AgBase Documentation

Release 1.0

Fiona M. McCarthy

Sep 06, 2023

Agbase Home

AgBase is a curated, open-source, Web-accessible resource for functional analysis of agricultural plant and animal gene products. Our long-term goal is to serve the needs of the agricultural research communities by facilitating post-genome biology for agriculture researchers and for those researchers primarily using agricultural species as biomedical models.

We use controlled vocabularies developed by the Gene Ontology (GO) Consortium to describe molecular function, biological process, and cellular component for genes and gene products in agricultural species. For more information about the AgBase database please visit our Educational Resources page or refer to our AgBase publications .

AgBase will also accept annotations from any interested party in the research communities. AgBase develops freely available tools for functional analysis, including tools for using GO. We appreciate any and all questions, comments, and suggestions. Please send us your ideas about how to make AgBase more useful for you.

CHAPTER 1

Acknowledgements

AgBase acknowledges the following groups for their help and support: The GO Consortium, especially DictyBase for providing the database schema and for technical assistance with implementation, MGI for providing training and continued support with manual curation issues and the EBI GOA Project for allowing us access to their tools and for their continued help, support and patience.

1.1 AgBase has received financial support from

- Mississippi State University
- Office of Research and Economic Development (ORED)
- Division of Agriculture, Forestry and Veterinary Medicine (DAFVM)
- Mississippi Agricultural and Forestry Experiment Station (MAFES)
- College of Veterinary Medicine
- Bagley College of Engineering
- Institute for Genomics, Biocomputing & Biotechnology (IGBB; formerly the Life Sciences and Biotechnology Institute)

1.2 Competitive Grants

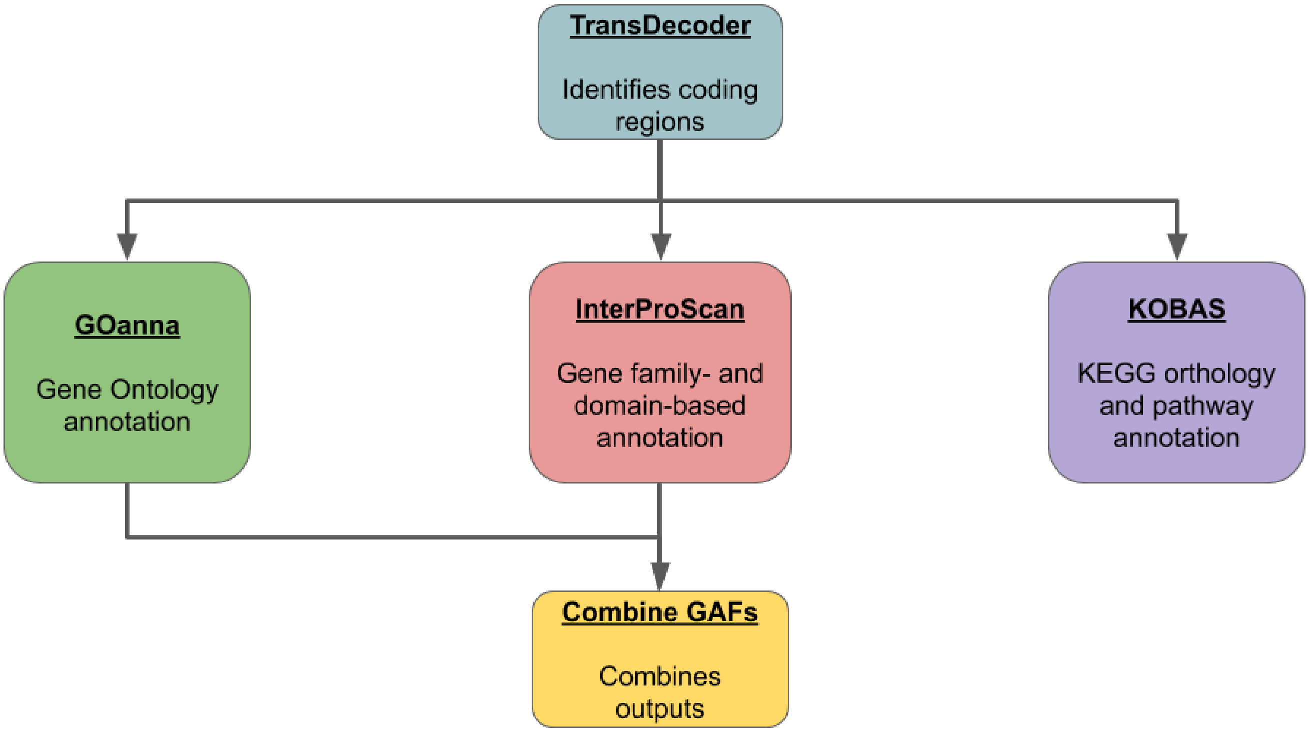
- USDA Agriculture and Food Research Initiative Competitive Grant no. 2011-67015-30332
- National Research Initiative of the USDA Cooperative State Research, Education and Extension Service, grant number 2007-35205-17941
- National Institutes of Health NIGMS project 07111084
- NSF EPSCoR award number EPS 0903787

CHAPTER 2

Contact Us

agbase@email.arizona.edu

2.1 Functional Annotation Workflow


TransDecoder
Identifies coding regions
GOanna
Gene Ontology annotation
InterProScan
Gene family- and domain-based annotation
KOBAS
KEGG orthology and pathway annotation
Combine GAFs
Combines outputs

This functional annotation workflow employs three annotation tools:

1. **GOanna:** It performs a BLAST search and transfers gene ontology (GO) annotations from BLAST matches to the query gene products.
2. **InterProScan:** InterPro is a database which integrates together predictive information about proteins' function from a number of partner resources, giving an overview of the families that a protein belongs to and the domains and sites it contains. InterProScan can also provide GO and pathway annotations.
3. **KOBAS:** It uses BLAST to annotate the input with KEGG Orthology terms and KEGG pathways

Results and analysis from the application of this functional annotation workflow to the Official gene set v3.0 protein set from *Diaphorina citri* followed by a differential expression analysis was presented at a seminar in the University of Arizona Animal and Comparative Biomedical Sciences in Fall 2020. The slides and video are available online.

Citation: Please cite the following preprint if you use annotation results from the workflow

Saha, S.; Cooksey, A.M.; Childers, A.K.; Poelchau, M.F.; McCarthy, F.M. Workflows for Rapid Functional Annotation of Diverse Arthropod Genomes. *Insects* 2021, 12, 748. https://doi.org/10.3390/insects12080748

Note: Each of these tools accepts a peptide FASTA file. For those users with nucleotide sequences some documentation has been provided for using **TransDecoder** (although other tools are also acceptable). The TransDecoder app is available through CyVerse or as a BioContainer for use on the command line.

Note: As both GOanna and InterProScan provide GO annotations, their outputs are provided in GAF format. The **'Combine GAFs'** tool can then be used to make a single GAF of GO annotations, if desired.

2.2 Intro

- GOanna performs a BLAST search, allows you to filter based on BLAST match parameters and transfers Gene Ontology (GO) functional annotations from the BLAST matches to your input genes.
- GOanna accepts a protein FASTA file as input.
- BLAST databases are created by AgBase based upon proteins that have GO available and subsetted by phyla. We recommend selecting the database most closely related to the sequence used as input.
- We strongly recommend selecting only GO annotations based on experimental evidence codes. This will ensure the best quality annotations for your data.
- The remaining parameters are standard BLAST parameters. More information on determining the best BLAST parameters for your specific data set can be found in the section below.

Results and analysis from the application of GOanna to the Official gene set v3.0 protein set from *Diaphorina citri* followed by a differential expression analysis was presented at a seminar in the University of Arizona Animal and Comparative Biomedical Sciences in Fall 2020. The slides and video are available online.

2.2.1 Where to Find GOanna

- Docker Hub
- CyVerse Discovery Environment
- AgBase

2.2.2 Getting the GOanna Databases

To run the tool you need some public data. These files are now available as gzipped files to aid downloading. The directories are best downloaded with iCommands. Once iCommands is setup you can use 'iget' to download the data.

1) agbase_database: species subset to run BLAST against (this command will download the entire directory)

```
iget -rPT /iplant/home/shared/iplantcollaborative/protein_blast_dbs/agbase_database
```

2) go_info: Uniprot GO annotations (this command will download the entire directory)

```
iget -rPT /iplant/home/shared/iplantcollaborative/protein_blast_dbs/go_info
```

Note: Each of these tools accepts a peptide FASTA file. For those users with nucleotide sequences some documentation has been provided for using **TransDecoder** (although other tools are also acceptable). The TransDecoder app is available through CyVerse or as a BioContainer for use on the command line.

2.2.3 Help and Usage Statement

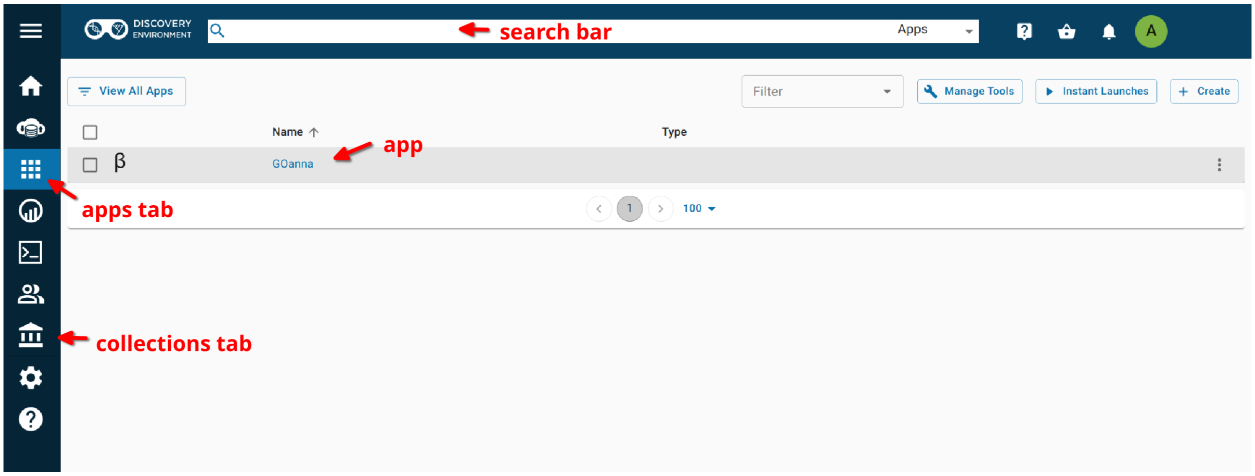
```
Options:
-a BLAST database basename ('arthropod', 'bacteria', 'bird', 'crustacean', 'fish',
↪'fungi', 'human', 'insecta',
   'invertebrates', 'mammals', 'nematode', 'plants', 'rodents' 'uniprot_sprot',
↪'uniprot_trembl', 'vertebrates'
    or 'viruses')
-c peptide fasta filename
-o output file basename
[-b transfer GO with experimental evidence only ('yes' or 'no'). Default = 'yes'.]
[-d database of query ID. If your entry contains spaces either substitute and␣
↪underscore (_) or,
    to preserve the space, use quotes around your entry. Default: 'user_input_db']
[-e Expect value (E) for saving hits. Default is 10.]
[-f Number of aligned sequences to keep. Default: 3]
[-g BLAST percent identity above which match should be kept. Default: keep all␣
↪matches.]
[-h help]
[-m BLAST percent positive identity above which match should be kept. Default: keep␣
↪all matches.]
[-s bitscore above which match should be kept. Default: keep all matches.]
[-k Maximum number of gap openings allowed for match to be kept.Default: 100]
[-l Maximum number of total gaps allowed for match to be kept. Default: 1000]
[-q Minimum query coverage per subject for match to be kept. Default: keep all␣
↪matches]
[-t Number of threads.  Default: 8]
[-u 'Assigned by' field of your GAF output file. If your entry contains spaces (eg.␣
↪firstname lastname)
    either substitute and underscore (_) or, to preserve the space, use quotes around␣
↪your entry (eg. "firstname lastname")
    Default: 'user']
[-x Taxon ID of the query species. Default: 'taxon:0000']
[-p parse_deflines. Parse query and subject bar delimited sequence identifiers]
```

2.3 GOanna on CyVerse

2.3.1 Accessing GOanna in the Discovery Environment

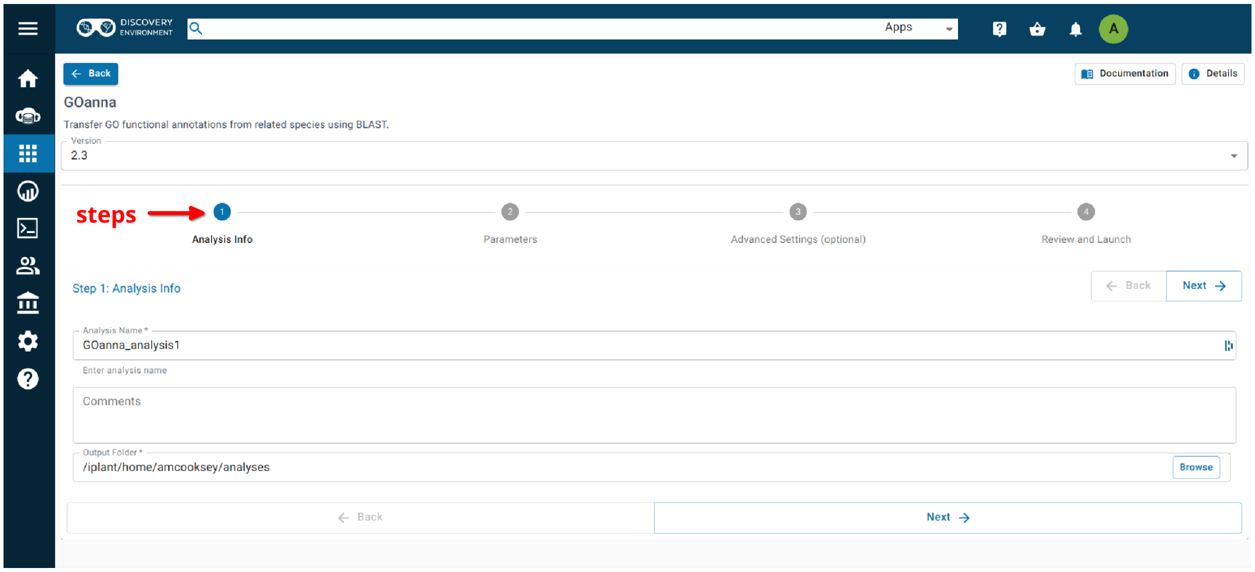
1. Create an account on CyVerse (free). The user guide can be found here.
2. Open the CyVerse Discovery Environment (DE) and login with your CyVerse credentials.
3. Click on the 'Data' tab at the left side of the screen to access your files/folders. Upload your data to the DE.
4. There are several ways to access the GOanna app:

- Use the direct link.
- Search for 'GOanna" in the search bar at the top of the 'apps' tab.
- Follow the AgBase collection (collections tab on left side of DE)



2.3.2 Using the GOanna App

Launching the App



GOanna

Transfer GO functional annotations from related species using BLAST.

Version: 2.3

Step 1: Analysis Info

Analysis Name *: GOanna_analysis1

Enter analysis name

Comments

Output Folder *: /iplant/home/amcooksey/analyses

Step 1. Analysis Info

Version: All of the versions of the GOanna are now available in one place. Plese select the you want from the drop down box. The latest version is best unless you need reproduce a previous analysis.

Analysis Name:GOanna_analysis1: This menu is used to name the job you will run so that you can find it later. The default name is "GOanna_analysis1". We recommend changing the 'analysis1' portion of this to reflect the data you are running.

Comments: (Optional) You can add additional information in the comments section to distinguish your analyses further.

Select output folder: This is where your results will be placed. The default (recommended) is your 'analyses' folder.

Step 2. Parameters

The 'input' section is used to select the BLAST database and your input file.

BLAST database basename: BLAST databases are created by AgBase based upon proteins that have GO available and subsetted by phyla. We recommend selecting the database most closely related to the sequence used as input.

Peptide FASTA file: Use the Browse button on the right hand side to navigate to your Data folder and select your protein sequence file.

Use the 'parameters' section to select your BLAST parameters.

Transfer GO with experimental evidence only: We strongly recommend selecting the "yes" option from the drop-down menu so that only GO annotations based on experimental evidence codes will be transferred . This will ensure the best quality annotations for your data.

The remaining parameters are standard BLAST parameters, and their defaults can be seen beneath the fields.

Determining BLAST Parameters to Use

BLAST parameters are contingent on the BLAST database used and the composition of the input file, and so will change for each analysis.

Make a subset of 100 randomly selected sequences from your larger dataset and use this as the input for GOanna to test for parameters that give good alignments.

1. To test for good parameters use GOanna by selecting the same database you will use and setting relaxed parameters.
2. Once you have run your subsetted file, use the html file to view alignments, select good alignments and note the parameters for these.

Parse query and subject bar delimited sequence identifiers: This option should be selected if you are using a fasta file with headers that include pipes (|). They will not format correctly otherwise.

> If the 'parse-deflines' option is not checked then BLAST will interpret the ID to be everything before the first space.

The 'output' section is used to format your GO annotation results into a standard gene association file format.

Output File basename: This will be the prefix for your output files. A good name choice is to use the fasta file name (without file extension).

Database of query ID: Use the database that sequences were obtained from (e.g. Refseq), or a recognizable project name if these sequences are not in a database (e.g., i5k project or Smith Lab). The default is 'user_input_db'.

'Assigned by' field of your GAF output file: Enter the name of the entity assigning the function (e.g. Agbase, or Smith Lab). This field is used to track who made the annotations. The default is 'user'.

Taxon ID of the query species: Enter the NCBI taxon number for your species. This can be found by searching for your species name (common or scientific) in the NCBI taxon database. The default is "0000".

Step3. Adavanced Settings (optional)

This page allows you specifiy compute requirements for your analysis (e.g. more memory if your analysis is particularly large). You should be able to leave the defaults for most analyses.

Step4. Review and Launch

This will display all of the parameters you have set (other than default). Missing information that is required will displayed in red. Make sure you are happy with your choices and then clicke the 'launch' button at the bottom.

2.3.3 Understanding Your Results

If all goes well, you should get 4 output files and a 'logs' folder.

<basename>.asn: This is standard BLAST output format that allows for conversion to other formats. You probably won't need to look at this output.

<basename>.html: This output displays in your web browser so that you can view pairwise alignments to determine BLAST parameters.

<basename>.tsv: This is the tab-delimited BLAST output that can be opened and sorted in Excel to determine BLAST parameter values. The file contains the following columns:

- query ID
- query length

- query start
- query end
- subject ID
- subject length
- subject start
- subject end
- e-value
- percent ID
- query coverage
- percent positive ID
- gap openings
- total gaps
- bitscore
- raw score

For more information on the BLAST output parameters see the NCBI BLAST documentation.

<basename>_goanna_gaf.tsv: This is the standard tab-separated GO annotation file format that is used by the GO Consortium and by software tools that accept GO annotation files to do GO enrichment.

If you see more files in your output folder there may have been an error in the analysis or there may have been no GO to transfer. Check the 'condor_stderr' file in the analysis output 'logs' folder.

If that doesn't clarify the problem contact us at agbase@email.arizona.edu or support@cyverse.org.

2.4 GOanna on the Command Line

2.4.1 Getting the databases

To run the tool you need some public data. These files are now available as gzipped files to aid downloading. The directories are best downloaded with iCommands. Once iCommands is setup you can use 'iget' to download the data.

1) agbase_database: species subset to run BLAST against (this command will download the entire directory)

```
iget /iplant/home/shared/iplantcollaborative/protein_blast_dbs/agbase_database
```

2) go_info: Uniprot GO annotations (this command will download the entire directory)

```
iget /iplant/home/shared/iplantcollaborative/protein_blast_dbs/go_info
```

2.4.2 Container Technologies

GOanna is provided as a Docker container.

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

There are two major containerization technologies: **Docker** and **Singularity**.

Docker containers can be run with either technology.

2.4.3 Running GOanna using Docker

About Docker

- Docker must be installed on the computer you wish to use for your analysis.
- To run Docker you must have 'root' permissions (or use sudo).
- Docker will run all containers as 'root'. This makes Docker incompatible with HPC systems (see Singularity below).
- Docker can be run on your local computer, a server, a cloud virtual machine (such as CyVerse Atmosphere) etc. Docker can be installed quickly on an Atmosphere instance by typing 'ezd'.
- For more information on installing Docker on other systems see this tutorial: Installing Docker on your machine.

Getting the GOanna container

The GOanna tool is available as a Docker container on Docker Hub: GOanna container

The container can be pulled with this command:

```
docker pull agbase/goanna:2.3
```

Remember

You must have root permissions or use sudo, like so:

sudo docker pull agbase/goanna:2.3

Running GOanna with Data

Getting the Help and Usage Statement

```
sudo docker run --rm -v $(pwd):/work-dir agbase/goanna:2.3 -h
```

See *Help and Usage Statement*

Tip: There are 3 directories built into this container. These directories should be used to mount data.

- /agbase_database
- /go_info
- /work-dir

GOanna has three required parameters:

```
-a BLAST database basename (acceptable options are listed in the help/usage)
-c peptide FASTA file to BLAST
-o output file basename
```

Example Command

```
sudo docker run \
--rm \
-v /location/of/agbase_database:/agbase_database \
-v /location/of/go_info:/go_info \
-v $(pwd):/work-dir \
agbase/goanna:2.3 \
-a invertebrates \
-c AROS_10.faa \
-o AROS_10_invert_exponly \
-p \
-g 70 \
-s 900 \
-d RefSeq \
-u "Amanda Cooksey" \
-x 37344 \
-k 9 \
-q 70
```

Command Explained

sudo docker run: tells docker to run

–rm: removes the container when the analysis has finished. The image will remain for future use.

-v /location/of/agbase_database:/agbase_database: tells docker to mount the 'agbase_database' directory you downloaded to the host machine to the '/agbase_database' directory within the container. The syntax for this is: <absolute path on host>:<absolute path in container>

-v /locaiton/of/go_info:/go_info: mounts 'go_info' directory on host machine into 'go_info' directory inside the container

-v $(pwd):/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

agbase/goanna:2.3: the name of the Docker image to use

Tip: All the options supplied after the image name are GOanna options

-a invertebrates: GOanna BLAST database to use–first of three required options.

-c AROS_10.faa: input file (peptide FASTA)–second of three required options

-o AROS_10_invert_exponly: output file basename–last of three required options

-p: our input file has NCBI deflines. This specifies how to parse them.

-g 70: tells GOanna to keep only those matches with at least 70% identity

-s 900: tells GOanna to keep only those matches with a bitscore above 900

-d RefSeq: database of query ID. This will appear in column 1 of the GAF output file.

-u "Amanda Cooksey": name to appear in column 15 of the GAF output file

-x 37344: NCBI taxon ID of input file species will appear in column 13 of the GAF output file

-k 9: tells GOanna to keep only those matches with a maximum number of 9 gap openings

-q 70: tells GOanna to keep only those matches with query coverage of 70 per subject

Understanding Your Results

If all goes well, you should get 4 output files:

<basename>.asn: This is standard BLAST output format that allows for conversion to other formats. You probably won't need to look at this output.

<basename>.html: This output displays in your web browser so that you can view pairwise alignments to determine BLAST parameters.

<basename>.tsv: This is the tab-delimited BLAST output that can be opened and sorted in Excel to determine BLAST parameter values. The file contains the following columns:

- Query ID
- query length
- query start
- query end
- subject ID
- subject length
- subject start
- subject end
- e-value
- percent ID
- query coverage
- percent positive ID
- gap openings
- total gaps
- bitscore
- raw score

For more information on the BLAST output parameters see the NCBI BLAST documentation.

<basename>_goanna_gaf.tsv: This is the standard tab-separated GO annotation file format that is used by the GO Consortium and by software tools that accept GO annotation files to do GO enrichment.

If you see more files in your output folder there may have been an error in the analysis or there may have been no GO to transfer. Contact us.

2.4.4 Running GOanna using Singularity

About Singularity

- does not require 'root' permissions
- runs all containers as the user that is logged into the host machine
- HPC systems are likely to have Singularity installed and are unlikely to object if asked to install it (no guarantees).
- can be run on any machine where is is installed
- more information about installing Singularity
- This tool was tested using Singularity 3.0. Users with Singularity 2.x will need to modify the commands accordingly.

HPC Job Schedulers

Although Singularity can be installed on any computer this documentation assumes it will be run on an HPC system. The tool was tested on a PBSPro system and the job submission scripts below reflect that. Submission scripts will need to be modified for use with other job scheduler systems.

Getting the GOanna Container

The GOanna tool is available as a Docker container on Docker Hub: GOanna container

The container can be pulled with this command:

```
singularity pull docker://agbase/goanna:2.3
```

Running GOanna with Data

Getting the Help and Usage Statement

Example PBS script:

```
#!/bin/bash
#PBS -N goanna
#PBS -W group_list=fionamcc
#PBS -l select=1:ncpus=28:mem=168gb
#PBS -q standard
#PBS -l walltime=6:0:0
#PBS -l cput=168:0:0

module load singularity

cd /where/to/save/singularity/image

singularity pull docker://agbase/goanna:2.3

singularity run \
```

(continues on next page)

(continued from previous page)

```
goanna_2.0.sif \
-h
```

See *Help and Usage Statement*

Tip: There are 3 directories built into this container. These directories should be used to mount data.

- /agbase_database
- /go_info
- /work-dir

GOanna has three required parameters:

```
-a BLAST database basename (acceptable options are listed in the help/usage)
-c peptide FASTA file to BLAST
-o output file basename
```

Example PBS Script

```
#!/bin/bash
#PBS -N goanna
#PBS -W group_list=fionamcc
#PBS -l select=1:ncpus=28:mem=168gb
#PBS -q standard
#PBS -l walltime=6:0:0
#PBS -l cput=168:0:0

module load singularity

cd /where/to/save/singularity/image

singularity pull docker://agbase/goanna:2.3

singularity run \
-B /location/of/agbase_database:/agbase_database \
-B /location/of/go_info:/go_info \
-B /directory/where/you/will/work:/work-dir \
goanna_2.3.sif \
-a invertebrates \
-c AROS_10.faa \
-o AROS_10_invert_exponly \
-p \
-g 70 \
-s 900 \
-d RefSeq \
-u "Amanda Cooksey" \
-x 37344 \
-t 28 \
-q 70 \
-k 9
```

Command Explained

singularity run: tells Singularity to run

-B /location/of/agbase_database:/agbase_database: tells docker to mount the 'agbase_database' directory I downloaded to the host machine to the '/agbase_database' directory within the container. The syntax for this is: <absolute path on host>:<absolute path in container>

-B /location/of/go_info:/go_info: mounts 'go_info' directory on host machine into 'go_info' directory inside the container

-B /directory/where/you/will/work:/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

goanna_2.3.sif: the name of the Singularity image file to use

Tip: All the options supplied after the image name are GOanna options

-a invertebrates: GOanna BLAST database to use–first of three required options.

-c AROS_10.faa: input file (peptide FASTA)–second of three required options

-o AROS_10_invert_exponly: output file basename–last of three required options

-p: our input file has NCBI deflines. This specifies how to parse them.

-g 70: tells GOanna to keep only those matches with at least 70% identity

-s 900: tells GOanna to keep only those matches with a bitscore above 900

-d RefSeq: database of query ID. This will appear in column 1 of the GAF output file.

-u "Amanda Cooksey": name to appear in column 15 of the GAF output file

-x 37344: NCBI taxon ID of input file species will appear in column 13 of the GAF output file

-t 28: number of threads to use for BLAST. This was run on a node with 28 cores.

-k 9: tells GOanna to keep only those matches with a maximum number of 9 gap openings

-q 70: tells GOanna to keep only those matches with query coverage of 70 per subject

Understanding Your Results

If all goes well, you should get 4 output files:

<basename>.asn: This is standard BLAST output format that allows for conversion to other formats. You probably won't need to look at this output.

<basename>.html: This output displays in your web browser so that you can view pairwise alignments to determine BLAST parameters.

<basename>.tsv: This is the tab-delimited BLAST output that can be opened and sorted in Excel to determine BLAST parameter values. The file contains the following columns:

- Query ID
- query length
- query start
- query end

- subject ID
- subject length
- subject start
- subject end
- e-value
- percent ID
- query coverage
- percent positive ID
- gap openings
- total gaps
- bitscore
- raw score

For more information on the BLAST output parameters see the NCBI BLAST documentation.

<basename>_goanna_gaf.tsv: This is the standard tab-separated GO annotation file format that is used by the GO Consortium and by software tools that accept GO annotation files to do GO enrichment.

If you see more files in your output folder there may have been an error in the analysis or there may have been no GO to transfer. Contact us.

2.5 Intro

InterPro is a database which integrates together predictive information about proteins' function from a number of partner resources, giving an overview of the families that a protein belongs to and the domains and sites it contains.

Basic functions of this tool

- removes special characters from FASTA sequences
- splits FASTA into groups of 1000 sequences
- runs InterProScan with user-specified options on each of the 1000-sequence files in parallel
- re-combines output files from all groups of 1000
- parses the XML output from InterProScan to generate a gene association file (GAF) (and several other files)

Results and analysis from the application of InterProScan annotation to the Official gene set v3.0 protein set from *Diaphorina citri* followed by a differential expression analysis was presented at a seminar in the University of Arizona Animal and Comparative Biomedical Sciences in Fall 2020. The slides and video are available online.

Note: This tool accepts a peptide FASTA file. For those users with nucleotide sequences some documentation has been provided for using **TransDecoder** (although other tools are also acceptable). The TransDecoder app is available through CyVerse or as a BioContainer for use on the command line.

Note: As both GOanna and InterProScan provide GO annotations, their outputs are provided in GAF format. The **'Combine GAFs'** tool can then be used to make a single GAF of GO annotations, if desired.

2.5.1 Where to Find InterProScan

Docker Hub (5.63-95)

CyVerse (5.36-75)

2.5.2 Getting the InterProScan Data

InterProScan Data (now includes Panther)

```
wget ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/5.63-95.0/alt/interproscan-data-
↪5.63-95.0.tar.gz
wget ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/5.63-95.0/alt/interproscan-data-
↪5.63-95.0.tar.gz.md5
md5sum -c interproscan-data-5.63-95.0.tar.gz.md
tar -pxvzf interproscan-data-5.63-795.0.tar.gz
```

tar options

- p = preserve the file permissions
- x = extract files from an archive
- v = verbosely list the files processed
- z = filter the archive through gzip
- f = use archive file

2.5.3 Help and Usage Statement

```
   Options:
-a  <ANALYSES>                              Optional, comma separated list of analyses.
↪  If this option
                                            is not set, ALL analyses will be run.

-b <OUTPUT-FILE-BASE>                      Optional, base output filename (relative or␣
↪absolute path).
                                            Note that this option, the output␣
↪directory (-d) option and
                                            the output file name (-o) option are␣
↪mutually exclusive.  The
                                            appropriate file extension for the output␣
↪format(s) will be
                                            appended automatically. By default the␣
↪input file
                                            path/name will be used.

-d <OUTPUT-DIR>                             Optional, output directory. Note that this␣
↪option, the
                                            output file name (-o) option and the␣
↪output file base (-b) option
                                            are mutually exclusive. The output␣
↪filename(s) are the
                                            same as the input filename, with the␣
↪appropriate file
```

(continues on next page)

(continued from previous page)

```
                                          extension(s) for the output format(s)␣
↪appended automatically .

-c                                        Optional.  Disables use of the␣
↪precalculated match lookup
                                          service.  All match calculations will be␣
↪run locally.

-C                                        Optional. Supply the number of cpus to use.

-e                                        Optional, excludes sites from the XML,␣
↪JSON output

-f <OUTPUT-FORMATS>                       Optional, case-insensitive, comma␣
↪separated list of output
                                          formats. Supported formats are TSV, XML,␣
↪JSON, GFF3, HTML and
                                          SVG. Default for protein sequences are TSV,
↪ XML and
                                          GFF3, or for nucleotide sequences GFF3 and␣
↪XML.

-g                                        Optional, switch on lookup of␣
↪corresponding Gene Ontology
                                          annotation (IMPLIES -l lookup option)

-h                                        Optional, display help information

-i <INPUT-FILE-PATH>                      Optional, path to fasta file that should␣
↪be loaded on
                                          Master startup. Alternatively, in CONVERT␣
↪mode, the
                                          InterProScan 5 XML file to convert.

-l                                        Also include lookup of corresponding␣
↪InterPro
                                          annotation in the TSV and GFF3 output␣
↪formats.

-m <MINIMUM-SIZE>                         Optional, minimum nucleotide size of ORF␣
↪to report. Will
                                          only be considered if n is specified as a␣
↪sequence type.
                                          Please be aware of the fact that if you␣
↪specify a too
                                          short value it might be that the analysis␣
↪takes a very long
                                          time!

-o <EXPLICIT_OUTPUT_FILENAME>             Optional explicit output file name␣
↪(relative or absolute
                                          path).  Note that this option, the output␣
↪directory -d option
                                          and the output file basename -b option are␣
↪mutually
                                          exclusive. If this option is given, you␣
↪MUST specify a
```

(continues on next page)

(continued from previous page)

```
                                           single output format using the -f option. ␣
↪The output file
                                           name will not be modified. Note that␣
↪specifying an output
                                           file name using this option OVERWRITES ANY␣
↪EXISTING FILE.

-p                                         Optional, switch on lookup of␣
↪corresponding Pathway
                                           annotation (IMPLIES -l lookup option)
-t <SEQUENCE-TYPE>                         Optional, the type of the input sequences␣
↪(dna/rna (n)
                                           or protein (p)).  The default sequence␣
↪type is protein.

-T <TEMP-DIR>                              Optional, specify temporary file directory␣
↪(relative or
                                           absolute path). The default location is␣
↪temp/.

-v                                         Optional, display version number

-r                                          Optional. 'Mode' required ( -r 'cluster')␣
↪to run in cluster mode. These options
                                           are provided but have not been tested with␣
↪this wrapper script. For
                                           more information on running InterProScan␣
↪in cluster mode:
                                           https://github.com/ebi-pf-team/
↪interproscan/wiki/ClusterMode

-R                                          Optional. Clusterrunid (crid) required␣
↪when using cluster mode.
                                           -R unique_id
```

Available InterProScan analyses:

- CDD
- COILS
- Gene3D
- HAMAP
- MOBIDB
- PANTHER
- Pfam
- PIRSF
- PRINTS
- PROSITE (Profiles and Patterns)
- SFLD
- SMART (unlicensed components only by default - this analysis has simplified post-processing that includes an E-value filter, however you should not expect it to give the same match output as the fully licensed version of

SMART)

- SUPERFAMILY
- NCBIFAM (includes the previous TIGRFAM analysis)

OPTIONS FOR XML PARSER OUTPUTS

-F <IPRS output directory>	This is the output directory from InterProScan.
-D <database>	Supply the database responsible for these annotations.
-x <taxon>	NCBI taxon ID of the ID being annotated
-y <type>	Transcript or protein
-n <biocurator>	Name of the biocurator who made these annotations
-M <mapping file>	Optional. Mapping file.
-B <bad seq file>	Optional. Bad input sequence file.

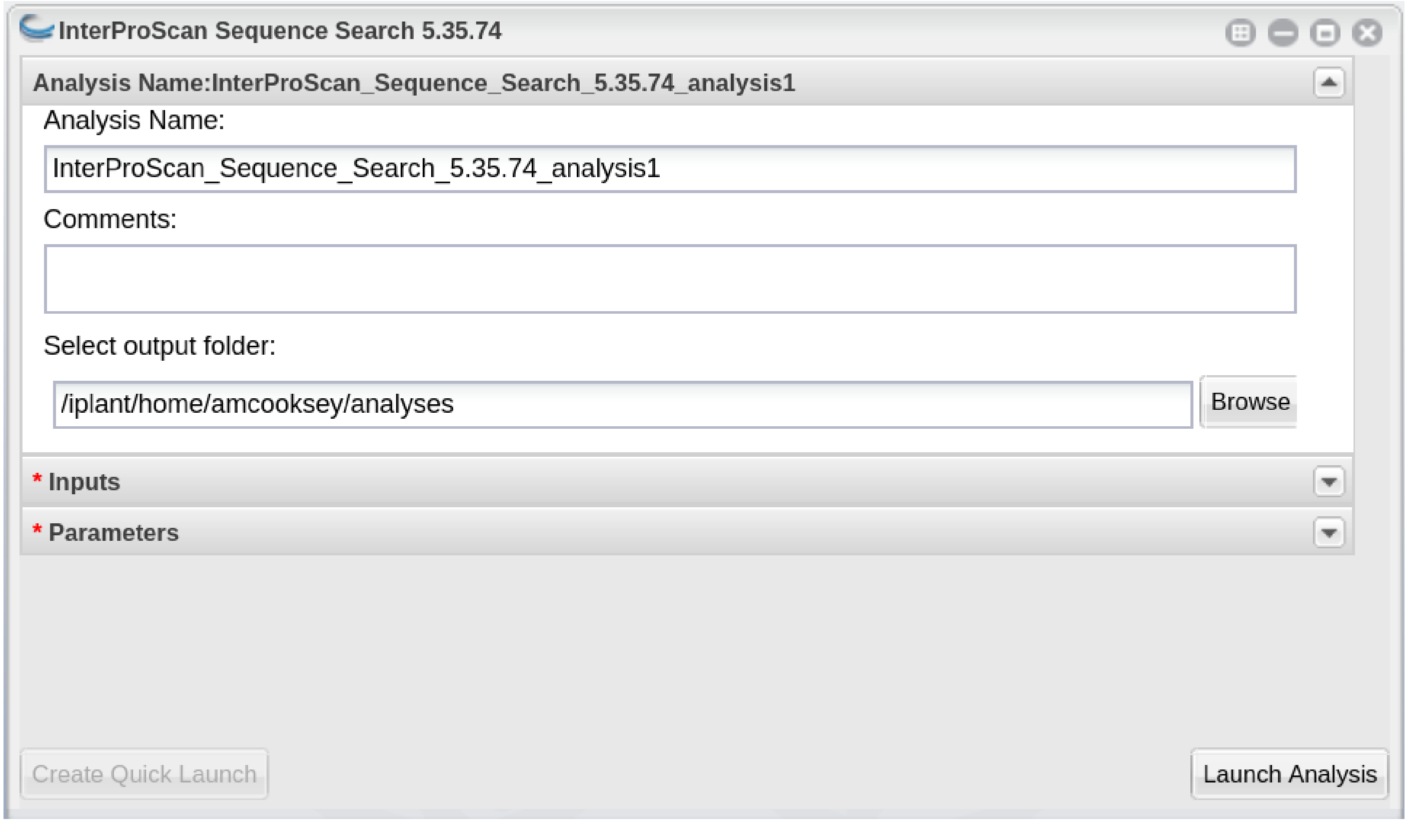
2.6 InterProScan on CyVerse

2.6.1 Accessing InterProScan in the Discovery Environment

1. Create an account on CyVerse (free)
2. Open the CyVerse Discovery Environment (DE) and login with your CyVerse credentials.
3. If you are new to the Discovery Environment (DE) the user guide can be found here.
4. Click on the 'Data' button at the left side of the screen to access your files/folders. Upload your data to the DE.
5. To access the InterProScan Sequence Search 5.36-75.0 app click on the 'Apps' button at the left side of the DE.
6. Search for 'interproscan' in the search bar at the top of the 'apps' window. The contents of the folder will appear in the main pane of the window. The InterProScan app is called 'InterProScan Sequence Search 5.36-75'; click on the name to open the app.

2.6.2 Using the InterProScan App

Launching the App



InterProScan Sequence Search 5.35.74

Analysis Name:InterProScan_Sequence_Search_5.35.74_analysis1

Analysis Name:

InterProScan_Sequence_Search_5.35.74_analysis1

Comments:

Select output folder:

/iplant/home/amcooksey/analyses

InterProScan_Sequence_Search_5.36.75_analysis1: This menu is used to name the job you will run so that you can find it later. Analysis Name: The default name is "InterProScan_Sequence_Search_5.36.75_analysis1". We recommend changing the 'analysis1' portion of this to reflect the data you are running.

Comments: (Optional) You can add additional information in the comments section to distinguish your analyses further.

Select output folder: This is where your results will be placed. The default (recommended) is your 'analyses' folder.

Retain Inputs: Enabling this flag will copy all the input files into the analysis result folder.

> **Warning:** Selecting this option will rapidly consume your allocated space. It is not recommended. Your inputs will always remain available in the folder in which you stored them.

Input

Peptide FASTA file: Use the Browse button on the right hand side to navigate to your Data folder and select your protein sequence file.

Parameters

Annotate each peptide with Gene Ontology information: Be sure this box is checked. This will ensure that you get GO annotations

Biocurator: This will be used to fill the 'assigned by' field of your GAF output file. If you do not fill it in the default "user" will be used instead.

Database: Use the database that sequences were obtained from (Genbank), or a recognizable project name if these sequences are not in a database (e.g., i5k project or Smith Lab).

Annotate each peptide with biological pathway information: This is optional. However, if you want pathways annotations be it is checked.

Taxon: Enter the NCBI taxon number for your species. This can be found by searching for your species name (common or scientific) in the NCBI taxon database.

InterProScan output directory name: This will be the name of the folder for your output files. The default folder name is 'outdir'.

2.6.3 Understanding Your Results

InterProScan Outputs

This app provides all six of the InterProScan output formats. For more details on the contents of each file please refer to the InterProScan outputs documentation.

<basename>.gff3

<basename>.tsv

<basename>.xml

<basename>.json

<basename>.html.tar.gz

<basename>.svg.tar.gz

This app also runs the 'InterProScan Results Function' on the XML output from InterProScan. This tool provides a GAF output file and a variety of summary (count) files described below.

InterProScan Results Function Outputs

<basename>_gaf.txt: -This table follows the formatting of a gene association file (gaf) and can be used in GO enrichment analyses.

<basename>_acc_go_counts.txt: -This table includes input accessions, the number of GO IDs assigned to each accession and GO ID names. GO IDs are split into BP (Biological Process), MF (Molecular Function) and CC (Cellular Component).

<basename>_go_counts.txt: -This table counts the numbers of sequences assigned to each GO ID so that the user can quickly identify all genes assigned to a particular function.

<basename>_acc_interpro_counts.txt: -This table includes input accessions, number of InterPro IDs for each accession, InterPro IDs assigned to each sequence and the InterPro ID name.

<basename>_interpro_counts.txt: -This table counts the numbers of sequences assigned to each InterPro ID so that the user can quickly identify all genes with a particular motif.

<basename>_acc_pathway_counts.txt: -This table includes input accessions, number of pathway IDs for the accession and the pathway names. Multiple values are separated by a semi-colon.

<basename>_pathway_counts.txt: -This table counts the numbers of sequences assigned to each Pathway ID so that the user can quickly identify all genes assigned to a pathway.

<basename>.err: -This file will list any sequences that were not able to be analyzed by InterProScan. Examples of sequences that will cause an error are sequences with a large run of Xs.

If you output doesn't look like you expect please check the 'condor_stderr' file in the analysis output 'logs' folder. If that doesn't clarify the problem contact us at agbase@email.arizona.edu or support@cyverse.org.

2.7 InterProScan on the Command Line

2.7.1 Getting the InterProScan Data (now including PANTHER)

```
wget ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/5.63-95.0/alt/interproscan-data-
↪5.63-95.0.tar.gz
wget ftp://ftp.ebi.ac.uk/pub/software/unix/iprscan/5/5.63-95.0/alt/interproscan-data-
↪5.63-95.0.tar.gz.md5
md5sum -c interproscan-data-5.63-95.0.tar.gz.md
tar -pxvzf interproscan-data-5.63-95.0.tar.gz
```

tar options

- p = preserve the file permissions
- x = extract files from an archive
- v = verbosely list the files processed
- z = filter the archive through gzip
- f = use archive file

2.7.2 Container Technologies

Interproscan is provided as a Docker container.

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

There are two major containerization technologies: **Docker** and **Singularity (also known as Apptainer**.

Docker containers can be run with either technology.

2.7.3 Running InterProScan using Docker

About Docker

- Docker must be installed on the computer you wish to use for your analysis.
- To run Docker you must have 'root' permissions (or use sudo).
- Docker will run all containers as 'root'. This makes Docker incompatible with HPC systems (see Singularity below).
- Docker can be run on your local computer, a server, a cloud virtual machine etc.
- For more information on installing Docker on other systems see this tutorial: Installing Docker on your machine.

Important: We have included this basic documentation for running InterProScan with Docker. However, InterProScan requires quite a lot of compute resources and may need to be run on an HPC system. If you need to use HPC see ‘Singularity’ below.

Getting the InterProScan Container

The InterProScan tool is available as a Docker container on Docker Hub where you can see all the available versions: InterProScan container

The latest container can be pulled with this command:

```
docker pull agbase/interproscan:5.63-95
```

Remember

You must have root permissions or use sudo, like so:

sudo docker pull agbase/interproscan:5.63-95

Running InterProScan with Data

Tip: There is one directory built into this container. This directory should be used to mount your working directory.

- /data

Getting the Help and Usage Statement

```
sudo docker run --rm -v $(pwd):/work-dir agbase/interproscan:5.63-95 -h
```

See iprsusage

Example Command

```
sudo docker run \
-v /your/local/data/directory:/data \
-v /where/you/downloaded/interproscan/data/interproscan-5.63-95.0/data:/opt/interproscan/data \
agbase/interproscan:5.63-95 \
-i /path/to/your/input/file/pnnl_10000.fasta \
-d outdir_10000 \
-f tsv,json,xml,gff3 \
-g \
-p \
-c \
-n curator \
```

(continues on next page)

(continued from previous page)

```
-x 109069 \
-D database \
-l
```

Command Explained

sudo docker run: tells docker to run

–rm: removes container when analysis finishes (image will remain for furture analyses)

-v /your/local/data/directory:/data: mount my working directory on the host machine into the /data directory in the container. The syntax for this is <absolute path on host machine>:<absolute path in container>

-v /where/you/downloaded/interproscan/data/interproscan-5.64-95.0/data:/opt/interproscan/data: mounts the InterProScan partner data (downloaded from FTP) on the host machine into the /opt/interproscan/data directory in the container

agbase/interproscan:5.63-95: the name of the Docker image to use

Tip: All the options supplied after the image name are Interproscan options

-i /path/to/your/input/file/pnnl_10000.fasta: local path to input FASTA file. You can also use the mounted file path: /data/pnnl_10000.fasta

-d outdir_10000: output directory name

-f tsv,json,xml,gff3: desired output file formats

-g: tells the tool to perform GO annotation

-p: tells tool to perform pathway annotaion

-c: tells tool to perform local compute and not connect to EBI. This only adds a little to the run time but removes error messages from network time out errors

-n curator: name of biocurator to include in column 15 of GAF output file

-x 109069: taxon ID of query species to be used in column 13 of GAF output file

-D database: database of query accession to be used in column 1 of GAF output file

-l: tells tools to include lookup of corresponding InterPro annotation in the TSV and GFF3 output formats.

Understanding Your Results

InterProScan outputs: https://github.com/ebi-pf-team/interproscan/wiki/OutputFormats

- <basename>.gff3
- <basename>.tsv
- <basename>.xml
- <basename>.json

Parser Outputs

<basename>_gaf.txt: -This table follows the formatting of a gene association file (gaf) and can be used in GO enrichment analyses.

<basename>_acc_go_counts.txt: -This table includes input accessions, the number of GO IDs assigned to each accession and GO ID names. GO IDs are split into BP (Biological Process), MF (Molecular Function) and CC (Cellular Component).

<basename>_go_counts.txt: -This table counts the numbers of sequences assigned to each GO ID so that the user can quickly identify all genes assigned to a particular function.

<basename>_acc_interpro_counts.txt: -This table includes input accessions, number of InterPro IDs for each accession, InterPro IDs assigned to each sequence and the InterPro ID name.

<basename>_interpro_counts.txt: -This table counts the numbers of sequences assigned to each InterPro ID so that the user can quickly identify all genes with a particular motif.

<basename>_acc_pathway_counts.txt: -This table includes input accessions, number of pathway IDs for the accession and the pathway names. Multiple values are separated by a semi-colon.

<basename>_pathway_counts.txt: -This table counts the numbers of sequences assigned to each Pathway ID so that the user can quickly identify all genes assigned to a pathway.

<basename>.err: -This file will list any sequences that were not able to be analyzed by InterProScan. Examples of sequences that will cause an error are sequences with a large run of Xs.

If you see more files in your output folder there may have been an error in the analysis or there may have been no GO to transfer. Contact us.

2.7.4 Running InterProScan with Singularity (or Apptainer) on HPC

About Singularity

- does not require 'root' permissions
- runs all containers as the user that is logged into the host machine
- HPC systems are likely to have Singularity (or Apptainer) installed and are unlikely to object if asked to install it (no guarantees).
- can be run on any machine where is is installed
- more information about Singularity and Apptainer
- This tool was tested using SingularityCE 3.11.4

HPC Job Schedulers

Although Singularity can be installed on any computer this documentation assumes it will be run on an HPC system. The tool was tested on a SLURM system and the job submission scripts below reflect that. Submission scripts will need to be modified for use with other job scheduler systems.

Getting the InterProScan Container

The InterProScan tool is available as a Docker container on Docker Hub: InterProScan container

The container can be pulled with this command:

```
singularity pull docker://agbase/interproscan:5.63-95
```

Getting the Help and Usage Statement

Example SLURM script:

```
#!/bin/bash
#SBATCH --job-name=jobname
#SBATCH --ntasks=48
#SBATCH --nodes=1
#SBATCH --mem=0
#SBATCH --time=48:00:00
#SBATCH --partition=short
#SBATCH --account=nal_genomics


module load singularityCE

singularity run \
interproscan_5.63-95.sif \
-h
```

See iprsusage

Running InterProScan with Data

Tip: There is one directory built into this container. This directory should be used to mount your working directory.

- /data

Example SLURM Script

```
#!/bin/bash
#SBATCH --job-name=jobname
#SBATCH --ntasks=48
#SBATCH --nodes=1
#SBATCH --mem=0
#SBATCH --time=48:00:00
#SBATCH --partition=short
#SBATCH --account=nal_genomics

module load singularityCE

singularity run \
-B /your/local/data/directory:/data \
```

(continues on next page)

(continued from previous page)

```
-B /where/you/downloaded/interproscan/data/interproscan-5.63-85.0/data:/opt/
↪interproscan/data \
interproscan_5.63-95.sif \
-i /your/local/data/directory/pnnl_10000.fasta \
-d outdir_10000 \
-f tsv,json,xml,gff3 \
-g \
-p \
-c \
-n biocurator \
-x 109069 \
-D database \
-l
```

Command Explained

singularity run: tells Singularity to run

-B /your/local/data/directory:/data: mounts my working directory on the host machine into the /data directory in the container the syntax for this is <aboslute path on host machine>:<aboslute path in container>

-B /where/you/downloaded/interproscan/data/interproscan-5.63-95.0/data:/opt/interproscan/data: mounts he InterProScan data directory that was downloaded from the FTP site into the InterProScan data directory in the container

interproscan_5.63-95.sif: name of the image to use

Tip: All the options supplied after the image name are options for this tool

-i /your/local/data/directory/pnnl_10000.fasta: input FASTA file

-d outdir_10000: output directory name

-f tsv,json,xml,gff3: desired output file formats

-g: tells the tool to perform GO annotation

-c: tells tool to perform local compute and not connect to EBI. This only adds a little to the run time but removes error messages from network time out errors

-p: tells tool to perform pathway annoation

-n biocurator: name of biocurator to include in column 15 of GAF output file

-x 109069: taxon ID of query species to be used in column 13 of GAF output file

-D database: database of query accession to be used in column 1 of GAF output file

-l: tells tools to include lookup of corresponding InterPro annotation in the TSV and GFF3 output formats.

Understanding Your Results

InterProScan outputs: https://github.com/ebi-pf-team/interproscan/wiki/OutputFormats

- <basename>.gff3

- <basename>.tsv
- <basename>.xml
- <basename>.json

Parser Outputs

<basename>_gaf.txt: -This table follows the formatting of a gene association file (gaf) and can be used in GO enrichment analyses.

<basename>_acc_go_counts.txt: -This table includes input accessions, the number of GO IDs assigned to each accession and GO ID names. GO IDs are split into BP (Biological Process), MF (Molecular Function) and CC (Cellular Component).

<basename>_go_counts.txt: -This table counts the numbers of sequences assigned to each GO ID so that the user can quickly identify all genes assigned to a particular function.

<basename>_acc_interpro_counts.txt: -This table includes input accessions, number of InterPro IDs for each accession, InterPro IDs assigned to each sequence and the InterPro ID name.

<basename>_interpro_counts.txt: -This table counts the numbers of sequences assigned to each InterPro ID so that the user can quickly identify all genes with a particular motif.

<basename>_acc_pathway_counts.txt: -This table includes input accessions, number of pathway IDs for the accession and the pathway names. Multiple values are separated by a semi-colon.

<basename>_pathway_counts.txt: -This table counts the numbers of sequences assigned to each Pathway ID so that the user can quickly identify all genes assigned to a pathway.

<basename>.err: -This file will list any sequences that were not able to be analyzed by InterProScan. Examples of sequences that will cause an error are sequences with a large run of Xs.

If you see more files in your output folder there may have been an error in the analysis or there may have been no GO to transfer. Contact us.

2.8 Intro

This tool can be used to combine the gene association file (GAF) outputs from GOanna and InterProScan.

The tool accepts two input files:

1. GOanna GAF output
2. InterProScan GAF output

Note: InterProScan itself does not produce a GAF file. The AgBase InterProScan container parses the XML output from InterProScan to produce the GAF file.

2.8.1 Where to Find Combine GAFs

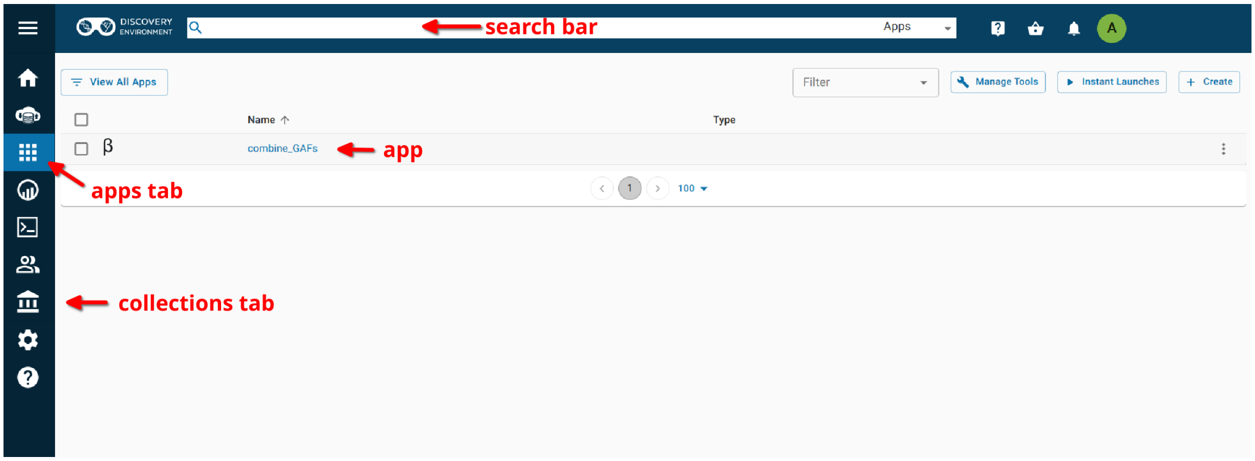
- Docker Hub
- Cyverse Discovery Environment

2.9 Combine GAFs on CyVerse

2.9.1 Accessing GOanna in the Discovery Environment

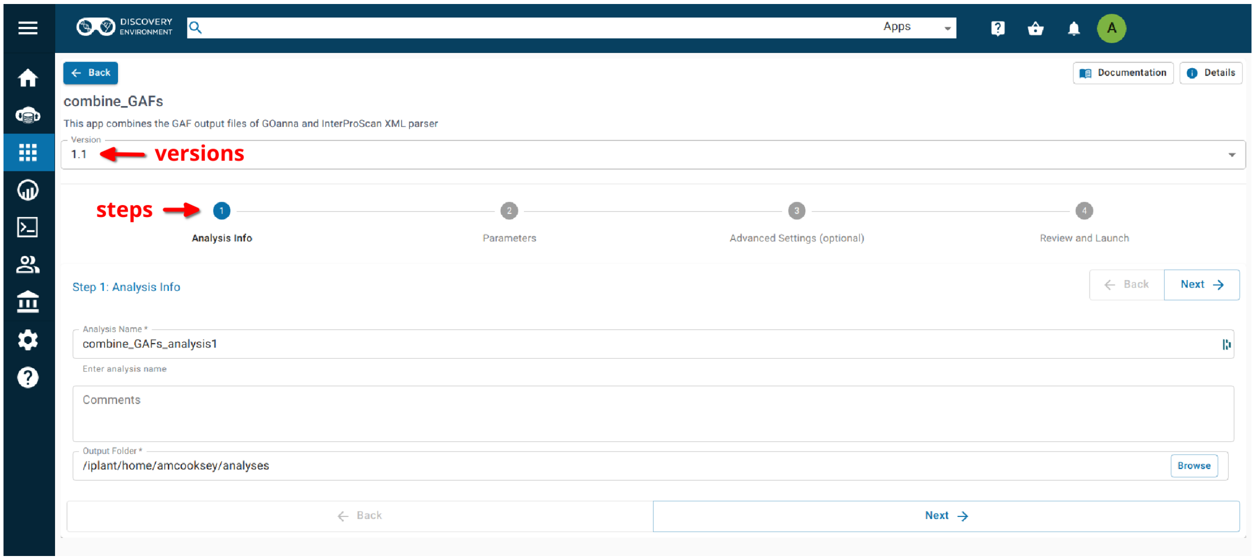
1. Create an account on CyVerse (free). The user guide can be found here.
2. Open the CyVerse Discovery Environment (DE) and login with your CyVerse credentials.
4. There are several ways to access the combine_GAFs app:

- Use the direct link.
- Search for ‘combine_GAFs in the search bar at the top of the ‘apps’ tab.
- Follow the AgBase collection (collections tab on left side of DE)



Using the Combine_GAFs App

Step 1. Analysis Info



combine_GAFs

This app combines the GAF output files of GOanna and InterProScan XML parser

Version: 1.1

Step 1: Analysis Info

Analysis Name *: combine_GAFs_analysis1

Enter analysis name

Comments

Output Folder *: /iplant/home/amcooksey/analyses

Analysis Name: Combine_GAFs_analysis1: This menu is used to name the job you will run so that you can find it later. Analysis Name: The default name is "Combine_GAFs_analysis1". We recommend changing the 'analysis1' portion of this to reflect the data you are running.

Comments: (Optional) You can add additional information in the comments section to distinguish your analyses further.

Select output folder: This is where your results will be placed. The default (recommended) is your 'analyses' folder.

Step 2. Parameters

GOanna GAF Output File: This is the GAF file generated by a GOanna analysis.

InterProScan XML Parser GAF Output File: This is the GAF output file generated by an InterProScan XML Parser analysis. InterProScan itself does not produce this file, though some IntperProScan apps include this analysis. If it is missing from your InterProScan output you can generate it using the InterProScan XML Parser app.

Output

Output File Basename: This will be the prefix for your output file (a .tsv extension will be added).

Step3. Adavanced Settings (optional)

This page allows you specifiy compute requirements for your analysis (e.g. more memory if your analysis is particularly large). You should be able to leave the defaults for most analyses.

Step4. Review and Launch

This will display all of the parameters you have set (other than default). Missing information that is required will displayed in red. Make sure you are happy with your choices and then clicke the 'launch' button at the bottom.

If your analysis fails please check the 'condor_stderr' file in the analysis output 'logs' folder. If that doesn't clarify the problem contact us at agbase@email.arizona.edu or support@cyverse.org.

2.10 Combine GAFs on the Command Line

2.10.1 Container Technologies

GOanna is provided as a Docker container.

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

There are two major containerization technologies: **Docker** and **Singularity (Apptainer)**.

Docker containers can be run with either technology.

2.10.2 Combine GAFs using Docker

About Docker

- Docker must be installed on the computer you wish to use for your analysis.

- To run Docker you must have 'root' permissions (or use sudo).
- Docker will run all containers as 'root'. This makes Docker incompatible with HPC systems (see Singularity below).
- Docker can be run on your local computer, a server, a cloud virtual machine etc.
- For more information on installing Docker on other systems see this tutorial: Installing Docker on your machine.

Getting the Combine GAFs container

The Combine GAFs tool is available as a Docker container on Docker Hub: Combine GAFs container

The container can be pulled with this command:

```
docker pull agbase/combine_gafs:1.1
```

Remember

You must have root permissions or use sudo, like so:

sudo docker pull agbase/combine_gafs:1.1

Running Combine GAFs with Data

Combine GAFs has three parameters:

```
-i InterProScan XML Parser GAF output
-g GOanna GAF output
-o output file basename
```

Example Command

```
sudo docker run \
--rm \
-v $(pwd):/work-dir \
agbase/combine_gafs:1.1 \
-i CFLO_1.fa_gaf.txt \
-g clfo1_v_insecta_goanna_gaf.tsv \
-o complete_gaf
```

Command Explained

sudo docker run: tells docker to run

–rm: removes the container when the analysis has finished. The image will remain for future use.

-v $(pwd):/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

agbase/combine_gafs:1.1: the name of the Docker image to use

Tip: All the options supplied after the image name are Combine_GAFs options

-i CFLO_1.fa_gaf.txt: InterProScan XML Parser GAF output file.

-g clfo1_v_insecta_goanna_gaf.tsv: GOanna GAF output file.

-o complete_gaf: output file basename–a .tsv extension will be added

2.10.3 Combine GAFs using Singularity (Apptainer)

About Singularity (Apptainer)

- does not require 'root' permissions
- runs all containers as the user that is logged into the host machine
- HPC systems are likely to have Singularity installed and are unlikely to object if asked to install it (no guarantees).
- can be run on any machine where is is installed
- more information about installing Singularity
- This tool was tested using Singularity 3.10.2.

HPC Job Schedulers

Although Singularity can be installed on any computer this documentation assumes it will be run on an HPC system. The tool was tested on a SLURM system and the job submission scripts below reflect that. Submission scripts will need to be modified for use with other job scheduler systems.

Getting the Combine GAFs Container

The Combine GAFs tool is available as a Docker container on Docker Hub: Combine GAFs container

The container can be pulled with this command:

```
singularity pull docker://agbase/combine_gafs:1.1
```

Running Combine GAFs with Data

Combine GAFs has three parameters:

```
-i InterProScan XML Parser GAF output
-g GOanna GAF output
-o output file basename
```

Example SLURM Script

```
#!/bin/bash
#SBATCH --job-name=combine_gafs
#SBATCH --ntasks=8
#SBATCH --time=2:00:00
#SBATCH --partition=short
#SBATCH --account=nal_genomics

module load singularityCE

singularity run \
-B /directory/you/want/to/work/in:/work-dir \
combine_gafs_1.1.sif \
-i CFLO_1.fa_gaf.txt \
-g clfo1_v_insecta_goanna_gaf.tsv \
-o complete_gaf
```

Command Explained

singularity run: tells Singularity to run

-B /directory/you/want/to/work/in:/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

combine_gafs_1.1.sif: the name of the Singularity image file to use

Tip: All the options supplied after the image name are GOanna options

-i CFLO_1.fa_gaf.txt: InterProScan XML Parser GAF output file.

-g clfo1_v_insecta_goanna_gaf.tsv: GOanna GAF output file.

-o complete_gaf: output file basename–a .tsv extension will be added

2.11 Intro

- KEGG Orthology Based Annotation System (KOBAS) is a standalone Python application.
- **Consists of two modules:**
    1. annotate–Assigns appropriate KO terms for queried sequences based on a similarity search.
    2. identify–Discovers enriched KO terms among the annotation results by frequency of pathways or statistical significance of pathways.

Results and analysis from the application of KOBAS annotation to the Official gene set v3.0 protein set from *Diaphorina citri* followed by a differential expression analysis was presented at a seminar in the University of Arizona Animal and Comparative Biomedical Sciences in Fall 2020. The slides and video are available online.

2.11.1 Where to Find KOBAS

KOBAS is provided as a Docker container for use on the command line and as a group of apps in the CyVerse Discovery Environment.

- Docker Hub
- KOBAS annotate 3.0.3
- KOBAS identify 3.0.3
- KOBAS annotate and identify 3.0.3
- KOBAS annotate and summarize
- KOBAS summary

Note: Each of these tools accepts a protein FASTA file. For those users with nucloetide sequences some documentation has been provided for using **TransDecoder** (although other tools are also acceptable). The TransDecoder app is available through CyVerse or as a BioContainer for use on the command line.

2.11.2 Getting the KOBAS Databases

Important: As of version 3.0.3_3 you no longer need to download the seq_pep and sqlite3 databases before you run KOBAS on the command line.

If you would like to download them they are still available in the CyVerse Data Store. The CyVerse files can be downloaded with wget, curl or iCommands.

With wget (copy the **public link** from the three dots menu to the right of the file name in the Discovery Environment):

```
wget https://data.cyverse.org/dav-anon/iplant/projects/iplantcollaborative/protein_
↪blast_dbs/kobas/sqlite3.tar
```

Or with curl (copy the **public link** from the three dots menu to the right of the file name in the Discovery Environment):

```
curl -o sqlite3.tar https://data.cyverse.org/dav-anon/iplant/projects/
↪iplantcollaborative/protein_blast_dbs/kobas/sqlite3.tar
```

Or with iCommnads copy the file **path** from the three dots menu to the right of then file name in the Discovery Environment

```
iget /iplant/home/shared/iplantcollaborative/protein_blast_dbs/kobas/sqlite3.tar
```

Once you have the tar files you can extract all the contents or only those you wish using tar. There is no need to unzip the .gz files before you run the tool.

2.11.3 Help and Usage Statement

On the command line the following help statement can be displayed with the option '-h'.

```
Options:
[-h prints this help statement]

[-a runs KOBAS annotate]
KOBAS annotate options:
    -i INFILE can be FASTA or one-per-lineidentifiers. See -t intype for details.
    -s SPECIES 3 or 4 letter species abbreviation (can be found here: ftp://ftp.cbi.
↪pku.edu.cn/pub/KOBAS_3.0_DOWNLOAD/species_abbr.txt or here: https://www.kegg.jp/
↪kegg/catalog/org_list.html)
```

(continues on next page)

(continued from previous page)

```
    -o OUTPUT file (Default is stdout.)
    -t INTYPE (fasta:pro, fasta:nuc, blastout:xml, blastout:tab, id:ncbigi,␣
↪id:uniprot, id:ensembl, id:ncbigene), default fasta:pro
    [-l LIST available species, or list available databases for a specific species]
    [-e EVALUE expect threshold for BLAST, default 1e-5]
    [-r RANK rank cutoff for valid hits from BLAST result, default is 5]
    [-C COVERAGE subject coverage cutoff for BLAST, default 0]
    [-z ORTHOLOG whether only use orthologs for cross-species annotation or not,␣
↪default NO (if only using orthologs, please provide the species abbreviation of␣
↪your input)]
    [-k KOBAS HOME The path to kobas_home, which is the parent directory of sqlite3/␣
↪and seq_pep/. This is the absolute path in the container.]
    [-v BLAST HOME The path to blast_home, which is the parent directory of blastx␣
↪and blastp. This is the absolute path in the container.]
    [-y BLASTDB The path to seq_pep/. This is the absolute path in the container.]
    [-q KOBASDB The path to sqlite3/, This is the absolute path in the container.]
    [-p BLASTP The path to blastp. This is the absolute path in the container.]
    [-x BLASTX The path to blastx. This is the absolute path in the container.]
    [-T number of THREADS to use in BLAST search. Default = 8]

[-g runs KOBAS identify]
    KOBAS identify options:
    -f FGFILE foreground file, the output of annotate
    -b BGFILE background file, species abbreviation, see this list for species codes:␣
↪https://www.kegg.jp/kegg/catalog/org_list.html
    -o OUTPUT file (Default is stdout.)
    [-d DB databases for selection, 1-letter abbreviation separated by "/": K for␣
↪KEGG PATHWAY, n for PID, b for BioCarta, R for Reactome, B for BioCyc, p for␣
↪PANTHER,
           o for OMIM, k for KEGG DISEASE, f for FunDO, g for GAD, N for NHGRI GWAS␣
↪Catalog and G for Gene Ontology, default K/n/b/R/B/p/o/k/f/g/N/]
    [-m METHOD choose statistical test method: b for binomial test, c for chi-square␣
↪test, h for hypergeometric test / Fisher's exact test, and x for frequency list,
           default hypergeometric test / Fisher's exact test
    [-n FDR choose false discovery rate (FDR) correction method: BH for Benjamini and␣
↪Hochberg, BY for Benjamini and Yekutieli, QVALUE, and None, default BH
    [-c CUTOFF terms with less than cutoff number of genes are not used for␣
↪statistical tests, default 5]
    [-k KOBAS HOME The path to kobas_home, which is the parent directory of sqlite3/␣
↪and seq_pep/. This is the absolute path in the container.]
    [-v BLAST HOME The path to blast_home, which is the parent directory of blastx␣
↪and blastp. This is the absolute path in the container.]
    [-y BLASTDB The path to seq_pep/. This is the absolute path in the container.]
    [-q KOBASDB The path to sqlite3/. This is the absolute path in the container.]
    [-p BLASTP The path to blastp. This is the absolute path in the container.]
    [-x BLASTX The path to blastx. This is the absolute path in the container.]

[-j runs both KOBAS annotate and identify]
```

2.12 KOBAS on CyVerse

2.12.1 Accessing KOBAS in the Discovery Environment

1. Create an account on CyVerse (free). The Discovery Environment (DE) the user guide can be found here.

2. Open the CyVerse Discovery Environment (DE) and login with your CyVerse credentials.
3. There are several ways to access the GOanna app:

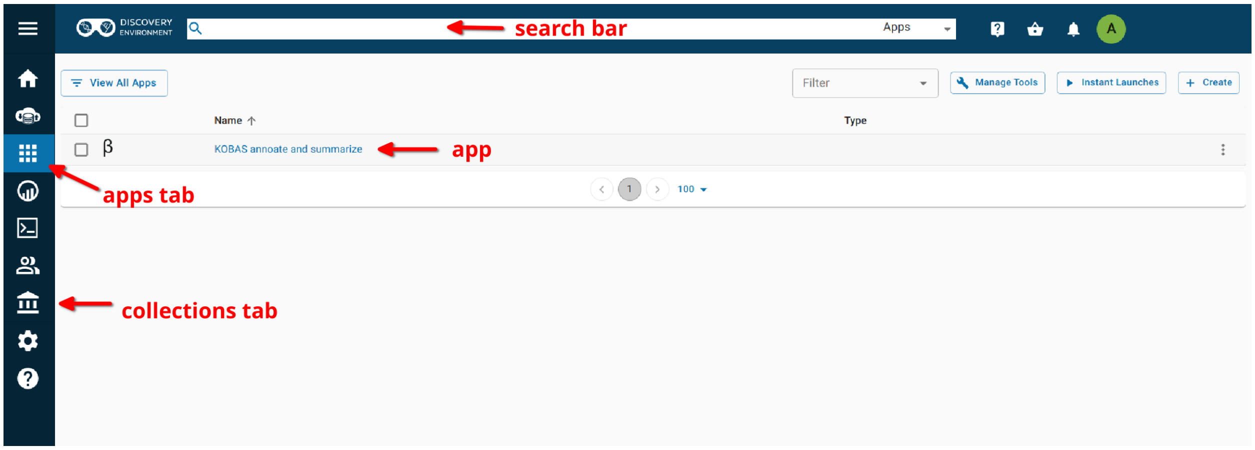
- Use the direct link.
- Search for 'KOBAS in the search bar at the top of the 'apps' tab.
- Follow the AgBase collection (collections tab on left side of DE)

The KOBAS apps are called:

- **NEW** KOBAS annotate and summarize
- KOBAS annotate 3.0.3
- KOBAS identify 3.0.3
- KOBAS annotate and identify 3.0.3

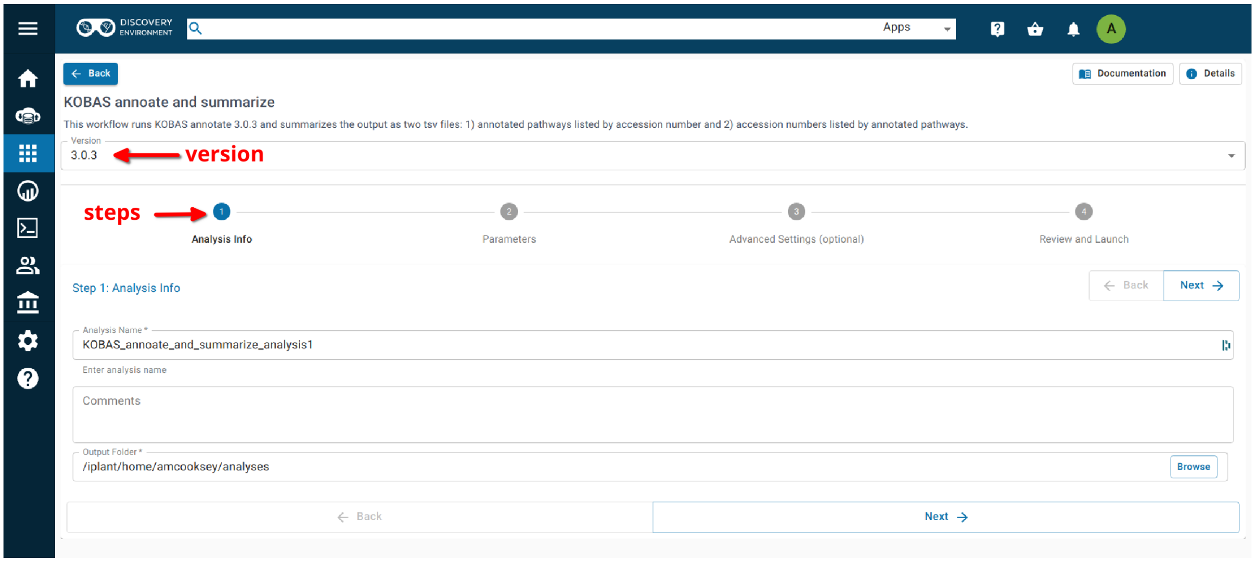
NEW KOBAS annotate and summarize

The new KOBAS annotate and summarize app is version 3.0.3. In addition to the annotate function it also performs a summary step. We recommend using this app rather than the annotate 3.0.3 app.



2.12.2 KOBAS annotate and summarize

Launching the app



KOBAS annoate and summarize

This workflow runs KOBAS annotate 3.0.3 and summarizes the output as two tsv files: 1) annotated pathways listed by accession number and 2) accession numbers listed by annotated pathways.

Version: 3.0.3

Step 1: Analysis Info

Analysis Name *: KOBAS_annoate_and_summarize_analysis1

Enter analysis name

Comments

Output Folder *: /iplant/home/amcooksey/analyses

Step 1. Analysis Info

Analysis Name: KOBAS_annoate_and_summarize_analysis1: This menu is used to name the job you will run so that you can find it later. Analysis Name: The default name is "KOBAS_annoate_and_summarize_analysis1". We recommend changing the 'analysis1' portion of this to reflect the data you are running.

Comments: (Optional) You can add additional information in the comments section to distinguish your analyses further.

Select output folder: This is where your results will be placed. The default (recommended) is your 'analyses' folder.

Step 2. Parameters

Input File: Use the 'browse' button on the right side of the field to navigate to your input file.

Input File Type: Select your input file type from the drop-down list. If your file type isn't there then the app does not support that file type.

Species Code: Enter the species for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html **Not all KEGG species are available through KOBAS.**

If your species of interest is not available then you should choose the code for the closest-related species available

E value: This is the evalue to use in the BLAST search. The default is 1e-5.

Rank: rank cutoff for valid hits from BLAST result. Default is 5.

Covergage: subject coverage cutoff for BLAST. Default is 0.

Ortholog: when checked KOBAS will only use orthologs for cross species annotation.

Output File Name: Provide an output file name .

For information on outputs see Understanding Your Results: Annotate

Step3. Adavanced Settings (optional)

This page allows you specifiy compute requirements for your analysis (e.g. more memory if your analysis is particularly large). You should be able to leave the defaults for most analyses.

Step4. Review and Launch

This will display all of the parameters you have set (other than default). Missing information that is required will displayed in red. Make sure you are happy with your choices and then clicke the 'launch' button at the bottom.

Understanding Your Annotate Results

If all goes well, you should get the following:

- **logs folder:** This folder contains the 'conder_stderr' and 'condor_stdout' files. The files record feedback, progress and, importantly, any errors the app encountered during the analysis. You won't normally need to look at these but they are very helpful in figuring out what may have happened if your output doesn't look like you expected.
- **seq_pep folder:** This folder contains the BLAST database files used in your analysis.
- **sqlite3 folder:** This folder contains the annotation database files used in your analysis
- **<species>.tsv:** This is the tab-delimited output from the BLAST search. It is unlikely that you will need to look at this file.
- **<output_prefix>.txt:** KOBAS-annotate generates a text file with the name you provide. It has two sections.
- **<output_prefix>.txt_KOBAS_acc_pathways.tsv:** A tab-delimited file with accession number and all of the annoations made to that accession.
- **<output_prefix>.txt_KOBAS_pathwyas_acc.tsv:** A tab-delimited file with pathways and of all of the accession annotated with that pathway.

The first section of <output_prefix>.txt looks like this:

```
##dme       Drosophila melanogaster (fruit fly)
##Method: BLAST     Options: evalue <= 1e-05
##Summary:  87 succeed, 0 fail

#Query      Gene ID|Gene name|Hyperlink
lcl|NW_020311285.1_prot_XP_012256083.1_15   dme:Dmel_CG34349|Unc-13-4B|http://www.
↪genome.jp/dbget-bin/www_bget?dme:Dmel_CG34349
lcl|NW_020311286.1_prot_XP_020708336.1_46   dme:Dmel_CG6963|gish|http://www.genome.jp/
↪dbget-bin/www_bget?dme:Dmel_CG6963
lcl|NW_020311285.1_prot_XP_020707987.1_39   dme:Dmel_CG30403||http://www.genome.jp/
↪dbget-bin/www_bget?dme:Dmel_CG30403
```

The second section of <output_prefix>.txt follows a dashed line and looks like this:

```
--------------------


////
Query:                      lcl|NW_020311285.1_prot_XP_012256083.1_15
```

(continues on next page)

(continued from previous page)

```
Gene:                       dme:Dmel_CG34349        Unc-13-4B
Entrez Gene ID:             43002
////
Query:                      lcl|NW_020311286.1_prot_XP_020708336.1_46
Gene:                       dme:Dmel_CG6963 gish
Entrez Gene ID:             49701
Pathway:                    Hedgehog signaling pathway - fly        KEGG PATHWAY    ␣
↪dme04341
////
Query:                      lcl|NW_020311285.1_prot_XP_020707987.1_39
Gene:                       dme:Dmel_CG30403
Entrez Gene ID:             246595
////
Query:                      lcl|NW_020311285.1_prot_XP_020707989.1_40
Gene:                       dme:Dmel_CG6148 Past1
Entrez Gene ID:             41569
Pathway:                    Endocytosis     KEGG PATHWAY    dme04144
                            Hemostasis      Reactome        R-DME-109582
                            Factors involved in megakaryocyte development and␣
↪platelet production   Reactome        R-DME-98323
```

The <output_prefix>.txt_KOBAS_acc_pathways.tsv file looks like this:

XP_018223853.1 Reactome:R-SCE-6782135,KEGG:sce03420,Reactome:R-SCE-113418,Reactome:R-SCE-3700989,Reactome:R-SCE-73894,Reactome:R-SCE-73776,KEGG:sce03022,Reactome:R-SCE-6796648 XP_018222686.1 Reactome:R-SCE-5689603,KEGG:sce03050,Reactome:R-SCE-392499,Reactome:R-SCE-168249,Reactome:R-SCE-597592,Reactome:R-SCE-1236975,Reactome:R-SCE-1236978 XP_018223153.1 KEGG:sce01100,KEGG:sce01110,KEGG:sce01130,KEGG:sce01200,KEGG:sce01230,BioCyc:NONOXIPENT-PWY,BioCyc:PENTOSE-P-PWY,KEGG:sce00030 XP_018220571.1 KEGG:sce01100,KEGG:sce00270,KEGG:sce00480,KEGG:sce00
PWY

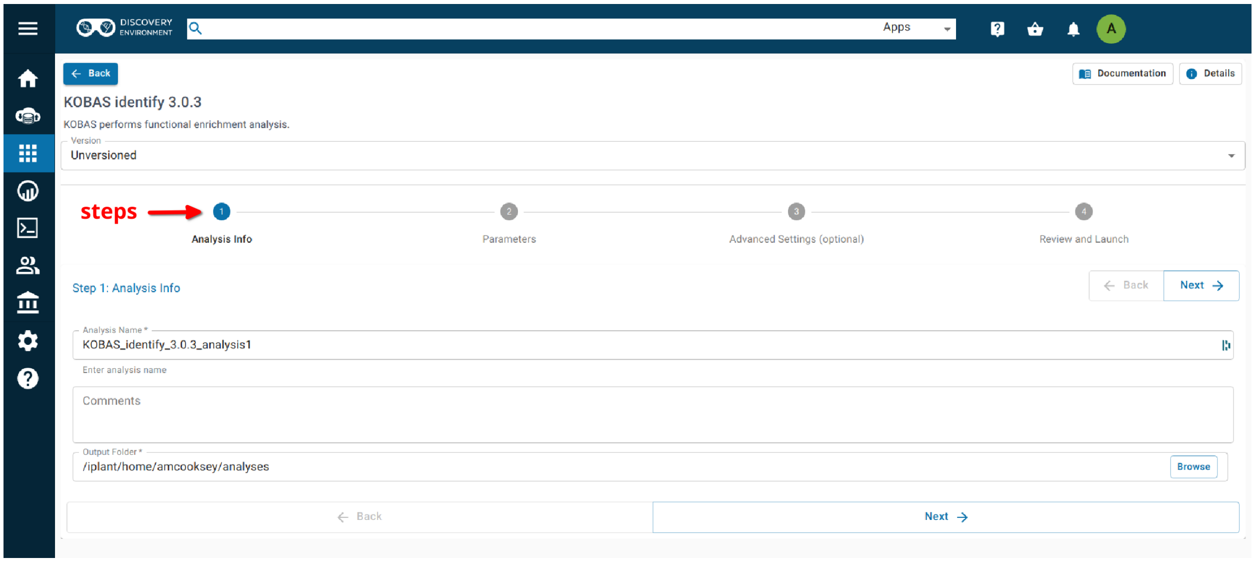
The <output_prefix>.txt_KOBAS_pathwyas_acc.tsv file looks like this:

KEGG:sce01100 XP_018223153.1,XP_018220571.1,XP_018219513.1 Reactome:R-SCE-5688426 XP_018222686.1 KEGG:sce03022 XP_018223853.1 Reactome:R-SCE-75105 XP_018219513.1 Reactome:R-SCE-597592 XP_018222686.1

If your analysis doesn't complete as you expected please look at your 'condor_stderr' and 'condor_stdout' files. If that doesn't clarify the problem contact us at agbase@email.arizona.edu or support@cyverse.org.

2.12.3 KOBAS identify 3.0.3

Launching the App



KOBAS identify 3.0.3

KOBAS performs functional enrichment analysis.

Version: Unversioned

Step 1: Analysis Info

Analysis Name *: KOBAS_identify_3.0.3_analysis1

Enter analysis name

Comments

Output Folder *: /iplant/home/amcooksey/analyses

Step 1. Analysis Info

Analysis Name: KOBAS_identify_3.0.3_analysis_1: This menu is used to name the job you will run so that you can find it later. Analysis Name: The default name is "KOBAS_identify_3.0.3_analysis1". We recommend changing the 'analysis1' portion of this to reflect the data you are running.

Comments: (Optional) You can add additional information in the comments section to distinguish your analyses further.

Select output folder: This is where your results will be placed. The default (recommended) is your 'analyses' folder.

Step 2. Parameters

Foreground File: Use the 'browse' button on the right side of the field to navigate to your input file. This should be the output of KOBAS annotate.

Background: Enter the species for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html

If your species of interest is not available then you should choose the code for the closest-related species available

Cutoff: Annotation terms with less than cutoff number of genes are not used for statistical tests. Default is 5.

Method: Choose the statistical method to be used from the drop-down list. Default is hypergeometric/Fisher's Exact.

FDR: Method for determining false discovery rate. Default is Benjamnini-Hochberg.

Output File Name: Provide an output file name.

Step3. Adavanced Settings (optional)

This page allows you specifiy compute requirements for your analysis (e.g. more memory if your analysis is particularly large). You should be able to leave the defaults for most analyses.

Step4. Review and Launch

This will display all of the parameters you have set (other than default). Missing information that is required will displayed in red. Make sure you are happy with your choices and then clicke the 'launch' button at the bottom.

Understanding Your Identify Results

If all goes well, you should get the following:

- **logs folder:** This folder contains the 'conder_stderr' and 'condor_stdout' files. The files record feedback, progress and, importantly, any errors the app encountered during the analysis. You won't normally need to look at these but they are very helpful in figuring out what may have happened if your output doesn't look like you expected.
- **sqlite3 folder:** This folder contains the annotation database files used in your analysis
- **<output_file_name_you_provided>:** KOBAS identify generates a text file with the name you provide.

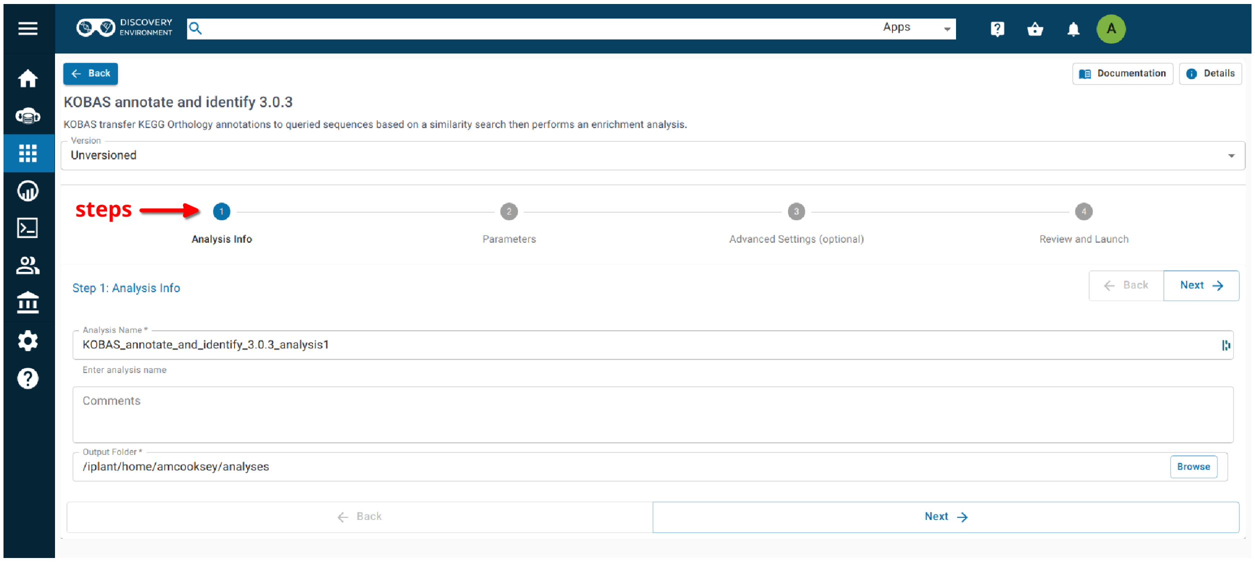
```
##Databases: PANTHER, KEGG PATHWAY, Reactome, BioCyc
##Statistical test method: hypergeometric test / Fisher's exact test
##FDR correction method: Benjamini and Hochberg

#Term       Database        ID      Input number    Background number       P-Value␣
↪Corrected P-Value       Input   Hyperlink
Hedgehog signaling pathway - fly    KEGG PATHWAY    dme04341        12      33      3.
↪20002656734e-18       1.76001461204e-16       lcl|NW_020311286.1_prot_XP_012256678.
↪1_51|lcl|NW_020311286.1_prot_XP_025602973.1_48|lcl|NW_020311286.1_prot_XP_012256683.
↪1_52|lcl|NW_020311286.1_prot_XP_012256679.1_55|lcl|NW_020311286.1_prot_XP_012256674.
↪1_54|lcl|NW_020311286.1_prot_XP_020708336.1_46|lcl|NW_020311285.1_prot_XP_012256108.
↪1_32|lcl|NW_020311286.1_prot_XP_012256682.1_53|lcl|NW_020311286.1_prot_XP_025603025.
↪1_47|lcl|NW_020311286.1_prot_XP_020708334.1_49|lcl|NW_020311285.1_prot_XP_012256109.
↪1_33|lcl|NW_020311286.1_prot_XP_020708333.1_50 http://www.genome.jp/kegg-bin/show_
↪pathway?dme04341/dme:Dmel_CG6963%09red/dme:Dmel_CG6054%09red
Hedgehog signaling pathway  PANTHER P00025  6       13      3.6166668094e-10        9.
↪94583372585e-09       lcl|NW_020311286.1_prot_XP_025602279.1_78|lcl|NW_020311286.1_
↪prot_XP_025602289.1_76|lcl|NW_020311286.1_prot_XP_025602264.1_79|lcl|NW_020311285.1_
↪prot_XP_012256108.1_32|lcl|NW_020311285.1_prot_XP_012256109.1_33|lcl|NW_020311286.1_
↪prot_XP_012256943.1_77     http://www.pantherdb.org/pathway/pathwayDiagram.jsp?
↪catAccession=P00025
Signaling by NOTCH2 Reactome        R-DME-1980145   3       8       2.00259649553e-05␣
↪      0.000275357018136       lcl|NW_020311285.1_prot_XP_012256118.1_28|lcl|NW_
↪020311285.1_prot_XP_012256117.1_27|lcl|NW_020311285.1_prot_XP_012256119.1_26   ␣
↪http://www.reactome.org/cgi-bin/eventbrowser_st_id?ST_ID=R-DME-1980145
```

If your analysis doesn't complete as you expected please look at your 'condor_stderr' and 'condor_stdout' files. If that doesn't clarify the problem contact us at agbase@email.arizona.edu or support@cyverse.org.

2.12.4 KOBAS annotate and identify 3.0.3

Launching the App



KOBAS annotate and identify 3.0.3

KOBAS transfer KEGG Orthology annotations to queried sequences based on a similarity search then performs an enrichment analysis.

Version: Unversioned

Step 1: Analysis Info

Analysis Name *: KOBAS_annotate_and_identify_3.0.3_analysis1

Enter analysis name

Comments

Output Folder *: /iplant/home/amcooksey/analyses

This app runs both the annotate and identify analyses together as a convenience for user who wish to run both steps.

Step 1. Analysis Info

Analysis Name: KOBAS_annotate_and_identify_3.0.3_analysis_1: This menu is used to name the job you will run so that you can find it later. Analysis Name: The default name is "KOBAS_annotate_identify_3.0.3_analysis1". We recommend changing the 'analysis1' portion of this to reflect the data you are running.

Comments: (Optional) You can add additional information in the comments section to distinguish your analyses further.

Select output folder: This is where your results will be placed. The default (recommended) is your 'analyses' folder.

Step 2. Parameters

Input

Input File: Use the 'browse' button on the right side of the field to navigate to your input file.

Input File Type: Select your input file type from the drop-down list. If your file type isn't there then the app does not support that file type.

Annotate Options

Species Code: Enter the species for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html

If your species of interest is not available then you should choose the code for the closest-related species available

E value: This is the evalue to use in the BLAST search. The default is 1e-5.

Rank: rank cutoff for valid hits from BLAST result. Default is 5.

Covergage: subject coverage cutoff for BLAST. Default is 0.

Ortholog: when checked KOBAS will only use orthologs for cross species annotation.

Identify Options

Cutoff: Annotation terms with less than cutoff number of genes are not used for statistical tests. Default is 5.

Method: Choose the statistical method to be used from the drop-down list. Default is hypergeometric/Fisher's Exact.

FDR: Method for determining false discovery rate. Default is Benjamnini-Hochberg.

Output

Output File Basename: This will the the prefix of your output files.

Step3. Adavanced Settings (optional)

This page allows you specifiy compute requirements for your analysis (e.g. more memory if your analysis is particularly large). You should be able to leave the defaults for most analyses.

Step4. Review and Launch

This will display all of the parameters you have set (other than default). Missing information that is required will displayed in red. Make sure you are happy with your choices and then clicke the 'launch' button at the bottom.

If your analysis doesn't complete as you expected please look at your 'condor_stderr' and 'condor_stdout' files. If that doesn't clarify the problem contact us at agbase@email.arizona.edu or support@cyverse.org.

2.13 KOBAS on the Command Line

2.13.1 Getting the databases

No longer required as of version 3.0.3_3.

2.13.2 Container Technologies

KOBAS is provided as a Docker container.

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

There are two major containerization technologies: **Docker** and **Apptainer (Singularity)**.

Docker containers can be run with either technology.

2.13.3 Running KOBAS using Docker

About Docker

- Docker must be installed on the computer you wish to use for your analysis.
- To run Docker you must have 'root' (admin) permissions (or use sudo).
- Docker will run all containers as 'root'. This makes Docker incompatible with HPC systems (see Singularity below).
- Docker can be run on your local computer, a server, a cloud virtual machine etc.
- For more information on installing Docker on other systems: Installing Docker.

Getting the KOBAS container

The KOBAS tool is available as a Docker container on Docker Hub: KOBAS container

The container can be pulled with this command:

```
docker pull agbase/kobas:3.0.3_3
```

Remember

You must have root permissions or use sudo, like so:

sudo docker pull agbase/kobas:3.0.3_3

Getting the Help and Usage Statement

```
sudo docker run --rm agbase/kobas:3.0.3_3 -h
```

Tip: The /work-dir directory is built into this container and should be used to mount your data.

KOBAS can perform two tasks - annotate (-a) - identify (enrichment) (-g)

KOBAS can also run both tasks with a single command (-j).

Annotate Example Command

```
sudo docker run \
--rm \
-v $(pwd):/work-dir \
agbase/kobas:3.0.3_3 \
-a \
-i GCF_001298625.1_SEUB3.0_protein.faa \
-s sce \
-t fasta:pro \
-o GCF_001298625.1
```

Command Explained

sudo docker run: tells docker to run

–rm: removes the container when the analysis has finished. The image will remain for future use.

-v $(pwd):/work-dir: mounts my current working directory on the host machine to '/work-dir' inside the container

agbase/kobas:3.0.3_3: the name of the Docker image to use

Tip: All the options supplied after the image name are KOBAS options

-a: Tells KOBAS to run the 'annotate' process.

-i GCF_001298625.1_SEUB3.0_protein.faa: input file (protein FASTA).

-s sce: Enter the species code for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html

If your species of interest is not available then you should choose the code for the closest-related species available

-t: input file type; in this case, protein FASTA.

-o GCF_001298625.1: prefix for the output file names

Reference *Understanding results*.

Identify Example Command

```
sudo docker run \
--rm \
-v $(pwd):/work-dir \
agbase/kobas:3.0.3_3 \
-g \
-f GCF_001298625.1_SEUB3.0_protein.faa \
-b sce \
-o ident_out
```

Command Explained

sudo docker run: tells docker to run

–rm: removes the container when the analysis has finished. The image will remain for future use.

-v $(pwd):/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

agbase/kobas:3.0.3_3: the name of the Docker image to use

Tip: All the options supplied after the image name are KOBAS options

-g: Tells KOBAS to runt he 'identify' process.

-f GCF_001298625.1_SEUB3.0_protein.faa: output file from KOBAS annotate

-b sce: background; enter the species code for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html

If your species of interest is not available then you should choose the code for the closest-related species available

-o ident_out: basename of output file

Reference *Understanding results*.

Annotate and Identify Pipeline Example Command

```
sudo docker run \
--rm \
-v $(pwd):/work-dir \
agbase/kobas:3.0.3_3 \
-j \
-i GCF_001298625.1_SEUB3.0_protein.faa \
-s sce \
-t fasta:pro
-o GCF_001298625.1
```

Command Explained

sudo docker run: tells docker to run

–rm: removes the container when the analysis has finished. The image will remain for future use.

-v $(pwd):/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

agbase/kobas:3.0.3_3: the name of the Docker image to use

Tip: All the options supplied after the image name are KOBAS options

-j: Tells KOBAS to run both the 'annotate' and 'identify' processes.

-i GCF_001298625.1_SEUB3.0_protein.faa: input file (protein FASTA)

-s sce: Enter the species code for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html

If your species of interest is not available then you should choose the code for the closest-related species available

-t: input file type; in this case, protein FASTA.

-o GCF_001298625.1: basename of output files

Note: This pipeline will automatically use the output of ‘annotate’ as the -f foreground input for ‘identify’. This will also use your species option as the -b background input for ‘identify’.

Reference *Understanding results*.

2.13.4 Running KOBAS using Singularity

About Singularity (now Apptainer)

- does not require ‘root’ permissions
- runs all containers as the user that is logged into the host machine
- HPC systems are likely to have Singularity installed and are unlikely to object if asked to install it (no guarantees).
- can be run on any machine where it is installed
- more information about installing Singularity
- This tool was tested using Singularity 3.10.2.

HPC Job Schedulers

Although Singularity can be installed on any computer this documentation assumes it will be run on an HPC system. The tool was tested on a Slurm system and the job submission scripts below reflect that. Submission scripts will need to be modified for use with other job scheduler systems.

Getting the KOBAS container

The KOBAS tool is available as a Docker container on Docker Hub: KOBAS container

Example Slurm script:

```
#!/bin/bash
#SBATCH --job-name=kobas
#SBATCH --ntasks=8
#SBATCH --time=2:00:00
#SBATCH --partition=short
#SBATCH --account=nal_genomics

module load singularity

cd /location/where/your/want/to/save/file

singularity pull docker://agbase/kobas:3.0.3_3
```

Running KOBAS with Data

Tip: There /work-dir directory is built into this container and should be used to mount data.

Example Slurm Script for Annotate Process

```
#!/bin/bash
#SBATCH --job-name=kobas
#SBATCH --ntasks=8
#SBATCH --time=2:00:00
#SBATCH --partition=short
#SBATCH --account=nal_genomics

module load singularity

cd /directory/you/want/to/work/in

singularity run \
-B /directory/you/want/to/work/in:/work-dir \
/path/to/your/copy/kobas_3.0.3_3.sif \
-a \
-i GCF_001298625.1_SEUB3.0_protein.faa \
-s sce \
-t fasta:pro \
-o GCF_001298625.1
```

Command Explained

singularity run: tells Singularity to run

-B /project/nal_genomics/amanda.cooksey/protein_sets/saceub/KOBAS:/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

/path/to/your/copy/kobas_3.0.3_3.sif: the name of the Singularity image to use

Tip: All the options supplied after the image name are KOBAS options

-a: Tells KOBAS to run the 'annotate' process.

-i GCF_001298625.1_SEUB3.0_protein.faa: input file (protein FASTA)

-s sce: Enter the species for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html

If your species of interest is not available then you should choose the code for the closest-related species available

-t: input file type; in this case, protein FASTA.

-o GCF_001298625.1: name of output file

Reference *Understanding results*.

Example Slurm Script for Identify Process

```
#!/bin/bash
#SBATCH --job-name=kobas
#SBATCH --ntasks=8
#SBATCH --time=2:00:00
#SBATCH --partition=short
#SBATCH --account=nal_genomics

module load singularity

cd /location/where/your/want/to/save/file

singularity pull docker://agbase/kobas:3.0.3_3

singularity run \
-B /directory/you/want/to/work/in:/work-dir \
kobas_3.0.3_3.sif \
-g \
-f GCF_001298625.1_SEUB3.0_protein.faa \
-b sce \
-o ident_out
```

Command Explained

singularity run: tells Singularity to run

-B /project/nal_genomics/amanda.cooksey/protein_sets/saceub/KOBAS:/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

kobas_3.0.3_3.sif: the name of the Singularity image to use

Tip: All the options supplied after the image name are KOBAS options

-g: Tells KOBAS to run the 'identify' process.

-f GCF_001298625.1_SEUB3.0_protein.faa: output file from 'annotate'

-b sce: background; enter the species for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html

If your species of interest is not available then you should choose the code for the closest-related species available

-o ident_out: name of output file

Reference *Understanding results*.

Example Slurm Script for Annotate and Identify Pipeline

```
#!/bin/bash
#SBATCH --job-name=kobas
#SBATCH --ntasks=8
#SBATCH --time=2:00:00
#SBATCH --partition=short
#SBATCH --account=nal_genomics

module load singularity

cd /location/where/your/want/to/save/file

singularity pull docker://agbase/kobas:3.0.3_3

singularity run \
-B /directory/you/want/to/work/in:/work-dir \
kobas_3.0.3_3.sif \
-j \
-i GCF_001298625.1_SEUB3.0_protein.faa \
-s sce \
-t fasta:pro \
-o GCF_001298625.1
```

Command Explained

singularity run: tells Singularity to run

-B /rsgrps/shaneburgess/amanda/i5k/kobas:/work-dir: mounts my current working directory on the host machine to '/work-dir' in the container

kobas_3.0.3_3.sif: the name of the Singularity image to use

Tip: All the options supplied after the image name are KOBAS options

-j: Tells KOBAS to runt he 'annotate' process.

-i GCF_001298625.1_SEUB3.0_protein.faa: input file (protein FASTA)

-s sce: Enter the species for the species of the sequences in your input file.

Note: If you don't know the code for your species it can be found here: https://www.kegg.jp/kegg/catalog/org_list.html

If your species of interest is not available then you should choose the code for the closest-related species available

-t: input file type; in this case, protein FASTA.

-o GCF_001298625.1: name of output file

Note: This pipeline will automatically use the output of 'annotate' as the -f foreground input for 'identify'. This will also use your species option as the -b background input for 'identify'.

2.13.5 Understanding Your Results

Annotate

If all goes well, you should get the following:

- **<species>.tsv:** This is the tab-separated output from the BLAST search. It is unlikely that you will need to look at this file.
- **<basename>:** KOBAS-annotate generates a text file with the name you provide. It has two sections (detailed below).
- **<basename>_KOBAS_acc_pathways.tsv:** Our post-processing script creates this tab-separated file. It lists each accession from your data and all of the pathways to which they were annotated.
- **<basename>_KOBAS_pathways_acc.tsv:** Our post-processing script creates this tab-separated file. It lists each pathway annotated to your data with all of the accessions annotated to that pathway.

The <basename> file has two sections. The first section looks like this:

```
#Query      Gene ID|Gene name|Hyperlink
XP_018220118.1      sce:YMR059W|SEN15|http://www.genome.jp/dbget-bin/www_bget?
 ↪sce:YMR059W
XP_018221352.1      sce:YJR050W|ISY1, NTC30, UTR3|http://www.genome.jp/dbget-bin/www_
 ↪bget?sce:YJR050W
XP_018224031.1      sce:YDR513W|GRX2, TTR1|http://www.genome.jp/dbget-bin/www_bget?
 ↪sce:YDR513W
XP_018222559.1      sce:YFR024C-A|LSB3, YFR024C|http://www.genome.jp/dbget-bin/www_
 ↪bget?sce:YFR024C-A
XP_018221254.1      sce:YJL070C||http://www.genome.jp/dbget-bin/www_bget?sce:YJL070C
```

The second section follows a dashed line and looks like this:

```
////
Query:                  XP_018222878.1
Gene:                   sce:YDL220C     CDC13, EST4
Entrez Gene ID:         851306
////
Query:                  XP_018219412.1
Gene:                   sce:YOR204W     DED1, SPP81
Entrez Gene ID:         854379
Pathway:                Innate Immune System    Reactome        R-SCE-168249
                        Immune System   Reactome        R-SCE-168256
                        Neutrophil degranulation        Reactome        R-SCE-6798695
```

<basename>_KOBAS_acc_pathways.tsv looks like this:

```
XP_018220118.1      BioCyc:PWY-6689
XP_018221352.1      Reactome:R-SCE-6782135,KEGG:sce03040,Reactome:R-SCE-73894,
 ↪Reactome:R-SCE-5696398,Reactome:R-SCE-6782210,Reactome:R-SCE-6781827
XP_018224031.1      BioCyc:GLUT-REDOX-PWY,BioCyc:PWY3O-592
```

<basename>_KOBAS_pathways_acc.tsv looks like this:

```
BioCyc:PWY3O-0  XP_018222002.1,XP_018222589.1
KEGG:sce00440   XP_018222406.1,XP_018219751.1,XP_018222229.1
Reactome:R-SCE-416476   XP_018223583.1,XP_018221814.1,XP_018222685.1,XP_018220832.1,
 ↪XP_018219073.1,XP_018218776.1,XP_018223466.1,XP_018223545.1,XP_018222256.1
Reactome:R-SCE-418346   XP_018220070.1,XP_018221774.1,XP_018221826.1,XP_018220071.1,
 ↪XP_018222218.1,XP_018220541.1,XP_018219550.1
```

(continues on next page)

(continued from previous page)

Identify

If all goes well, you should get the following:

- **<output_file_name_you_provided>:** KOBAS identify generates a text file with the name you provide.

```
##Databases: PANTHER, KEGG PATHWAY, Reactome, BioCyc
##Statistical test method: hypergeometric test / Fisher's exact test
##FDR correction method: Benjamini and Hochberg

#Term   Database        ID      Input number    Background number       P-Value␣
↪Corrected P-Value       Input   Hyperlink
Metabolic pathways      KEGG PATHWAY    sce01100        714     754     0.
↪00303590229485        0.575578081959  XP_018221856.1|XP_018220917.1|XP_018222719.1|.
↪..link
Metabolism      Reactome        R-SCE-1430728   419     438     0.0147488189928 0.
↪575578081959  XP_018221856.1|XP_018221742.1|XP_018219354.1|XP_018221740.1|...link
Immune System   Reactome        R-SCE-168256    304     315     0.0267150787723 0.
↪575578081959  XP_018223955.1|XP_018222962.1|XP_018223268.1|XP_018222956.1|...link
```

Contact us.