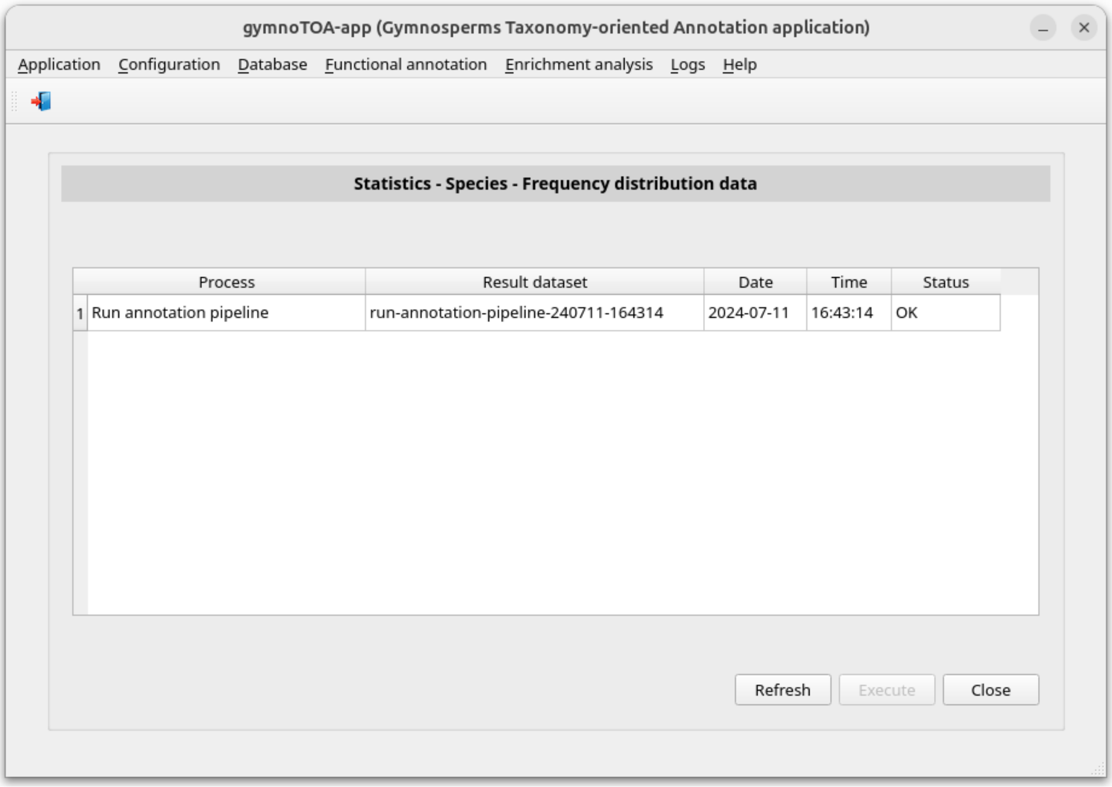
Main menu > Functional annotation > Statistics > Species > Frequency distribution data

In the new window (Figure 23), double-click on the annotation process or select with a click on it and press the *Execute* button. Pop-up window will appear with data of frequency distribution of species (Figure 24).

We view the plot corresponding to the top ten data selecting the menu item

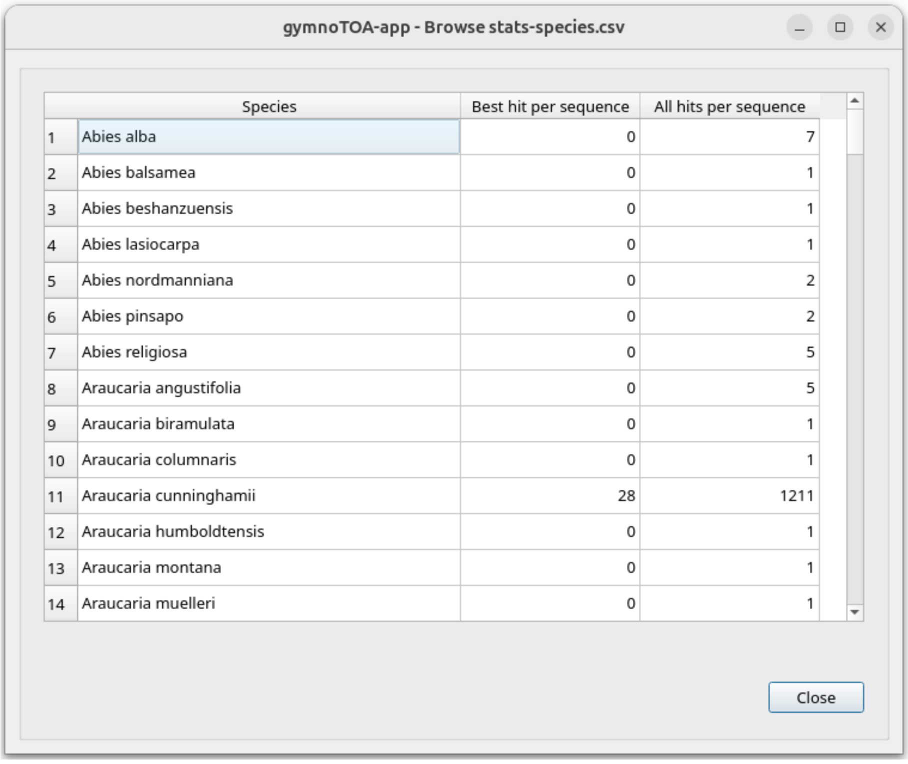
Main menu > Functional annotation > Statistics > Species > Frequency distribution data

In the window shown below (Figure 25), you can modify the default values of the file name and its format and resolution. After, double-click on the annotation process or select with a click on it and press the *Execute* button. Pop-up window will appear with the plot of frequency distribution of species (Figure 26). The corresponding file is saved in the subdirectory of ...\gymnoTOA-results\run corresponding to the script run.


gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application
Configuration
Database
Functional annotation
Enrichment analysis
Logs
Help
Statistics - Species - Frequency distribution data
Refresh
Execute
Close

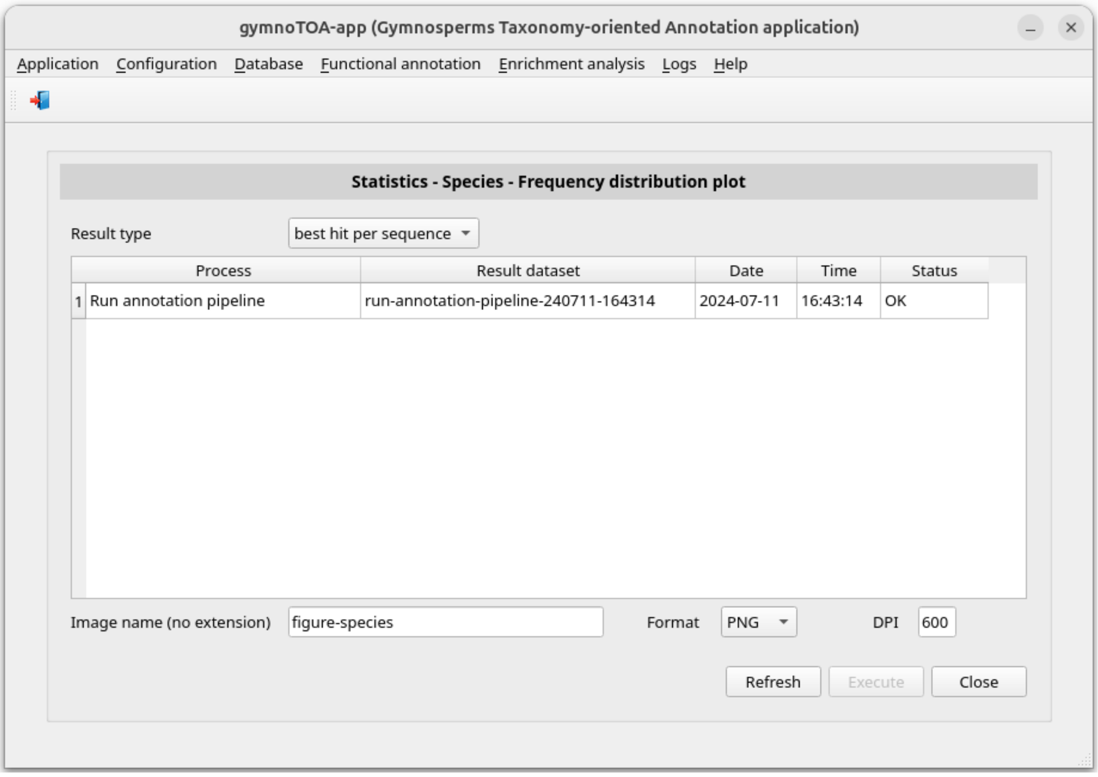
	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Figure 23. Window *Statistics - Species - Frequency distribution data* showing annotation pipelines finished.



	Species	Best hit per sequence	All hits per sequence
1	Abies alba	0	7
2	Abies balsamea	0	1
3	Abies beshanzuensis	0	1
4	Abies lasiocarpa	0	1
5	Abies nordmanniana	0	2
6	Abies pinsapo	0	2
7	Abies religiosa	0	5
8	Araucaria angustifolia	0	5
9	Araucaria biramulata	0	1
10	Araucaria columnaris	0	1
11	Araucaria cunninghamii	28	1211
12	Araucaria humboldtensis	0	1
13	Araucaria montana	0	1
14	Araucaria muelleri	0	1

Figure 24. Popup window showing the complete list of frequency distribution performed by the process *run-annotation-pipeline-240711-164314*.



Statistics - Species - Frequency distribution plot

Result type: best hit per sequence

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Image name (no extension): figure-species

Format: PNG

DPI: 600

Figure 25. Window *Statistics - Species - Frequency distribution plot* showing annotation pipelines finished.

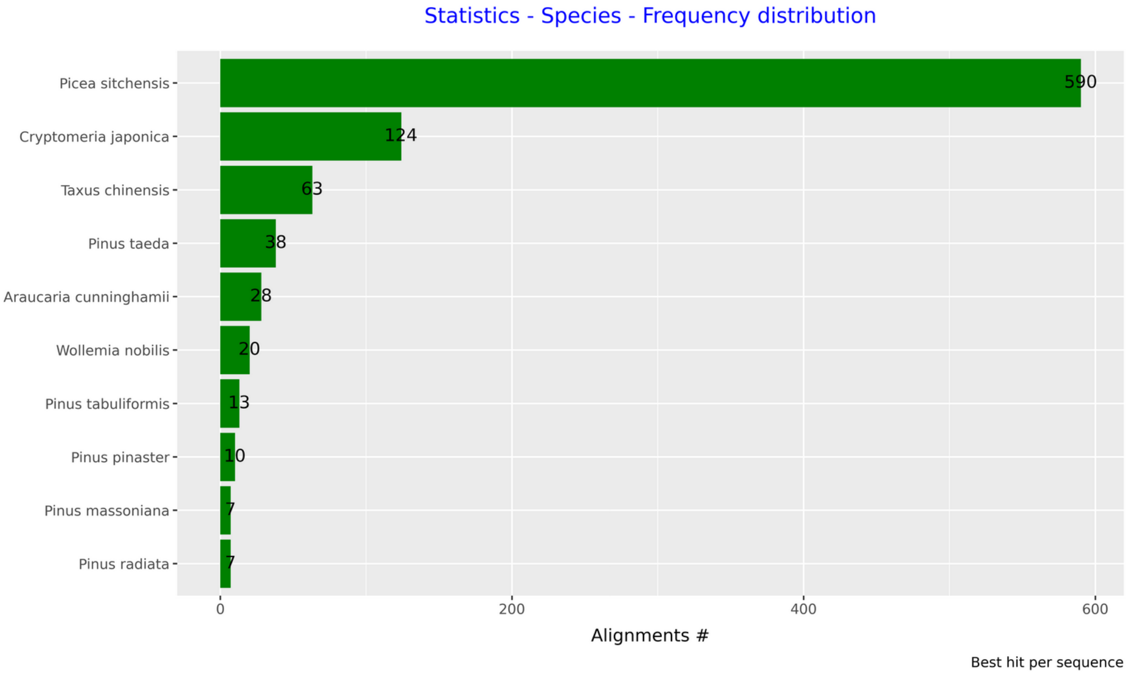


Figure 26.Plot *Statistics - Species - Frequency distribution* with the top ten of species.

The frequency distribution per GO term can be consulted in the menu item with this path:

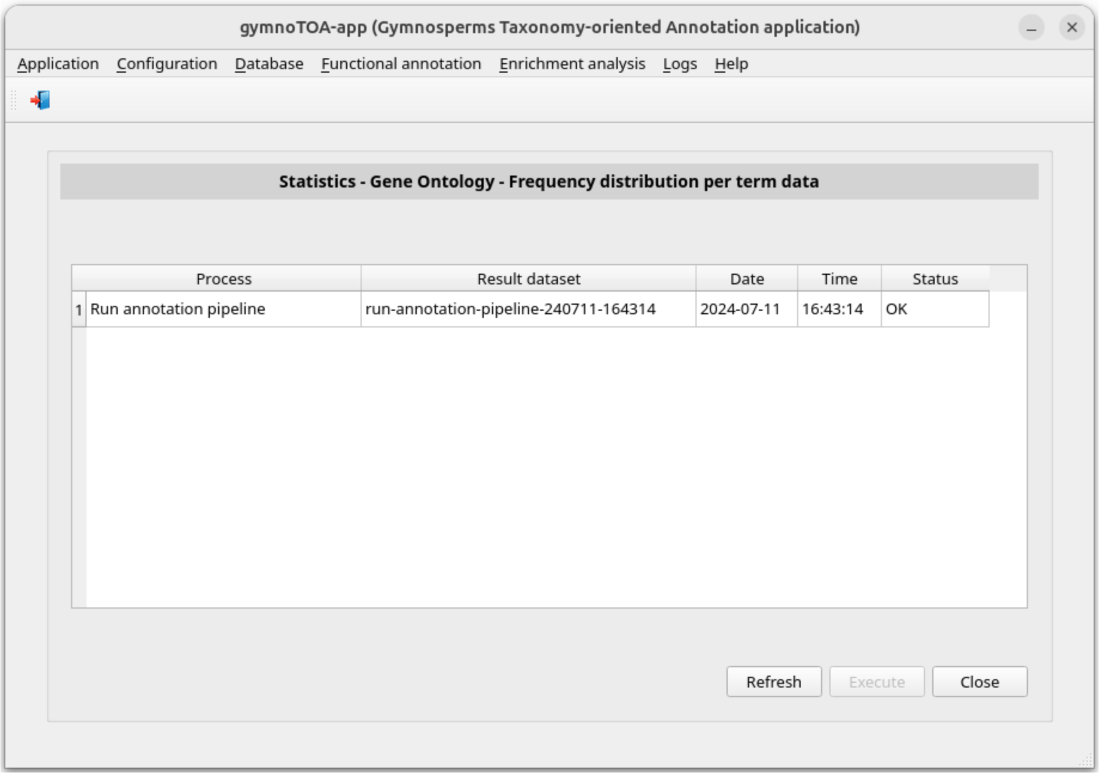
> *Main menu > Functional annotation > Statistics > Gene Ontology > Frequency distribution per GO term data*

In the new window (Figure 27), double-click on the annotation process or select with a click on it and press the *Execute* button. Pop-up window will appear with data of frequency distribution per GO term (Figure 28).

We view the plot corresponding to the top ten data selecting the menu item:

> *Main menu > Functional annotation > Statistics > Gene Ontology > Frequency distribution per GO term plot*

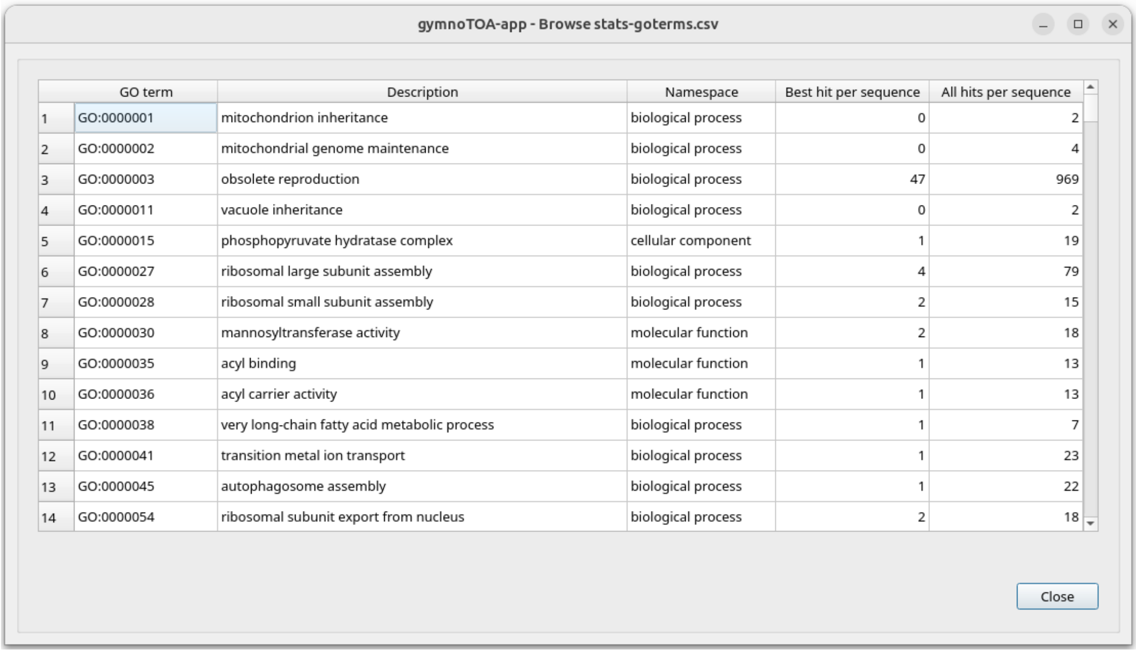
In the window shown below (Figure 29), you can modify the default values of the file name and its format and resolution. After, double-click on the annotation process or select with a click on it and press the *Execute* button. Pop-up window will appear with the plot of frequency distribution per GO term (Figure 30). The corresponding file is saved in the subdirectory of …\gymnoTOA-results\run corresponding to the script run.



Statistics - Gene Ontology - Frequency distribution per term data

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

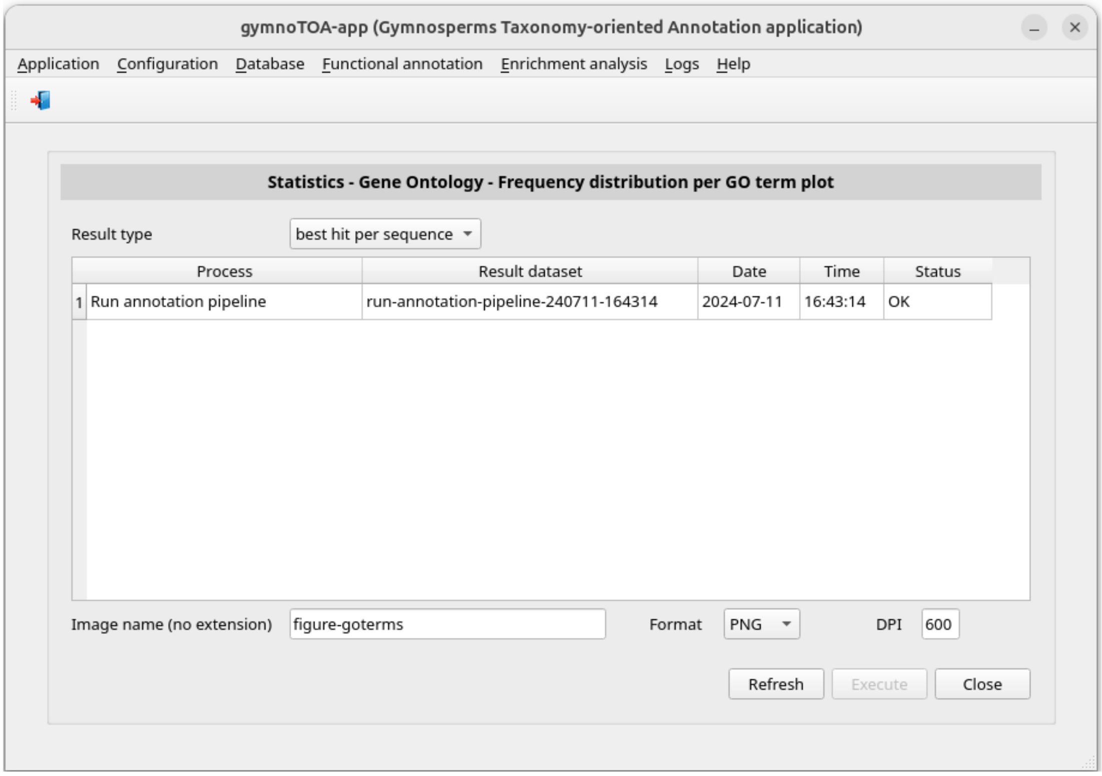
Figure 27. Window *Statistics – Gene Ontology - Frequency distribution per term data* showing annotation pipelines finished.



gymnoTOA-app - Browse stats-goterms.csv

	GO term	Description	Namespace	Best hit per sequence	All hits per sequence
1	GO:0000001	mitochondrion inheritance	biological process	0	2
2	GO:0000002	mitochondrial genome maintenance	biological process	0	4
3	GO:0000003	obsolete reproduction	biological process	47	969
4	GO:0000011	vacuole inheritance	biological process	0	2
5	GO:0000015	phosphopyruvate hydratase complex	cellular component	1	19
6	GO:0000027	ribosomal large subunit assembly	biological process	4	79
7	GO:0000028	ribosomal small subunit assembly	biological process	2	15
8	GO:0000030	mannosyltransferase activity	molecular function	2	18
9	GO:0000035	acyl binding	molecular function	1	13
10	GO:0000036	acyl carrier activity	molecular function	1	13
11	GO:0000038	very long-chain fatty acid metabolic process	biological process	1	7
12	GO:0000041	transition metal ion transport	biological process	1	23
13	GO:0000045	autophagosome assembly	biological process	1	22
14	GO:0000054	ribosomal subunit export from nucleus	biological process	2	18

Figure 28. Popup window showing the complete list of frequency distribution performed by the process *run-annotation-pipeline-240711-164314*.



gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)

Application Configuration Database Functional annotation Enrichment analysis Logs Help

Statistics - Gene Ontology - Frequency distribution per GO term plot

Result type: best hit per sequence

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Image name (no extension): figure-goterms

Format: PNG

DPI: 600

Figure 29. Window *Statistics - Gene Ontology - Frequency distribution per GO term* showing annotation pipelines finished.

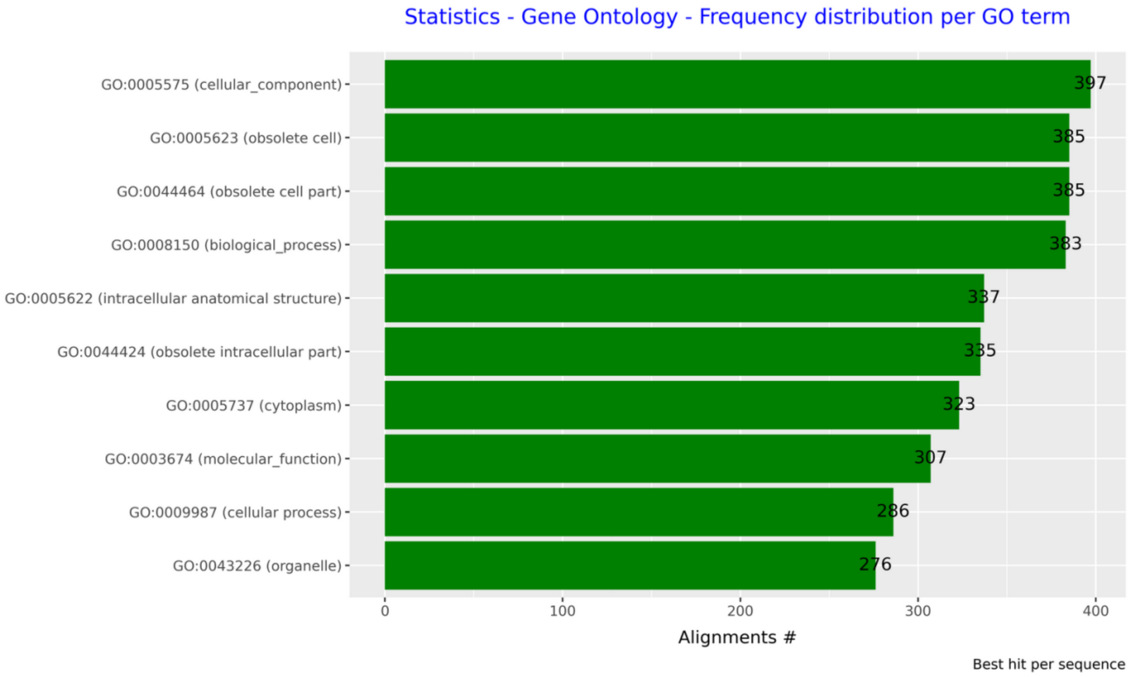


Figure 30.Plot *Statistics - Gene Ontology - Frequency distribution per GO term* with the top ten of GO terms.

In order to consult the frequency distribution per namespace, go to the menu item with this path:

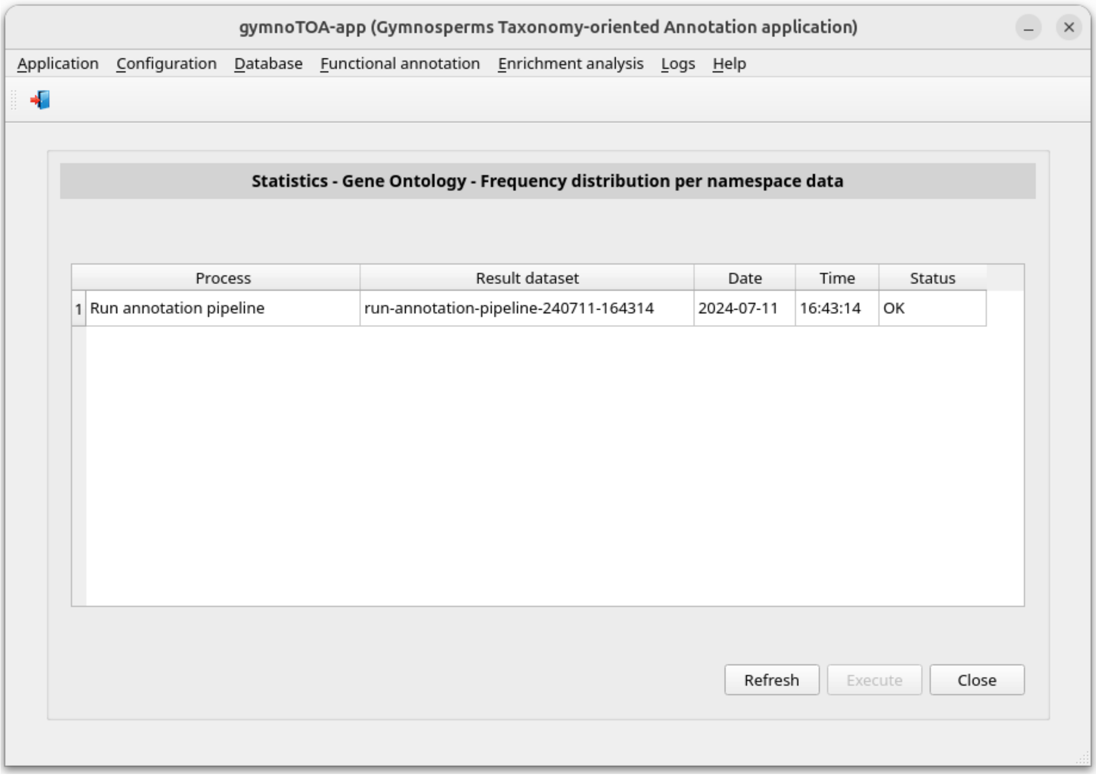
> *Main menu > Functional annotation > Statistics > Gene Ontology > Frequency distribution per namespace data*

In the new window (Figure 31), double-click on the annotation process or select with a click on it and press the *Execute* button. Pop-up window will appear with data of frequency distribution per namespace (Figure 32).

We view the plot corresponding to this data selecting the menu item:

> *Main menu > Functional annotation > Statistics > Gene Ontology > Frequency distribution per namespaces data*

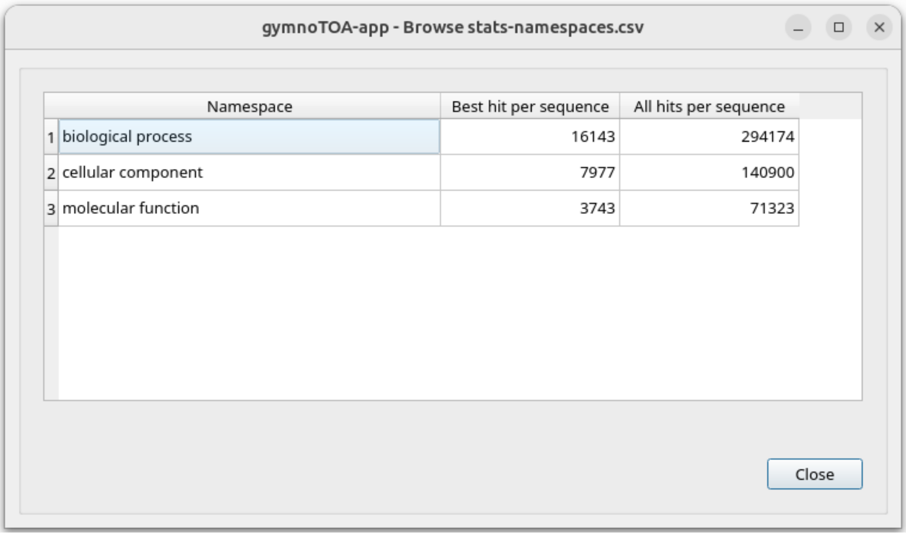
In the window shown below (Figure 33), you can modify the default values of the image name and its format and resolution. After, double-click on the annotation process or select with a click on it and press the *Execute* button. Pop-up window will appear with the plot of frequency distribution per namespace (Figure 34). The corresponding file is saved in the subdirectory of …\gymnoTOA-results\run corresponding to the script run.



Statistics - Gene Ontology - Frequency distribution per namespace data

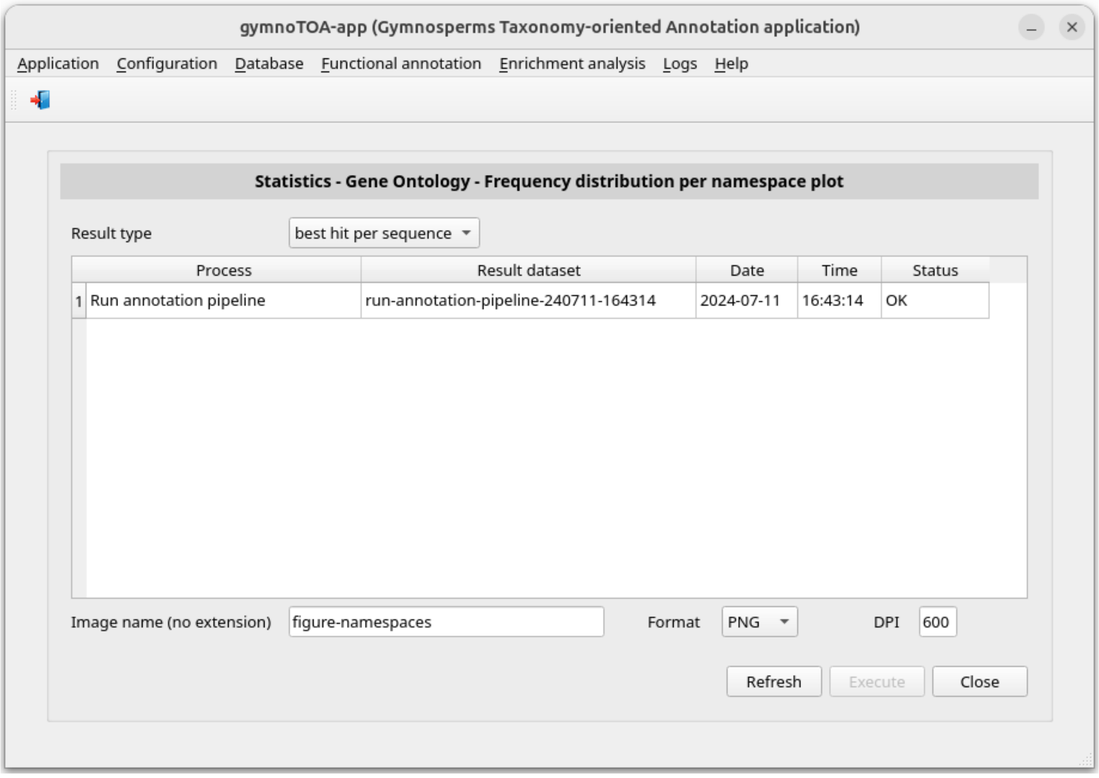
	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Figure 31. Window *Statistics - Gene Ontology - Frequency distribution per namespace data* showing annotation pipelines finished.



	Namespace	Best hit per sequence	All hits per sequence
1	biological process	16143	294174
2	cellular component	7977	140900
3	molecular function	3743	71323

Figure 32. Popup window showing the complete list of frequency distribution per namespace performed by the process *run-annotation-pipeline-240711-164314*.



Statistics - Gene Ontology - Frequency distribution per namespace plot

Result type: best hit per sequence

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Image name (no extension): figure-namespaces

Format: PNG

DPI: 600

Figure 33. Window *Statistics - Gene Ontology - Frequency distribution per namespace plot* showing annotation pipelines finished.

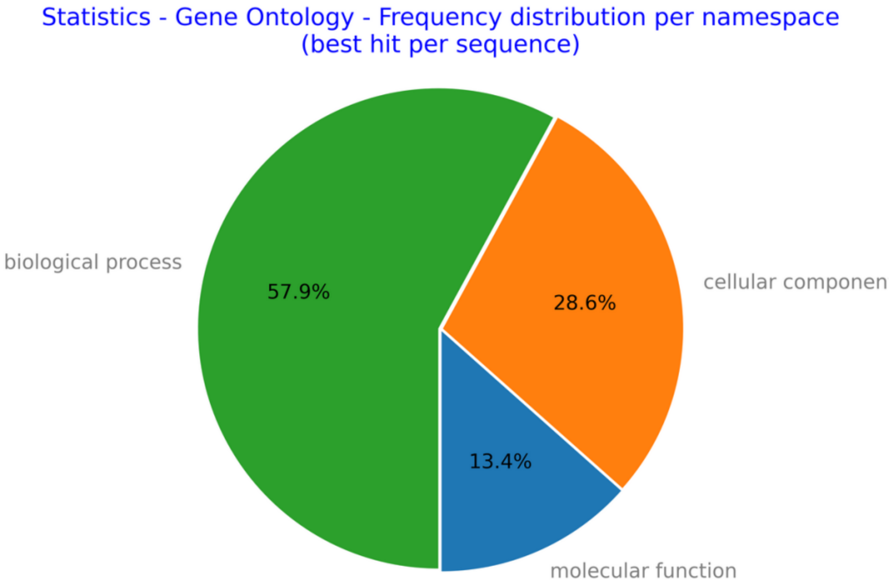


Figure 34. Plot *Statistics - Gene Ontology - Frequency distribution per namespace*.

The statistics of sequences number per GO terms number can be consulted in the menu item with this path:

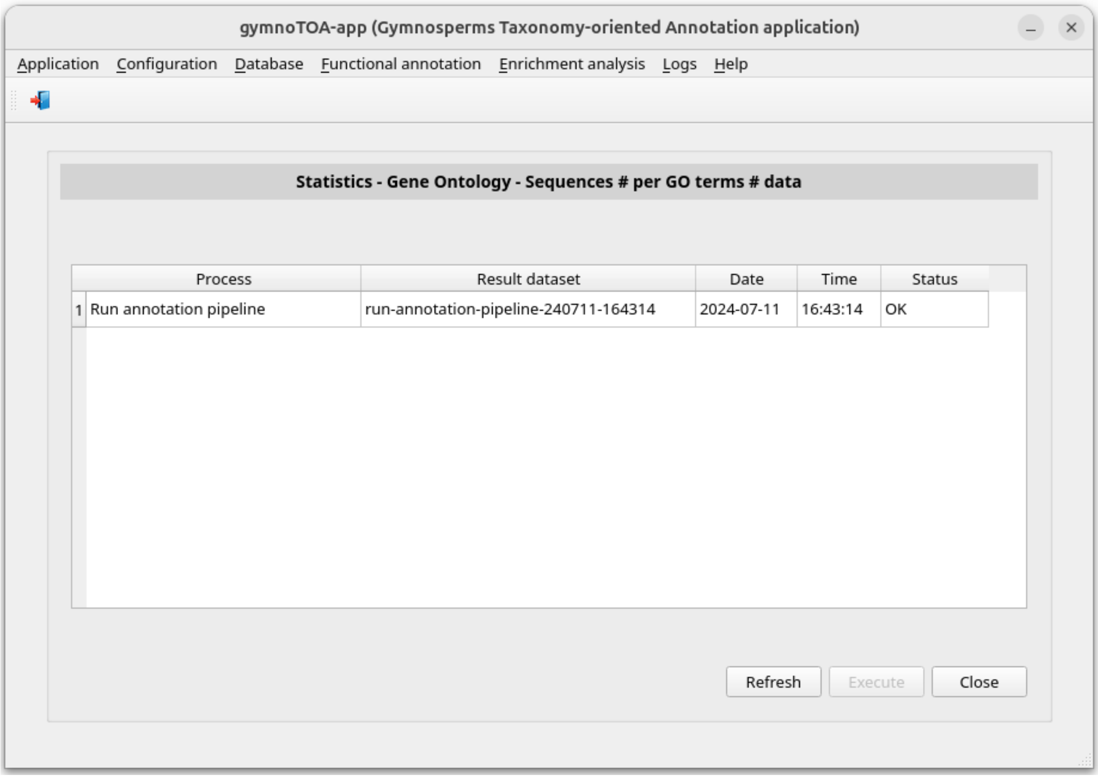
> *Main menu > Functional annotation > Statistics > Gene Ontology > Sequences # per GO terms # data*

In the new window (Figure 35), double-click on the annotation process or select with a click on it and press the *Execute* button. Pop-up window will appear with data of sequences number per GO terms number (Figure 36).

We view the plot corresponding to the top ten data selecting the menu item

> *Main menu > Functional annotation > Statistics > Gene Ontology > Sequences # per GO terms # plot*

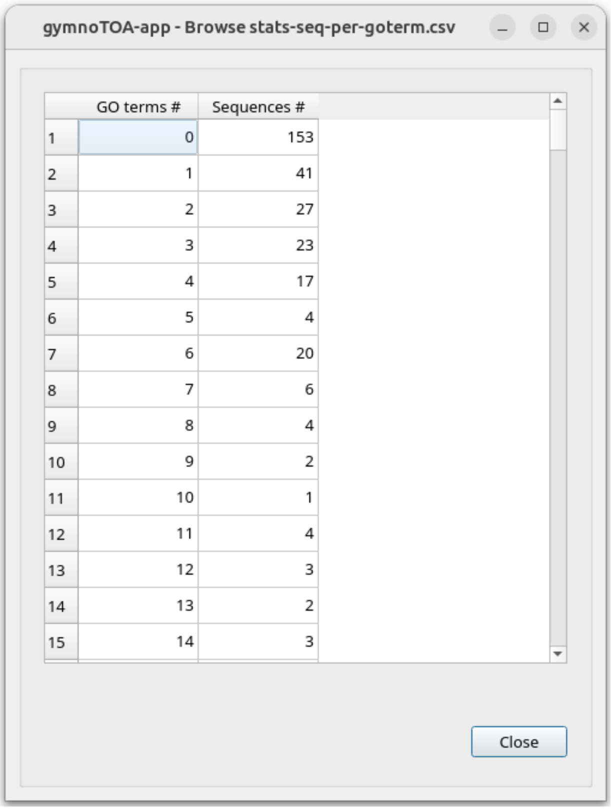
In the window shown below (Figure 37), you can modify the default values of the file name and its format and resolution. After, double-click on the annotation process or select with a click on it and press the *Execute* button. Pop-up window will appear with the plot of sequences number per GO terms number (Figure 38). The corresponding file is saved in the subdirectory of …\gymnoTOA-results\run corresponding to the script run.



Statistics - Gene Ontology - Sequences # per GO terms # data

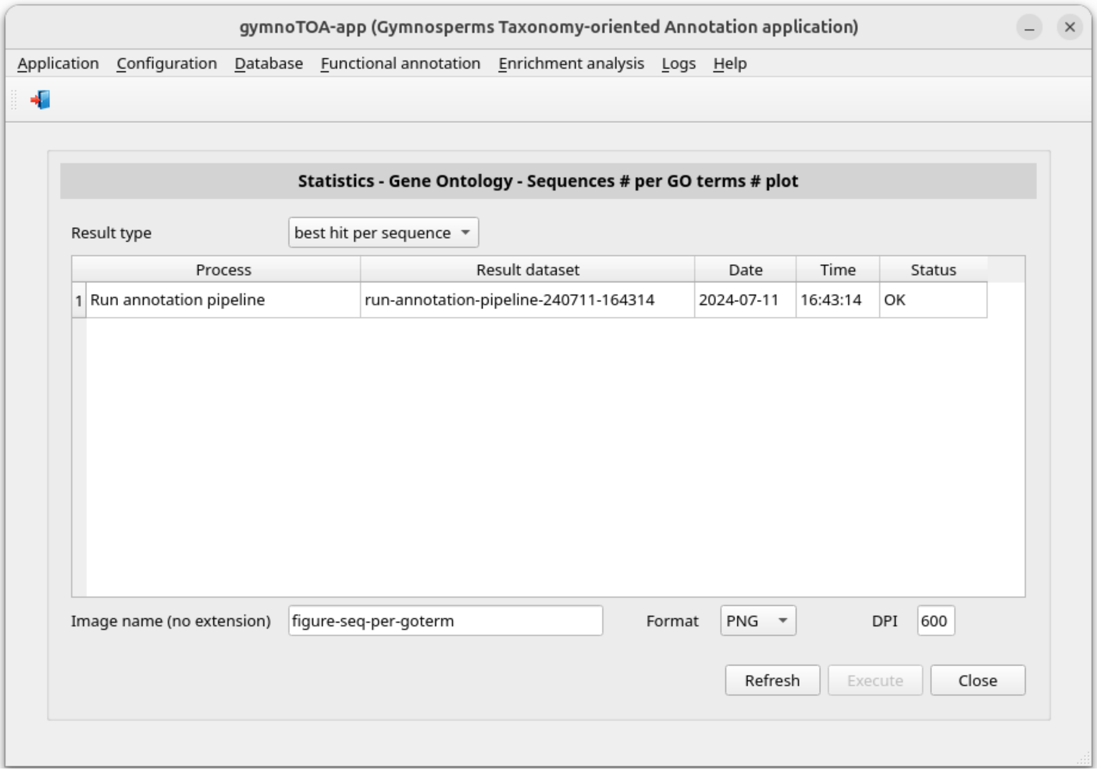
	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Figure 35. Window *Statistics - Gene Ontology – Sequences # per Go terms # data* showing annotation pipelines finished.



	GO terms #	Sequences #
1	0	153
2	1	41
3	2	27
4	3	23
5	4	17
6	5	4
7	6	20
8	7	6
9	8	4
10	9	2
11	10	1
12	11	4
13	12	3
14	13	2
15	14	3

Figure 36. Popup window showing the complete list of GO terms number per sequences number performed by the process *run-annotation-pipeline-240711-164314*.



gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application Configuration Database Functional annotation Enrichment analysis Logs Help
Statistics - Gene Ontology - Sequences # per GO terms # plot

Result type: best hit per sequence

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Image name (no extension): figure-seq-per-goterm

Format: PNG

DPI: 600

Figure 37. Window *Statistics - Gene Ontology - Sequences # per Go terms # plot* showing annotation pipelines finished.

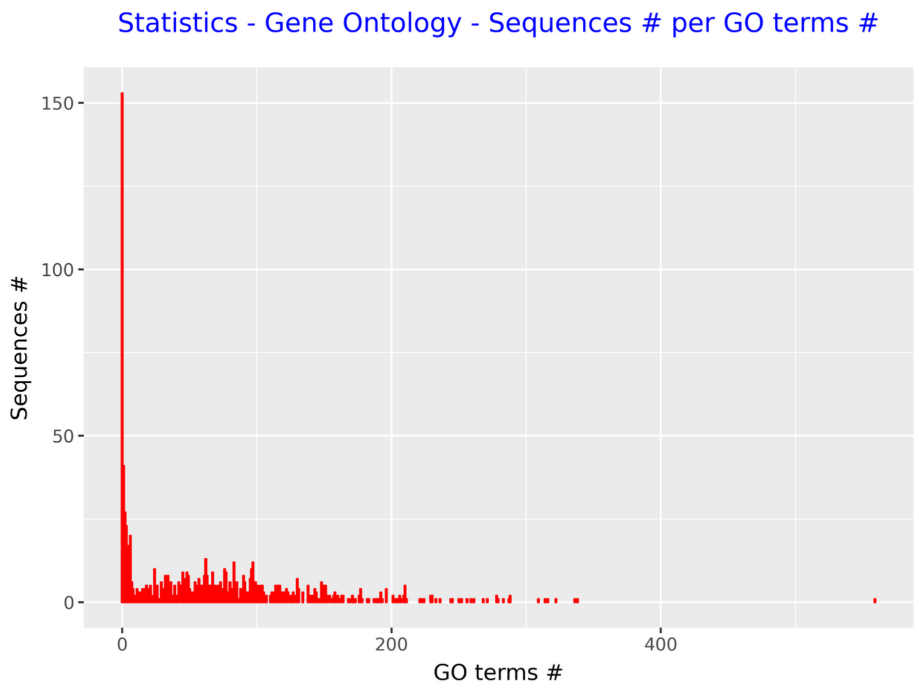


Figure 38. Plot *Statistics - Gene Ontology – Sequences # per GO terms #*.

Enrichment analysis

Run an enrichment analysis

To perform the enrichment analysis (Figure 3) with data of an annotation process we are going to select the menu item with this path:

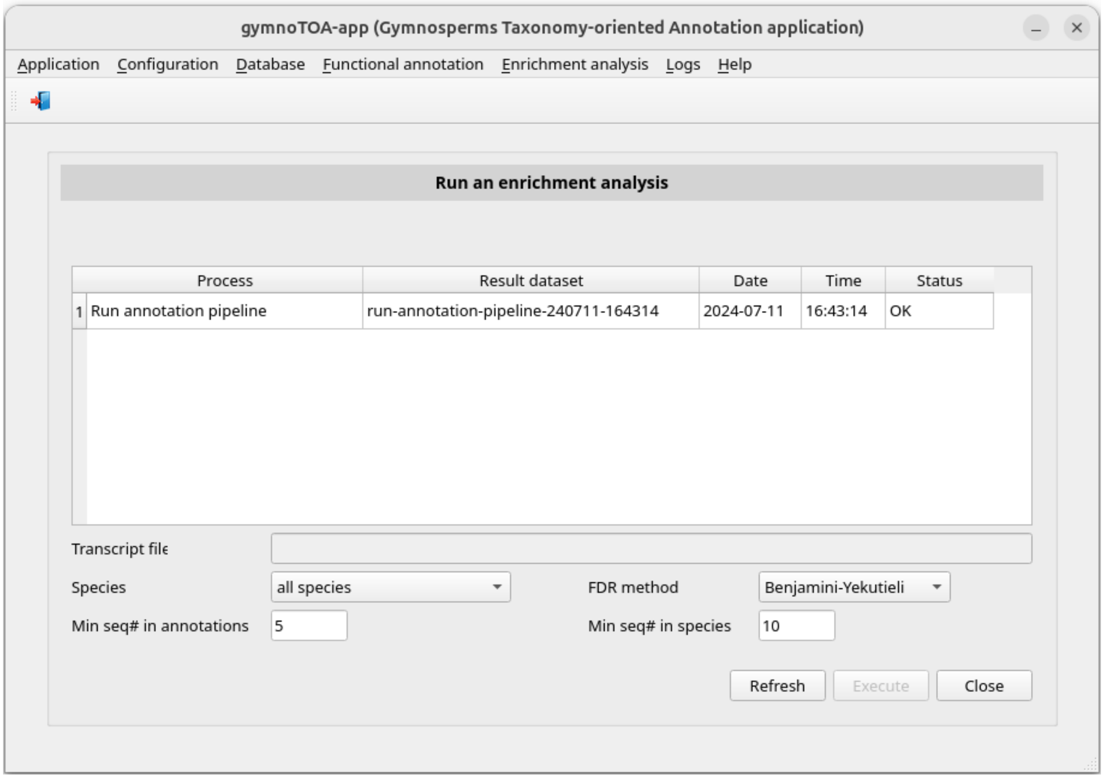
Main menu > Enrichment analysis > Run analysis

The window *Run an enrichment analysis* appears (Figure 39). Then select the annotation pipeline to be studied, in our example *run-annotation-pipeline-240711-164314*. The window will be updated showing the transcript file of the annotation process (Figure 40). Then you can modify the default values by choosing **all species** or a specific species, changing the FDR method (**Benjamini-Hochberg** or **Benjamini-Yekutieli**), and the minimum sequence number of annotations and species to be considered. Below, click on the push button *Execute*.

You can check the process status and when it has been completed reviewing its log. So, click the following menu item:

Main menu > Logs > Result logs

Select **run** in the *Process type* combo-box. Then the window is updated (Figure 41).


gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application Configuration Database Functional annotation Enrichment analysis Logs Help
Run an enrichment analysis
Refresh
Execute
Close

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Transcript file

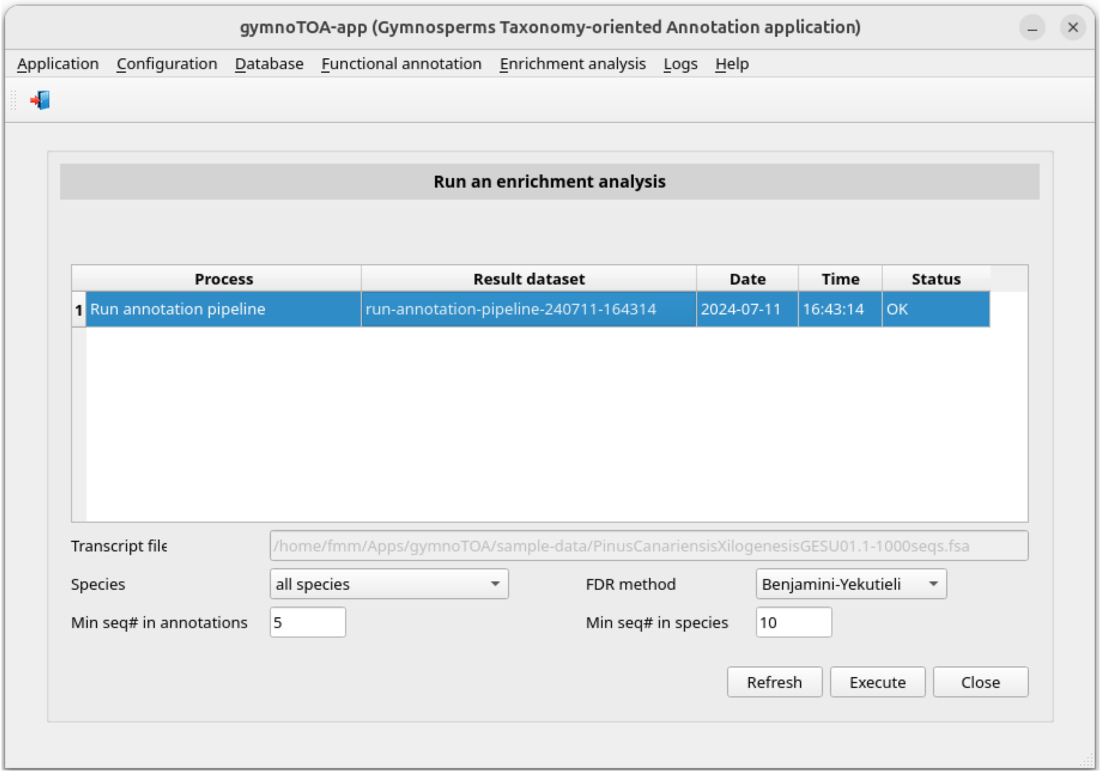
Species: all species

FDR method: Benjamini-Yekutieli

Min seq# in annotations: 5

Min seq# in species: 10

Figure 39. Initial appearance of the window *Run an enrichment analysis*.


gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application Configuration Database Functional annotation Enrichment analysis Logs Help
Run an enrichment analysis
Refresh
Execute
Close

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK

Transcript file: /home/fmm/Apps/gymnoTOA/sample-data/PinusCanariensisXilogenesisGESU01.1-1000seqs.fsa

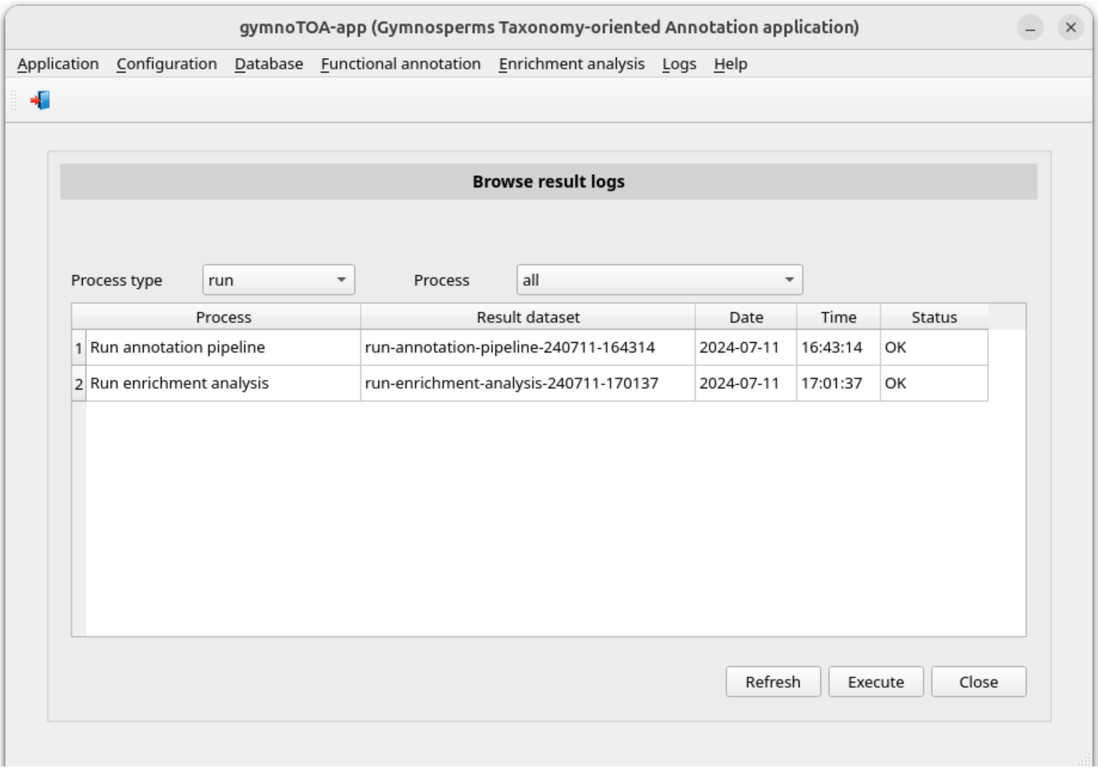
Species: all species

FDR method: Benjamini-Yekutieli

Min seq# in annotations: 5

Min seq# in species: 10

Figure 40. Window *Run an enrichment analysis* once the annotation pipeline has been selected.



Process type: run

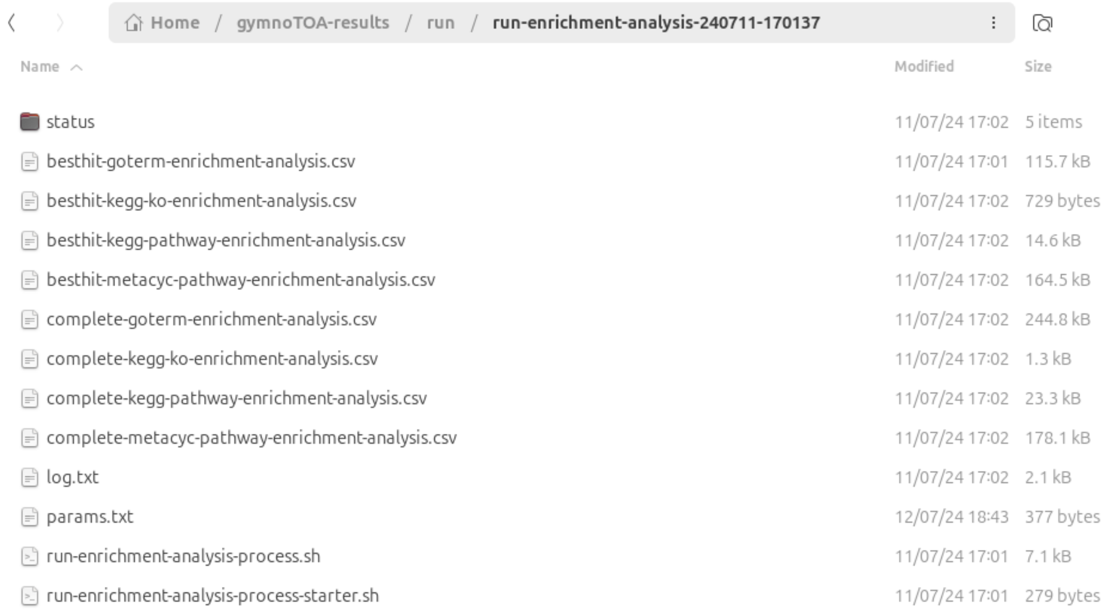
Process: all

	Process	Result dataset	Date	Time	Status
1	Run annotation pipeline	run-annotation-pipeline-240711-164314	2024-07-11	16:43:14	OK
2	Run enrichment analysis	run-enrichment-analysis-240711-170137	2024-07-11	17:01:37	OK

Figure 41. Window *Browse results logs* showing the annotation process and enrichment analysis finished.

When the process has finished, we could go the subdirectory of *…\gymnoTOA-app-results\run* corresponding to the script run, e.g. *run-enrichment_analysis-240711-170137*, and consult the generated files (Figure 42). Some of these files are:

- Analysis using the file that contains the annotations of the subject sequence identification with best hit yielded by blast programs for every query sequence identification:
  - *besthit-goterm-enrichment-analysis.csv*: enrichment of GO terms
  - *besthit-kegg-ko-enrichment-analysis.csv*: enrichment of KEGG KOs
  - *besthit-kegg-pathway-enrichment-analysis.csv*: enrichment of KEGG pathways
  - *besthit-metacyc-pathway-enrichment-analysis.csv*: enrichment of MetaCyc pathways
- Analysis using the file that contains all annotations yielded by blast programs for every query sequence identification.
  - *complete-goterm-enrichment-analysis.csv*: enrichment of GO terms
  - *complete-kegg-ko-enrichment-analysis.csv*: enrichment of KEGG KOs
  - *complete-kegg-pathway-enrichment-analysis.csv*: enrichment of KEGG pathways
  - *complete-metacyc-pathway-enrichment-analysis.csv*: enrichment of MetaCyc pathways



Home / gymnoTOA-results / run / run-enrichment-analysis-240711-170137

Name	Modified	Size
status	11/07/24 17:02	5 items
besthit-goterm-enrichment-analysis.csv	11/07/24 17:01	115.7 kB
besthit-kegg-ko-enrichment-analysis.csv	11/07/24 17:02	729 bytes
besthit-kegg-pathway-enrichment-analysis.csv	11/07/24 17:02	14.6 kB
besthit-metacyc-pathway-enrichment-analysis.csv	11/07/24 17:02	164.5 kB
complete-goterm-enrichment-analysis.csv	11/07/24 17:02	244.8 kB
complete-kegg-ko-enrichment-analysis.csv	11/07/24 17:02	1.3 kB
complete-kegg-pathway-enrichment-analysis.csv	11/07/24 17:02	23.3 kB
complete-metacyc-pathway-enrichment-analysis.csv	11/07/24 17:02	178.1 kB
log.txt	11/07/24 17:02	2.1 kB
params.txt	12/07/24 18:43	377 bytes
run-enrichment-analysis-process.sh	11/07/24 17:01	7.1 kB
run-enrichment-analysis-process-starter.sh	11/07/24 17:01	279 bytes

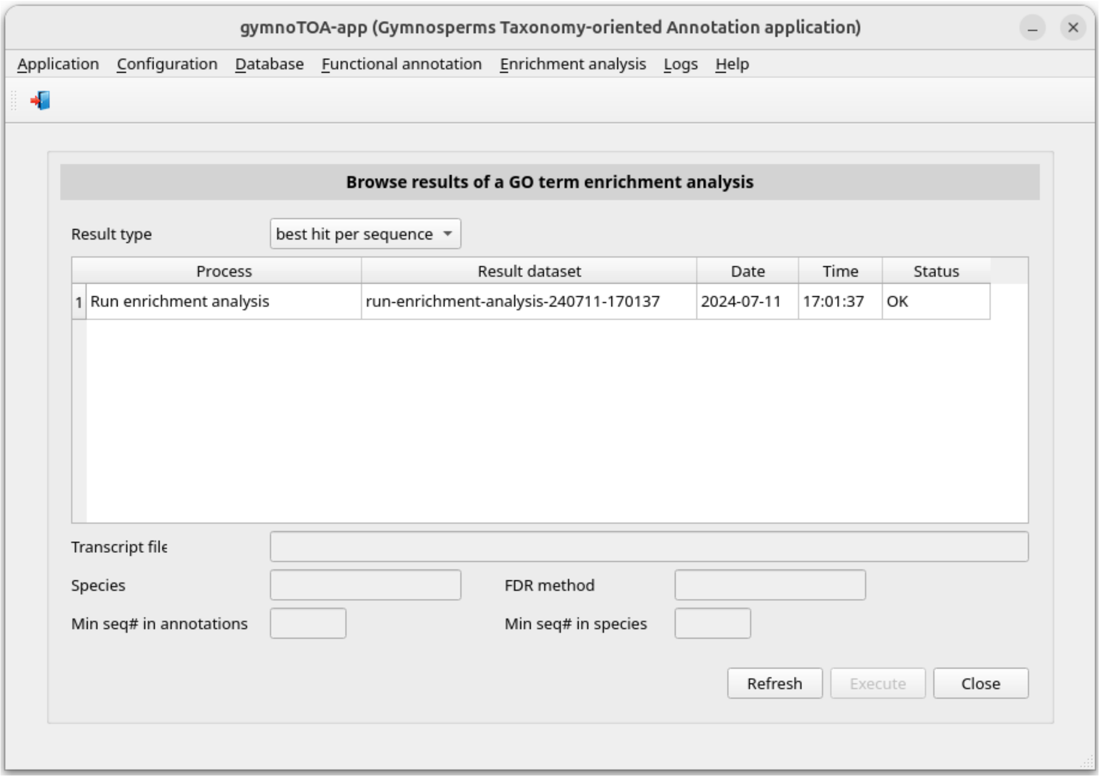
Figure 42. Folders and files in *…/gymnoTOA-app-results/run/run-enrichment_analysis-240711-170137* yielded by the enrichment analysis.

Browse results of the enrichment analysis

If you edit the files **-enrichment-analysis*, you will see the result of the enrichment analysis. You can browse them using GYMNOTOA-APP. For example, if you want to browse the enrichment analysis of GO terms, select the menu item:

Main menu > Enrichment analysis > Browse results > GO enrichment analysis

Below, the window (Figure 43) allows you to choose **best hit per sequence** (to see functional-annotations-besthit.csv) or **all hits per sequence** (to see functional-annotations-complete.csv) in the combo-box Result type. Then, select the enrichment analysis process that you are interested in consulting clicking on the corresponding row, in our example *run-enrichment-analysis -240711-170137*. The parameters of the process will be shown (Figure 44). Finally, click on the push button *Execute*. A popup window will appear showing the enrichment analysis data (Figure 45).


gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application Configuration Database Functional annotation Enrichment analysis Logs Help
Refresh
Execute
Close

Browse results of a GO term enrichment analysis

Result type: best hit per sequence

	Process	Result dataset	Date	Time	Status
1	Run enrichment analysis	run-enrichment-analysis-240711-170137	2024-07-11	17:01:37	OK

Transcript file

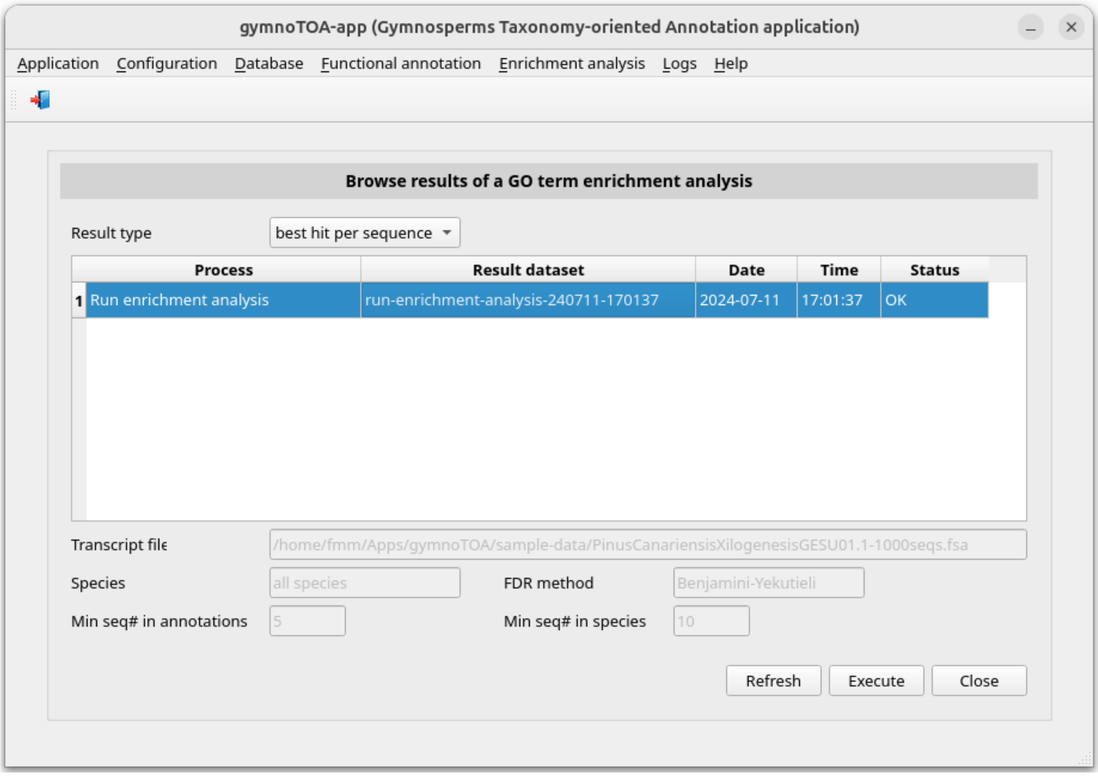
Species

FDR method

Min seq# in annotations

Min seq# in species

Figure 43. Window *Browse results of a GO term enrichment analysis* showing enrichment analysis finished.


gymnoTOA-app (Gymnosperms Taxonomy-oriented Annotation application)
Application Configuration Database Functional annotation Enrichment analysis Logs Help
Refresh
Execute
Close

Browse results of a GO term enrichment analysis

Result type: best hit per sequence

	Process	Result dataset	Date	Time	Status
1	Run enrichment analysis	run-enrichment-analysis-240711-170137	2024-07-11	17:01:37	OK

Transcript file: /home/fmm/Apps/gymnoTOA/sample-data/PinusCanariensisXilogenesisGESU01.1-1000seqs.fsa

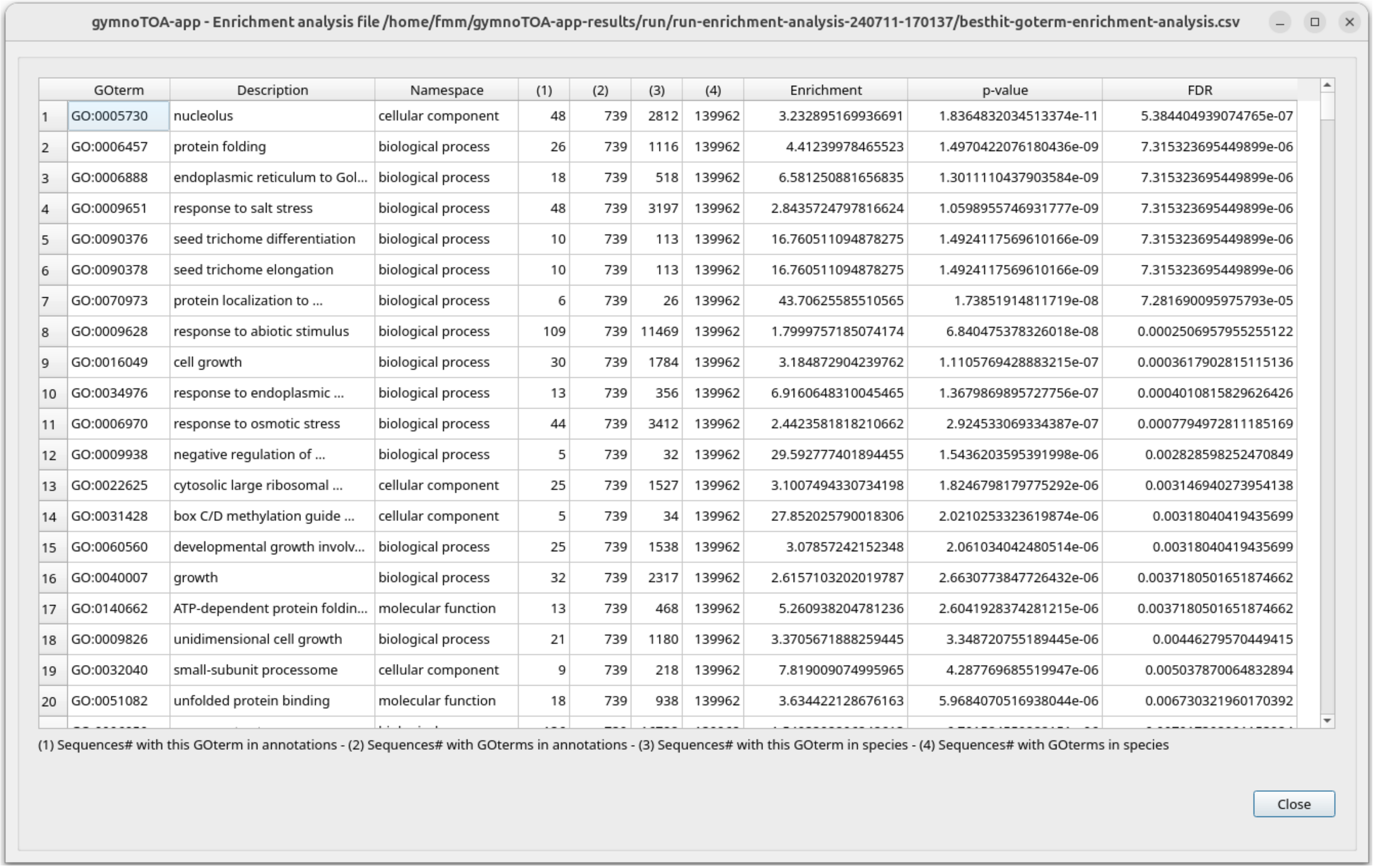
Species: all species

FDR method: Benjamini-Yekutieli

Min seq# in annotations: 5

Min seq# in species: 10

Figure 44. Window *Browse results of a GO term enrichment analysis* after *run-enrichment-analysis -240711-170137* was selected.



gymnoTOA-app - Enrichment analysis file /home/fmm/gymnoTOA-app-results/run/run-enrichment-analysis-240711-170137/besthit-goterm-enrichment-analysis.csv

	GOterm	Description	Namespace	(1)	(2)	(3)	(4)	Enrichment	p-value	FDR
1	GO:0005730	nucleolus	cellular component	48	739	2812	139962	3.232895169936691	1.8364832034513374e-11	5.384404939074765e-07
2	GO:0006457	protein folding	biological process	26	739	1116	139962	4.41239978465523	1.4970422076180436e-09	7.315323695449899e-06
3	GO:0006888	endoplasmic reticulum to Gol...	biological process	18	739	518	139962	6.581250881656835	1.3011110437903584e-09	7.315323695449899e-06
4	GO:0009651	response to salt stress	biological process	48	739	3197	139962	2.8435724797816624	1.0598955746931777e-09	7.315323695449899e-06
5	GO:0090376	seed trichome differentiation	biological process	10	739	113	139962	16.760511094878275	1.4924117569610166e-09	7.315323695449899e-06
6	GO:0090378	seed trichome elongation	biological process	10	739	113	139962	16.760511094878275	1.4924117569610166e-09	7.315323695449899e-06
7	GO:0070973	protein localization to ...	biological process	6	739	26	139962	43.70625585510565	1.73851914811719e-08	7.281690095975793e-05
8	GO:0009628	response to abiotic stimulus	biological process	109	739	11469	139962	1.7999757185074174	6.840475378326018e-08	0.0002506957955255122
9	GO:0016049	cell growth	biological process	30	739	1784	139962	3.184872904239762	1.1105769428883215e-07	0.0003617902815115136
10	GO:0034976	response to endoplasmic ...	biological process	13	739	356	139962	6.9160648310045465	1.3679869895727756e-07	0.0004010815829626426
11	GO:0006970	response to osmotic stress	biological process	44	739	3412	139962	2.4423581818210662	2.924533069334387e-07	0.0007794972811185169
12	GO:0009938	negative regulation of ...	biological process	5	739	32	139962	29.592777401894455	1.5436203595391998e-06	0.002828598252470849
13	GO:0022625	cytosolic large ribosomal ...	cellular component	25	739	1527	139962	3.1007494330734198	1.8246798179775292e-06	0.003146940273954138
14	GO:0031428	box C/D methylation guide ...	cellular component	5	739	34	139962	27.852025790018306	2.0210253323619874e-06	0.00318040419435699
15	GO:0060560	developmental growth involv...	biological process	25	739	1538	139962	3.07857242152348	2.061034042480514e-06	0.00318040419435699
16	GO:0040007	growth	biological process	32	739	2317	139962	2.6157103202019787	2.6630773847726432e-06	0.0037180501651874662
17	GO:0140662	ATP-dependent protein foldin...	molecular function	13	739	468	139962	5.260938204781236	2.6041928374281215e-06	0.0037180501651874662
18	GO:0009826	unidimensional cell growth	biological process	21	739	1180	139962	3.3705671888259445	3.348720755189445e-06	0.00446279570449415
19	GO:0032040	small-subunit processome	cellular component	9	739	218	139962	7.819009074995965	4.287769685519947e-06	0.005037870064832894
20	GO:0051082	unfolded protein binding	molecular function	18	739	938	139962	3.634422128676163	5.9684070516938044e-06	0.006730321960170392

(1) Sequences# with this GOterm in annotations - (2) Sequences# with GOterms in annotations - (3) Sequences# with this GOterm in species - (4) Sequences# with GOterms in species

Figure 45. Popup window showing the analysis performed by the process *run-enrichment-analysis -240711-170137*.

Functional annotation and enrichment analysis on Linux servers

GYMNOTOA-APP is a desktop application designed with a user-friendly front-end where researchers can perform functional annotations and enrichment analysis as well as browse their results with a few clicks. Nevertheless GYMNOTOA-APP software includes two bash scripts to run the functional annotation and the enrichment analysis processes from other Bash scripts on a Linux server without having to enter the GYMNOTOA-APP graphical environment.

Below are the steps for executing these processes in an Ubuntu Server using some locations ($HOME/Apps/Miniforge3, $HOME/Apps/gymnoTOA-app, $HOME/BioData/gymnoTOA-db) that must be adapted to right locations on the server where the processes will be run. Also, the environments names (gymnotoa, gymnotoa-blast, gymnotoa-codan and gymnotoa-diamond) could be adapted. These locations and environments names are in red typeface.

GYMNOTOA-APP installation

The unzip command is used to decompress some ZIP files. To install this command, if necessary, type the following command in a terminal:

```
$ sudo apt install unzip
```

Now, we will download the GYMNOTOA-APP software in the proper location, e.g. $HOME/Apps/gymnoTOA-app, and decompress the ZIP file using the following command:

```
$ mkdir -p $HOME/Apps

$ cd $HOME/Apps

$ wget --output-document main.zip https://github.com/GGFHF/gymnoTOA-
app/archive/refs/heads/main.zip

$ unzip main.zip

$ mv gymnoTOA-app-main gymnoTOA-app

$ rm main.zip
```

We will have the directory gymnoTOA-app in $HOME/Apps. Then, the execution permissions of the programs must be set by using these commands:

```
$ find $HOME/Apps/gymnoTOA-app -type f \( -name "*.py" -o -name "*.sh" \)
-exec chmod +x {} \;
```

GYMNOTOA-DB download

To download and decompress in the directory $HOME/BioData/gymnoTOA-db, we will type the following commands:

```
$ mkdir -p $HOME/BioData/gymnoTOA-db

$ cd $HOME/BioData

$ wget --output-document gymnoTOA-db.zip
https://drive.upm.es/s/LTssGhCHe1Wh9Os/download?files=gymnoTOA-db.zip

$ unzip gymnoTOA-db.zip

$ rm gymnoTOA-db.zip
```

Conda and bioinformatics software installation

Conda is an open-source package management and environment management tool that allows you to install, manage, and configure software packages and dependencies, particularly for Python and data science projects. We will use Miniforge3 (https://github.com/conda-forge/miniforge) that is a minimal installer for Conda that uses the conda-forge channel by default (https://conda-forge.org/), but you could use other Conda installer.

The following are the commands for the installation of Miniforge3 in the right location, e.g. $HOME/Apps/Miniforge3, if there is no previous installation:

```
$ mkdir -p $HOME/Apps

$ cd $HOME/Apps

$ wget "https://github.com/conda-
forge/miniforge/releases/latest/download/Miniforge3-$(uname)-$(uname -
m).sh"

$ bash Miniforge3-$(uname)-$(uname -m).sh -b -p $HOME/Apps/Miniforge3

$ rm Miniforge3-$(uname)-$(uname -m).sh

$ $HOME/Apps/Miniforge3/condabin/conda init bash
```

Close the terminal and open a new terminal. The Conda environment base is now active. Then type the following commands to configure the download environment:

```
$ conda config --add channels bioconda

$ conda config --add channels conda-forge
$ conda config --set channel_priority strict
```

Now we will create the gymnotoa environment used to run gymnoTOA-app programs typing the following command:

```
$ conda env create -f $HOME/Apps/gymnoTOA-app/yml/gymnotoa.yml
```

Finally, we will create the bioinformatics software environments corresponding to BLAST+, CodAn and DIAMOND. We will use three environments to avoid dependencies problems. Use the following command to create the BLAST+ environment (gymnotoa-blast):

```
$ conda create --yes --name gymnotoa-blast blast
```

In order to create the CodAn environment (gymnotoa-codan) and to download the plant models, type the following commands:

```
$ conda create --yes --name gymnotoa-codan codan

$ conda activate gymnotoa-codan

$ MODELS_DIR=`echo $CONDA_PREFIX`/models

$ mkdir -p $MODELS_DIR

$ wget --output-document $MODELS_DIR/PLANTS_full.zip
https://github.com/pedronachtigall/CodAn/raw/master/models/PLANTS_full
.zip

$ unzip -o -d $MODELS_DIR $MODELS_DIR/PLANTS_full.zip

$ rm $MODELS_DIR/PLANTS_full.zip

$ wget --output-document $MODELS_DIR/PLANTS_partial.zip
https://github.com/pedronachtigall/CodAn/raw/master/models/PLANTS_part
ial.zip

$ unzip -o -d $MODELS_DIR $MODELS_DIR/PLANTS_partial.zip

$ rm $MODELS_DIR/PLANTS_partial.zip

$ conda deactivate
```

The DIAMOND environment (gymnotoa-dimanond) is created with the command:

```
$ conda create --yes --name gymnotoa-diamond diamond
```

Running functional annotation and enrichment analysis processes

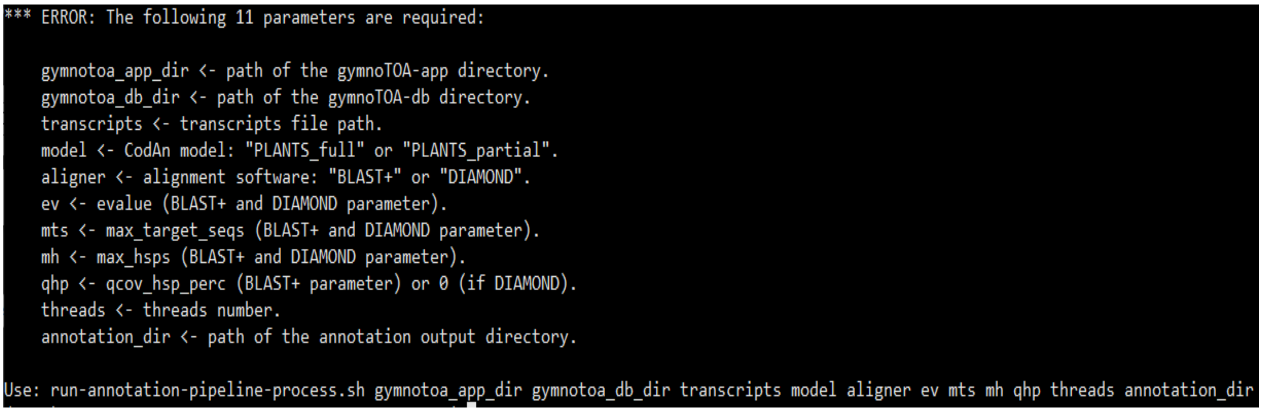
Two bash scripts are included in the folder pipelines of the GYMNOTOA-APP software package: **run-annotation-pipeline-process.sh** and **run-enrichment-analysis-process.sh**. These scripts can be used to run functional annotation and enrichment analysis processes from other Bash scripts without having to enter the GYMNOTOA-APP graphical environment once the above steps have been carried out. There is another script in the folder, **test-gymnotoa-processes.sh**, which is an example of running the annotation and enrichment processes from another Bash script.

The script **run-annotation-pipeline-process.sh** needs 12 input parameters detailed in the following table:

Table: **run-annotation-pipeline-process.sh** parameters

Parameter	Comment
gymnotoa_app_dir	path of the GYMNOTOA-APP directory
gymnotoa_db_dir	path of the GYMNOTOA-DB directory
fasta_type	FASTA file type: "TRANSCRIPTS" or "PROTEINS"
fasta_file	transcripts file path
model	CodAn model: "PLANTS_full" or "PLANTS_partial"
aligner	alignment software: "BLAST+" or "DIAMOND"
ev	evalue (BLAST+ and DIAMOND parameter)
mts	max_target_seqs (BLAST+ and DIAMOND parameter)
mh	max_hsps (BLAST+ and DIAMOND parameter)
qhp	qcov_hsp_perc (BLAST+ parameter) or 0 (if DIAMOND)
threads	threads number
annotation_dir	path of the annotation output directory

If **run-annotation-pipeline-process.sh** is called with a number other than 11 parameters, a warning message is displayed (Figure 46):



```
*** ERROR: The following 11 parameters are required:

    gymnotoa_app_dir <- path of the gymnoTOA-app directory.
    gymnotoa_db_dir <- path of the gymnoTOA-db directory.
    transcripts <- transcripts file path.
    model <- CodAn model: "PLANTS_full" or "PLANTS_partial".
    aligner <- alignment software: "BLAST+" or "DIAMOND".
    ev <- evalue (BLAST+ and DIAMOND parameter).
    mts <- max_target_seqs (BLAST+ and DIAMOND parameter).
    mh <- max_hsps (BLAST+ and DIAMOND parameter).
    qhp <- qcov_hsp_perc (BLAST+ parameter) or 0 (if DIAMOND).
    threads <- threads number.
    annotation_dir <- path of the annotation output directory.

Use: run-annotation-pipeline-process.sh gymnotoa_app_dir gymnotoa_db_dir transcripts model aligner ev mts mh qhp threads annotation_dir
```

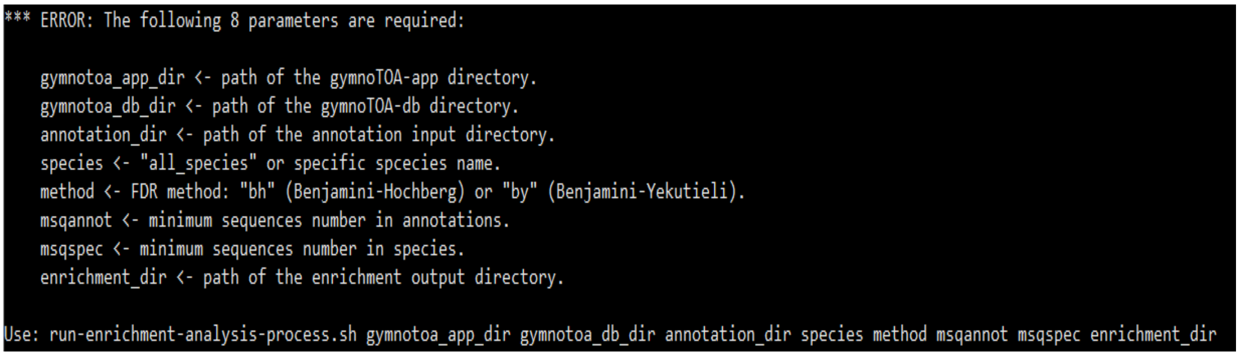
Figure 46. Warning message displayed when the parameters are not correctly passed to the script **run-annotation-pipeline-process.sh**.

The script **run-enrichment-analysis-process.sh** needs 8 input parameters detailed in the following table:

Table: **run-enrichment-analysis-process.sh** parameters

Parameter	Comment
gymnotoa_app_dir	path of the gymnoTOA-app directory
gymnotoa_db_dir	path of the gymnoTOA-db directory
annotation_dir	path of the annotation output directory
species	"all_species" or specific spcecies name
method	FDR method: "bh" (Benjamini-Hochberg) or "by" (Benjamini-Yekutieli)
msqannot	minimum sequences number in annotations
msqspec	minimum sequences number in species
enrichment_dir	path of the enrichment output directory

If **run-enrichment-analysis-process.sh** is called with a number other than 8 parameters, the script displays a warning message (Figure 47).



```
*** ERROR: The following 8 parameters are required:

    gymnotoa_app_dir <- path of the gymnoTOA-app directory.
    gymnotoa_db_dir <- path of the gymnoTOA-db directory.
    annotation_dir <- path of the annotation input directory.
    species <- "all_species" or specific spcecies name.
    method <- FDR method: "bh" (Benjamini-Hochberg) or "by" (Benjamini-Yekutieli).
    msqannot <- minimum sequences number in annotations.
    msqspec <- minimum sequences number in species.
    enrichment_dir <- path of the enrichment output directory.

Use: run-enrichment-analysis-process.sh gymnotoa_app_dir gymnotoa_db_dir annotation_dir species method msqannot msqspec enrichment_dir
```

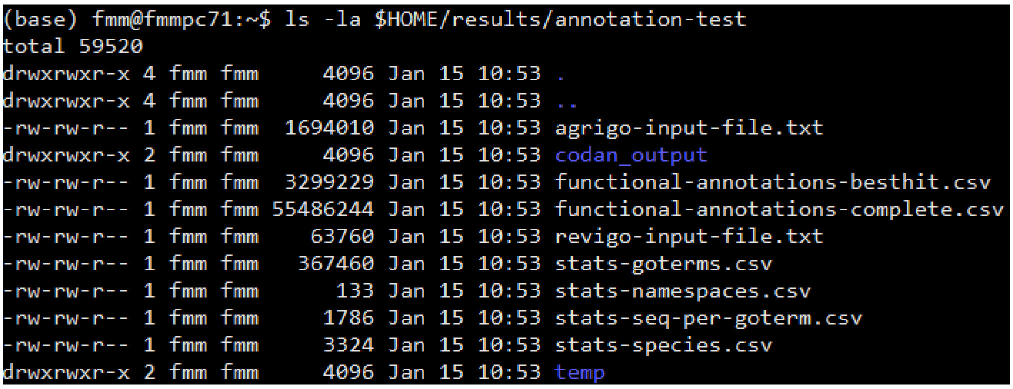
Figure 47. Warning message displayed when the parameters are not correctly passed to the script **run-enrichment-analysis-process.sh**.

No changes need to be made to the scripts **run-annotation-pipeline-process.sh** and **run-enrichment-analysis-process.sh** unless you have chosen environment names other than gymnotoa, gymnotoa-blast, gymnotoa-codan and gymnotoa-diamond.

Running the scripts **run-annotation-pipeline-process.sh** and **run-enrichment-analysis-process.sh** can be made directly from the command line by passing the values of the required parameters, although a more convenient way is to call them from another script. The script **test-gymnotoa-processes.sh** is an example of calling functional annotation and enrichment analysis scripts from other bash scripts. You can run this script as a test without any changes as long as the installation has been performed in the indicated locations. If not, simply modify the path of the affected directories. The following command will run **test-gymnotoa-processes.sh**:

```
$ $HOME/Apps/gymnoTOA-app/pipelines/test-gymnotoa-processes.sh
```

In this test the value passed to annotation_dir parameter is $HOME/results/annotation-test. Figure 48 shows the folders and files yielded in this directory when **run-annotation-pipeline-process.sh** is ended.



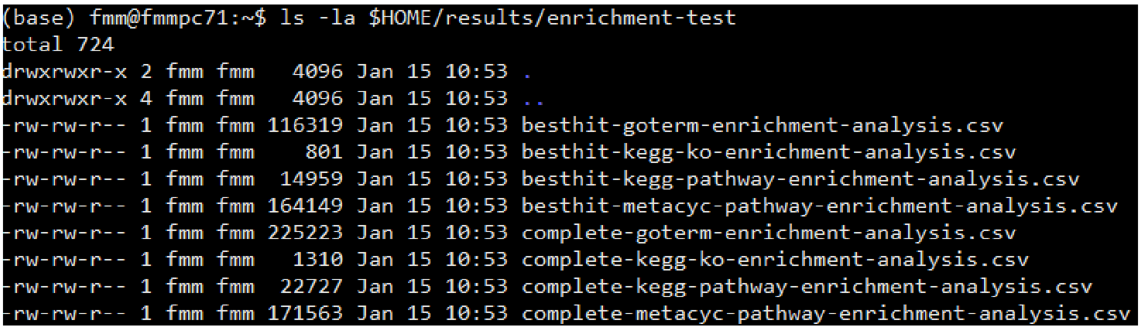
```
(base) fmm@fmmpc71:~$ ls -la $HOME/results/annotation-test
total 59520
drwxrwxr-x 4 fmm fmm     4096 Jan 15 10:53 .
drwxrwxr-x 4 fmm fmm     4096 Jan 15 10:53 ..
-rw-rw-r-- 1 fmm fmm  1694010 Jan 15 10:53 agrigo-input-file.txt
drwxrwxr-x 2 fmm fmm     4096 Jan 15 10:53 codan_output
-rw-rw-r-- 1 fmm fmm  3299229 Jan 15 10:53 functional-annotations-besthit.csv
-rw-rw-r-- 1 fmm fmm 55486244 Jan 15 10:53 functional-annotations-complete.csv
-rw-rw-r-- 1 fmm fmm    63760 Jan 15 10:53 revigo-input-file.txt
-rw-rw-r-- 1 fmm fmm   367460 Jan 15 10:53 stats-goterms.csv
-rw-rw-r-- 1 fmm fmm      133 Jan 15 10:53 stats-namespaces.csv
-rw-rw-r-- 1 fmm fmm     1786 Jan 15 10:53 stats-seq-per-goterm.csv
-rw-rw-r-- 1 fmm fmm     3324 Jan 15 10:53 stats-species.csv
drwxrwxr-x 2 fmm fmm     4096 Jan 15 10:53 temp
```

Figure 48. Folders and files yielded by **run-annotation-pipeline-process.sh**.

These files are:

- Annotation:
  - *functional-annotations-complete.csv*: It contains all annotations yielded by blast programs for every query sequence identification.
  - *functional-annotations-besthit.csv*: It contains the annotations of the subject sequence identification with best hit yielded by blast programs for every query sequence identification.
- Statistics:
  - *stats-goterms.csv*: It contains the frequency distribution per GO term.
  - *stats-namespaces.csv*: It contains the frequency distribution per namespace.
  - *stats-seq-per-goterm.csv*: It contains the sequence number per GO term number.
  - *stats-species.csv*: It contains the frequency distribution of species.
- Other applications inputs:
  - *agrigo-input-file.txt*: It contains data to be used as agriGO input.
  - *revigo-input-file.txt*: It contains data to be used as REVIGO input.

In this test the value passed to enrichment_dir parameter is $HOME/results/enrichment-test. Figure 49 shows the folders and files yielded in this directory when **run-enrichment-analysis-process.sh** is ended.



```
(base) fmm@fmmpc71:~$ ls -la $HOME/results/enrichment-test
total 724
drwxrwxr-x 2 fmm fmm   4096 Jan 15 10:53 .
drwxrwxr-x 4 fmm fmm   4096 Jan 15 10:53 ..
-rw-rw-r-- 1 fmm fmm 116319 Jan 15 10:53 besthit-goterm-enrichment-analysis.csv
-rw-rw-r-- 1 fmm fmm    801 Jan 15 10:53 besthit-kegg-ko-enrichment-analysis.csv
-rw-rw-r-- 1 fmm fmm  14959 Jan 15 10:53 besthit-kegg-pathway-enrichment-analysis.csv
-rw-rw-r-- 1 fmm fmm 164149 Jan 15 10:53 besthit-metacyc-pathway-enrichment-analysis.csv
-rw-rw-r-- 1 fmm fmm 225223 Jan 15 10:53 complete-goterm-enrichment-analysis.csv
-rw-rw-r-- 1 fmm fmm   1310 Jan 15 10:53 complete-kegg-ko-enrichment-analysis.csv
-rw-rw-r-- 1 fmm fmm  22727 Jan 15 10:53 complete-kegg-pathway-enrichment-analysis.csv
-rw-rw-r-- 1 fmm fmm 171563 Jan 15 10:53 complete-metacyc-pathway-enrichment-analysis.csv
```

Figure 49. Files yielded by the script **run-enrichment-analysis-process.sh**.

These files are:

- Analysis using the file that contains the annotations of the subject sequence identification with best hit yielded by blast programs for every query sequence identification:
    - *besthit-goterm-enrichment-analysis.csv*: enrichment of GO terms
    - *besthit-kegg-ko-enrichment-analysis.csv*: enrichment of KEGG KOs
    - *besthit-kegg-pathway-enrichment-analysis.csv*: enrichment of KEGG pathways
    - *besthit-metacyc-pathway-enrichment-analysis.csv*: enrichment of MetaCyc pathways
- Analysis using the file that contains all annotations yielded by blast programs for every query sequence identification.
    - *complete-goterm-enrichment-analysis.csv*: enrichment of GO terms
    - *complete-kegg-ko-enrichment-analysis.csv*: enrichment of KEGG KOs
    - *complete-kegg-pathway-enrichment-analysis.csv*: enrichment of KEGG pathways
    - *complete-metacyc-pathway-enrichment-analysis.csv*: enrichment of MetaCyc pathways

How to cite

If you are using GYMNOTOA-APP or GYMNOTOA-DB, you should cite the following paper:

Fernando Mora-Márquez, Mikel Hurtado & Unai López de Heredia (under review). GYMNOTOA-DB: a database and application to optimize functional annotation in gymnosperms. DOI: https://doi.org/x

Appendix A: Description of the gymnoTOA-db tables

This annex contains the description of the tables of the SQLite database of gymnoTOA-db:

Table: mmseq2_relationships

Column	Type	Index	Comment
cluster_id	TEXT	1	cluster identification
seq_id	TEXT	2	NCBI protein sequence identification
description	TEXT		description from the NCBI protein sequence
species	TEXT		species from the NCBI protein sequence

Table: interproscan_annotations

Column	Type	Index	Comment
cluster_id	TEXT	1	cluster identification
interpro_goterms	TEXT		concatenated list of GO terms from InterPro
panther_goterms	TEXT		concatenated list of GO terms from Panther
x_goterms	TEXT		concatenated list of GO terms from other sources
metacyc_pathways	TEXT		concatenated list of pathway identifications from MetaCyc
reactome_pathways	TEXT		concatenated list of pathway identifications from Reactome
x_pathways	TEXT		concatenated list of pathway identifications from other sources

Table: emapper_annotations

Column	Type	Index	Comment
cluster_id	TEXT	1	cluster identification
ortholog_seq_id	TEXT		ortholog sequence identification from eggNOG
ortholog_species	TEXT		species from eggNOG
eggnog_ogs	TEXT		OGs (Orthologous Groups) of proteins from eggNOG
cog_category	TEXT		COG (Cluster of Orthologous Genes) from eggNOG
description	TEXT		description from eggNOG
goterms	TEXT		concatenated list of GO terms from eggNOG
ec	TEXT		concatenated list of EC (Enzyme Commission) numbers
kegg_kos	TEXT		concatenated list of KO from KEGG
kegg_pathways	TEXT		concatenated list of pathway identifications from KEGG
kegg_modules	TEXT		concatenated list of module identifications from KEGG
kegg_reactions	TEXT		concatenated list of chemical reactions identifications from KEGG
kegg_rclasses	TEXT		concatenated list of reactions classification identifications from KEGG

brite	TEXT		functional hierarchy of OGs assigned to the sequence
kegg_tc	TEXT		T cell receptor (TCR) signaling pathway
cazy	TEXT		concatenated list of Carbohydrate-Active Enzymes (CAZymes)
pfams	TEXT		concatenated list of protein families from Pfam

Table: tair10_orthologs

Column	Type	Index	Comment
cluster_id	TEXT	1	cluster identification
ortholog_seq_id	TEXT		ortholog sequence identification of *A. thalina*

Table: go_ontology

Column	Type	Index	Comment
go_id	TEXT	1	GO term identification
go_name	TEXT		GO term description
namespace	TEXT		Molecular function, biological process or cellular component