enTAP Documentation

Release 0.5.0

Alex Hart, Jill Wegrzyn

Apr 15, 2018

Contents

EnTAP is an eukaryotic non-model annotation pipeline developed by Alexander Hart and Dr. Jill Wegrzyn of the Plant Computational Genomics Lab at the University of Connecticut.

CHAPTER 1

EnTAP Introduction

The Eukaryotic Non-Model Transcriptome Annotation Pipeline (*EnTAP*) is designed to improve the accuracy, speed, and flexibility of functional gene annotation for de novo assembled transcriptomes in non-model eukaryotes.

This software package addresses the fragmentation and related assembly issues that result in inflated transcript estimates and poor annotation rates. Following filters applied through assessment of true expression and frame selection, open-source tools are leveraged to functionally annotate the translated proteins.

Downstream features include fast similarity search across multiple databases, protein domain assignment, orthologous gene family assessment, Gene Ontology term assignment, and KEGG pathway annotation.

The final annotation integrates across multiple databases and selects an optimal assignment from a combination of weighted metrics describing similarity search score, taxonomic relationship, and informativeness. Researchers have the option to include additional filters to identify and remove potential contaminants and prepare the transcripts for enrichment analysis. This fully featured pipeline is easy to install, configure, and runs much faster than comparable functional annotation packages. It is developed to contend with many of the issues in existing software solutions.

EnTAP is optimized to generate extensive functional information for the gene space of organisms with limited or poorly characterized genomic resources.

1.1 Pipeline Stages:

- **Transcriptome Filtering: designed to remove assembly artifacts and identify true CDS (complete and partial genes)**

    1. Expression Filtering (RSEM)
    2. Frame Selection (GeneMARKS-T)

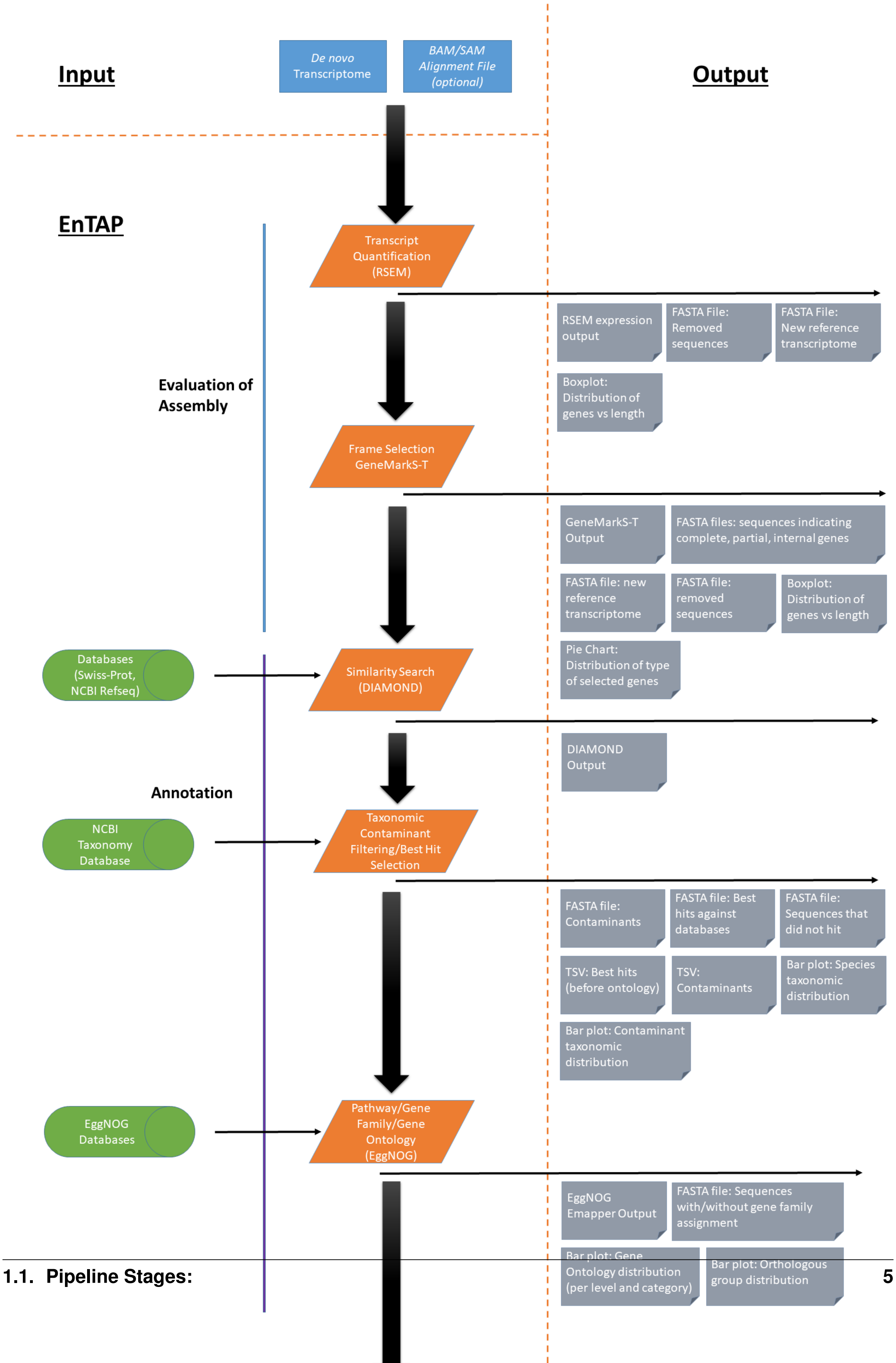
- **Annotation**

    3. Similarity Search: optimized search against user-selected databases (DIAMOND).
    4. Contaminant Filtering and Best Hit Selection: selects final annotation and identifies potential contaminants

5. Orthologous Group Assignment: independent assignment of translated protein sequences to gene families (eggNOG). Includes protein domains (SMART/Pfam), Gene Ontology (GO) terms, and KEGG pathway assignment.
6. InterProScan (optional): sequence search against the families of InterPro databases to assign protein domains, Gene Ontology terms, and pathway information

Note: All of the software integrated into this pipeline are packaged within the EnTAP repository with the exception of GeneMarkS-T (free for academic use). Installation and usage of EnTAP is documented in this guide.

The figure below represents the typical EnTAP pipeline


Input
De novo Transcriptome
BAM/SAM Alignment File (optional)
Output
EnTAP
Evaluation of Assembly
Transcript Quantification (RSEM)
RSEM expression output
FASTA File: Removed sequences
FASTA File: New reference transcriptome
Boxplot: Distribution of genes vs length
Frame Selection GeneMarkS-T
GeneMarkS-T Output
FASTA files: sequences indicating complete, partial, internal genes
FASTA file: new reference transcriptome
FASTA file: removed sequences
Boxplot: Distribution of genes vs length
Pie Chart: Distribution of type of selected genes
Databases (Swiss-Prot, NCBI Refseq)
Similarity Search (DIAMOND)
DIAMOND Output
Annotation
NCBI Taxonomy Database
Taxonomic Contaminant Filtering/Best Hit Selection
FASTA file: Contaminants
FASTA file: Best hits against databases
FASTA file: Sequences that did not hit
TSV: Best hits (before ontology)
TSV: Contaminants
Bar plot: Species taxonomic distribution
Bar plot: Contaminant taxonomic distribution
EggNOG Databases
Pathway/Gene Family/Gene Ontology (EggNOG)
EggNOG Emapper Output
FASTA file: Sequences with/without gene family assignment
Bar plot: Gene Ontology distribution (per level and category)
Bar plot: Orthologous group distribution

1.2 Citations:

[1] **A. Mitchell, H.-Y. Chang, and L. Daugherty, "The InterPro protein families database: the** classification resource after 15 years," Nucleid Acids Research, vol. 43, no. D1, 2015.

[2] **B. Buchfink, Xie C., D. Huson, "Fast and sensitive protein alignment using** DIAMOND", Nature Methods 12, 59-60 (2015).

[3] **eggNOG 4.5: a hierarchical orthology framework with improved functional** annotations for eukaryotic, prokaryotic and viral sequences. Jaime Huerta-Cepas, Damian Szklarczyk, Kristoffer Forslund, Helen Cook, Davide Heller, Mathias C. Walter, Thomas Rattei, Daniel R. Mende, Shinichi Sunagawa, Michael Kuhn, Lars Juhl Jensen, Christian von Mering, and Peer Bork. Nucl. Acids Res. (04 January 2016) 44 (D1): D286-D293. doi: 10.1093/nar/gkv1248

[4] **G. O. Consortium, "Gene Ontology Consortium: going forward," (in eng), Nucleic Acids Res,** vol. 43, no. Database issue, pp. D1049-56, Jan 2015.

[5] **J. Huerta-Cepas et al., "Fast genome-wide functional annotation through orthology** assignment by eggNOG-mapper," (in eng), Mol Biol Evol, Apr 2017.

[6] **M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima, "KEGG:** new perspectives on genomes, pathways, diseases and drugs," (in eng), Nucleic Acids Res, vol. 45, no. D1, pp. D353-D361, Jan 2017

[7] **B. Li and C. N. Dewey, "RSEM: accurate transcript quantification from** RNA-Seq data with or without a reference genome," (in eng), BMC Bioinformatics, vol. 12, p. 323, Aug 2011.

[8] **P. Jones et al., "InterProScan 5: genome-scale protein function classification," (in eng),** Bioinformatics, vol. 30, no. 9, pp. 1236-40, May 2014.

[9] **S. Tang, A. Lomsadze, and M. Borodovsky, "Identification of protein coding regions in** RNA transcripts," Nucleic Acids Research, 2015

1.2.1 Software contained or used within this pipeline:

- RSEM
- DIAMOND
- EggNOG
- GeneMarkS-T
- InterProScan

CHAPTER 2

Installation

EnTAP is packaged with all of the software necessary to fully annotate a set of transcripts. It is optimized to allow a single-command execution for all steps in the pathway, including paramterization by the user. EnTAP does not have a graphical user interface but it does generate visual summaries for the user at each stage as well as detailed summary files and logs.

1. System Requirements
2. *Dependency Check*
3. *Pipeline Software*
4. *EnTAP*

Before full EnTAP installation, dependencies must be checked to see if they are included in your system (many are by default) and the accompanying pipeline software will need to be installed (unless is already present on the system).

2.1 System Requirements

- Operating System
  - UNIX-based systems
  - Tested on 64 bit systems: ubuntu 16.04, Rocks 6.1, Centos 6.3
- Storage Minimum
  - Gene Ontology References: 6Mb
  - NCBI Taxonomy: 400Mb
  - EggNOG Database: 30Gb
  - DIAMOND Databases: ~13Gb (with RefSeq Complete Protein + Uniprot Swiss-Prot)
  - Additional storage for files generated depending on transcriptome size (at least 5 times the size)
  - Typical total: 48Gb (with a fully processed, 30,000 sequence transcriptome)

- Memory
    - At least 16 Gb of RAM (will very depending on DIAMOND database sizes)

2.2 Dependency Check

Before continuing on in the installation process, ensure that the following dependencies are fully installed on your system:

- C++11 compiler (GCC 4.8.1 or later)
- CMake (2.8 or later)
- Boost C++ Libraries (1.50 or later)
    - Ensure this is installed with the C++11 compiler
- Python (2.7.12 or later) with support for the following modules
    - SQLITE
    - Matplotlib (figures generated by EnTAP)
- Unix wget (generally included in most distros)
- Unix gzip (generally included in most distros)

2.3 Pipeline Software

EnTAP leverages several software distributions within the pipeline to provide the best quality annotations. The packages used (and their tested versions) can be seen below. This is not to say that newer versions will not be compatible, however they have not been tested yet with EnTAP.

Note: If the software is already installed on your system, this stage can be skipped

Software:

- RSEM (Expression Filtering with alignment file): version 1.3.0
- GeneMarkS-T (Frame Selection): version 5.1
- DIAMOND (Similarity Search): version 0.8.31
- EggNOG-Emapper (Orthologous Group Assignment): version 0.7.4.1-beta
- InterProScan (Protein Databases): This is not included within the EnTAP repo, but is supported

If you have downloaded the full repository from the GitLab page, each of these (with the exception of GeneMarkS-T) are contained within the /libs directory. GeneMarkS-T must be acquired from the website linked previously due to licensing (free for academic use).

RSEM and DIAMOND both require compilation from source code while EggNOG-Emapper or GeneMarkS-T does not. To compile these, run the script within the main directory:

```
./setup.sh
```

This will make+install DIAMOND and make RSEM.

Warning: Ensure that DIAMOND was properly installed (global installation required by EggNOG-Emapper)

If there are any problems with the setup script, installation steps can be found on the GitHub pages for each.

2.4 EnTAP Installation

Once dependencies and pipeline software have been installed, you can now continue to install EnTAP!

First, download and extract the latest release(tagged) version from GitLab: https://gitlab.com/EnTAP/EnTAP/tags

Within the main directory, execute the following command:

```
cmake CMakeLists.txt
```

This will generate a MakeFile. Then execute:

```
make
```

Or to install to a destination directory:

```
cmake CMakeLists.txt -DCMAKE_INSTALL_PREFIX=/destination/dir
```

```
make install
```

This will complete the installation process. You are ready to start using EnTAP!

CHAPTER 3

Basic Usage

EnTAP has two stages of execution, *configuration* and *run*. Configuration should be completed before the first run and everytime any of the source databases have been updated by the user. This should also be run if you would like to include the latest version of the NCBI Taxonomy database and the Gene Ontology database. All of these are updated regularly at the source and you can ensure you have the most recent version by running configuration before your annotation runs.

3.1 Configuration

Configuration is the first stage of EnTAP that will download and configure the necessary databases for full functionality. This is run if you would like to change/update the databases that EnTAP is reading from. I'll break this up into two sections, *folder hierarchy* and *usage*. The folder hierarchy section will just describe how everything is 'typically' setup with EnTAP, however these paths can be easily changed in the entap_config.txt (more on that later!). The usage section will go over the basic usage during the Configuration stage of EnTAP.

3.1.1 Folder Hierarchy

The EnTAP folder organization is referred to as the execution directory where all files will be made available. This is essentially the hierarchy that was downloaded from the repository.

The /EnTAP directory contains:

- /EnTAP /libs
- /EnTAP /src
- /EnTAP /bin (created during configuration)
- /EnTAP /databases (created during configuration)

In addition to some other files/directories.

Recognition of EnTAP databases, src files, and execution accompanying pipeline software can rely on this 'default' directory hierarchy. However, any necessary files/directories can be changed from the default with the entap_config.txt file (by specifying the paths flag).

Warning: Ensure you are pointing to the correct paths if not using the defaults!

The entap_config.txt file mentioned above has the following defaults:

- diamond_exe_path=/EnTAP/libs/diamond-0.8.31/bin/diamond
- rsem_exe_path=/EnTAP/libs/RSEM-1.3.0 (this is a path to the directory)
- genemarkst_exe_path=/EnTAP/libs/gmst_linux_64/gmst.pl
- eggnog_exe_path=/EnTAP/libs/eggnog-mapper/emapper.py
- eggnog_download_exe=/EnTAP/libs/eggnog-mapper/download_eggnog_data.py
- eggnog_database=/EnTAP/libs/eggnog-mapper/data/eggnog.db (downloaded during Configuration)
- entap_tax_bin_database=/EnTAP/bin/ncbi_tax_bin.entp (binary version, downloaded during Configuration)
- entap_tax_text_database=/EnTAP/databases/ncbi_tax.entp (text version, downloaded during Configuration)
- entap_tax_download_script=/EnTAP/src/download_tax.pl
- entap_go_database=/EnTAP/bin/go_term.entp (binary version, downloaded during Configuration)
- entap_graphing_script=/EnTAP/src/entap_graphing.py
- interpro_exe_path=interproscan.sh

These can be changed to whichever path you would prefer. If something is globally installed, just put how you'd normally run the software after the '=', such as 'diamond' for DIAMOND. EnTAP will recognize these paths first and they will override defaults.

This configuration file will be automatically detected if it is in the working directory, otherwise the path to it can be specified through the paths flag. The config file is mandatory! The entap_tax_text_database will only be used in Configuration and thus is not needed in Execution.You can specify this if you downloaded the text version of the database (from python script) separately

Note: Be sure you set the paths before moving on (besides the databases that haven't been downloaded yet)!

3.1.2 Usage

All source databases must be provided in FASTA format so that they can be indexed for use by DIAMOND. This can be completed independent of EnTAP with DIAMOND (- - makedb flag) or as part of the configuration phase of EnTAP. While any FASTA database can be used, it is recommended to use NCBI (Genbank) sourced databases such as RefSeq databases or NR. In addition, EnTAP can easily accept EBI databases such as UniProt/SwissProt.

EnTAP can recognize the species information from these header formats ONLY (NCBI and UniProt):

- [homo sapiens]
- OS=homo sapiens

If the individual FASTAs in a custom database you create do not adhere to one of these two formats, it will not be possible to weight taxonomic or contaminant status from them.

The following FTP sites contain common reference databases that EnTAP can recognize:

- RefSeq: ftp://ftp.ncbi.nlm.nih.gov/refseq/release/complete/
- Plant RefSeq: ftp://ftp.ncbi.nlm.nih.gov/refseq/release/plant/
- Mammalian RefSeq: ftp://ftp.ncbi.nlm.nih.gov/refseq/release/vertebrate_mammalian/
- NR: ftp://ftp.ncbi.nlm.nih.gov/blast/db/
- SwissProt: ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.fasta.gz
  - Reviewed
- TrEMBL: ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_trembl.fasta.gz
  - Unreviewed

Both Uniprot databases (SwissProt and TrEMBL) can be downloaded on a Unix system through the following command:

```
wget ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/
↪complete/uniprot_sprot.fasta.gz
```

Or, for the TrEMBL database:

```
wget ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/
↪complete/uniprot_trembl.fasta.gz
```

Alternatively, the NCBI databases must be downloaded in separate, smaller files, and concatenated together. As an example, the following commands will download and combine the NR database files:

Download:

```
wget ftp://ftp.ncbi.nlm.nih.gov/blast/db/nr.*.tar.gz
```

Decompress/Concatenate:

```
tar -xvzf nr.*

cat nr.* > nr_database.fasta
```

It is generally recommended that a user select at least three databases with varying levels of curation. Unless the species is very non-model (i.e. does not have close relatives in databases such as RefSeq, it is not necessary to use the full NR database which is less curated).

To run configuration with a sample database, the command is as follows:

```
EnTAP --config -d path/to/database
```

This stage must be done at least once prior to *running*. Once the database is configured, you need not do it again unless you updated your original database or plan on configuring several others.

Note: If you already have DIAMOND (.dmnd) configured databases, you can skip the configuration of that database. Although, due to other EnTAP database downloading (taxonomy and ontology), configuration must still be ran at least once without any flags.

Configuration can be ran without formatting a database as follows:

```
EnTAP --config
```

In both cases, the following databases will be downloaded:

- NCBI Taxonomic Database (indexed for EnTAP)
- Gene Ontology Database (indexed for EnTAP)
- EggNOG DIAMOND Database
- EggNOG SQL Database

Note: This is the only stage that requires connection to the Internet.

If you experience any trouble in downloading the databases indexed for EnTAP (taxonomy and gene ontology), you can use the databases contained in the repo download, databases.tar.gz. Just be sure to set the configuration file to these database paths (as these are the binaries)!

EnTAP will always check the databases specified in the configuration file first. Otherwise, EnTAP will check a different location for the databases during Configuration. This location will be either the path specified by - -database-out or, if not selected, the current working directory. The final databases will also be sent there! As a result, the **- -database-out flag is recommended for Configuration.**

3.1.3 Flags:

Required Flags:

- **(- - config)**
  - The only required flag.
  - Although in order to run the full EnTAP pipeline, you must have a .dmnd configured database.

Optional Flags:

- **(-d/ - - database)**
  - Specify any number of FASTA formatted databases you would like to configure for EnTAP
  - Not necessary if you already have DIAMOND configured databases (.dmnd)
- **(- - paths)**
  - Point to entap_config.txt for specifying paths
- **(- - database-out)**
  - Specify an output directory for the databases to be sent to (recommended)
  - This will send the Taxonomic Database, GO Database, and any DIAMOND databases to this location
  - EggNOG database will not be sent here as it must remain in the EggNOG directory
- **(- t/ - - threads)**
  - Specify thread number for Configuration

3.2 Test Data

Before continuing on to the *run* stage, it is advised to do a test run of EnTAP to ensure that everything is properly configured. There should be no errors in the test run. The test data resides within the /test_data directory of the main EnTAP directory. This will walk you through configuring a database for DIAMOND (if you haven't already done so) and executing EnTAP with and without frame selection.

Before we begin, make sure that the paths in the configuration file are correct. Since we are running the configuration stage, EnTAP will check to make sure you have the other databases downloaded (which should have been done prior to this). To begin the test, execute the following command to configure the test DIAMOND database:

```
EnTAP --config -d /test_data/swiss_prot_test.fasta --database-out /test_data
```

This should finish very shortly without any errors and you should find a uniprot_sprot_test.dmnd file within the /test_data directory.

Next up is verifying the main execution stage! Once again, first ensure that the configuration file has all of the correct paths. We are going to check an execution with and without frame selection. If you are not going to use frame selection, you may skip this test!

Note: The following tests will take longer as they will be testing the entire pipeline and running against the larger EggNOG database.

To test EnTAP with frame selection, execute the following command:

```
EnTAP --runP -i /test_data/trinity.fnn -d /test_data/uniprot_sprot_test.dmnd
```

To test EnTAP without frame selection, execute the following command:

```
EnTAP --runP -i /test_data/trinity.faa -d /test_data/uniprot_sprot_test.dmnd
```

These should run without error and you should have several files within the created /outfiles directory. The final_annotations_lvl0.tsv file should resemble the test_data/final_annotations_test.tsv file.

If any failures were seen during the above executions, be sure to go through each stage of installation and configuration to be sure everything was configured correctly before continuing!

3.3 Run

The run stage of *EnTAP* is the main annotation pipeline. After configuration is ran at least once, this can be ran continually without requiring configuration to be ran again (unless more databases will be configured).

The following stages will be ran:

1. *Expression Filtering* (optional)
2. *Frame Selection* (optional)
3. Similarity Search
4. Orthologous Group Assignment
5. InterProScan (optional)

3.3.1 Input Files:

Required:

- .FASTA formatted transcriptome file (either protein or nucleotide)
- .dmnd (DIAMOND) indexed databases, which can be formatted in the :ref:‘configuration<config-label>‘stage.

Optional:

- **.BAM/.SAM alignment file. If left unspecified expression filtering will not be performed.**
  - This can be generated by software that does not perform gapped alignments such as Bowtie (not Bowtie2). All you need to generate an alignment file is a pair of reads and your assembled transcriptome!

3.3.2 Sample Run:

A specific run flag (**runP/runN**) must be used:

- runP: Indicates blastp. Frame selection will be ran if nucleotide sequences are inputted
- runN: Indicates blastx. Frame selection will not be ran with this input

An example run with a nucleotide transcriptome:

```
EnTAP --runN -i path/to/transcriptome.fasta -d path/to/database.dmnd -d path/to/
↪database2.dmnd -a path/to/alignment.sam
```

With the above command, the entire EnTAP pipeline will run. Both frame selection and expression filtering can be skipped if preferred by the user. EnTAP would require protein sequences (indicated by –runP) in order to avoid frame selection. If there is not a short read alignment file provided in SAM/BAM format, then expression filtering via RSEM will be skipped.

3.3.3 Flags:

Required Flags:

- **(- - runP/- - runN)**
  - Specify a blastp or blastx annotation
  - If - -runP is selected with a nucleotide input, frame selection will be ran and annotation stages will be executed with protein sequences (blastp)
  - If - -runP is selected with a protein input, frame selection will not be ran and annotation will be executed with protein sequences (blastp)
  - If - -runN is selected with nucleotide input, frame selection will not be ran and annotation will be executed with nucleotide sequences (blastx)
- **(-i/- - input)**
  - Path to the transcriptome file (either nucleotide or protein)
- **(-d/- - database)**
  - Specify up to 5 DIAMOND indexed (.dmnd) databases to run similarity search against

Optional Flags:

- **(-a/- -align)**

  - Path to alignment file (either SAM or BAM format)
  - **Note:** Ignoring this flag will skip expression filtering
  - If you have ran alignment with single end reads be sure to use the - -single-end flag as well (paired-end is default)
  - Be sure to specify an FPKM threshold

- **(- - contam)**
  - Specify *contaminant* level of filtering
  - Multiple contaminants can be selected through repeated flags
- **(- - taxon)**
  - This flag will allow for *taxonomic* 'favoring' of hits that are closer to your target species or lineage. Any lineage can be used as referenced by the NCBI Taxonomic database, such as genus, phylum, or species.
  - Format **must** replace all spaces with underscores ('_') as follows: "- -taxon homo_sapiens" or "- -taxon primates"
- **(- - level)**
  - Specify Gene Ontology levels you would like to normalize to
  - Any amount of these flags can be used
  - Default: 0 (every level), 3, 4
  - More information at: http://geneontology.org/page/ontology-structure
- **(- - out-dir)**
  - Specify output folder labelling.
  - Default: /outfiles
- **(- - fpkm)**
  - Specify FPKM cutoff for expression filtering
  - Default: 0.5
- **(-e)**
  - Specify minimum E-value cutoff for similarity searching
  - Default: 10E-5
- **(- - tcoverage)**
  - Specify minimum target coverage for similarity searching
  - Default: 50%
- **(- - qcoverage)**
  - Specify minimum query coverage for similarity searching
  - Default: 50%
- **(- - overwrite)**
  - All previously ran files will be overwritten if the same - -tag flag is used
  - Without this flag EnTAP will *recognize* previous runs and skip things that were already ran

- **(- - single-end)**
  - Signify your reads are single end for RSEM execution
  - Default: paired-end
- **(- - graph)**
  - This will check whether or not your system has graphing functionality supported
  - If Python with the Matplotlib module are installed on your system graphing should be enabled!
  - This can be specified on its own
- **(-t/ - - threads)**
  - Specify the number of threads of execution
- **(- - trim)**
  - This flag will trim your sequence headers to anything before a space. It will make your data easier to read if you have a lot of excess information you do not need in your headers.
  - Example:
    - >TRINITY_231.1 protein12312_43_inform
    - >TRINITY_231.1
- **(- - state)**
  - Precise control over execution *stages*. This flag allows for certain parts to be ran while skipping others.
  - Warning: This may cause issues depending on what you plan on running!
- **(- - ontology)**
  - Specify which ontology packages you would like to use
    - 0 - EggNOG (default)
    - 1 - InterProScan
  - Both or either can be specified with multiple flags
    - Ex: - - ontology 0 - - ontology 1
    - This will run both EggNOG and InterProScan
- **(- - protein)**
  - Use this option if you would like to run InterProScan
  - Specify databases to run against (you must have them already installed)
    - tigrfam
    - sfld
    - prodom
    - hamap
    - pfam
    - smart
    - cdd
    - prositeprofiles

    - prositepatterns
    - superfamily
    - prints
    - panther
    - gene3d
    - pirsf
    - coils
    - mobidblite

- **(- - version)**
  - Prints the current EnTAP version you are running
- **(- - uninformative)**
  - Path to a list of terms you would like to be deemed "uninformative"
  - The file **must** be formatted with one term on each line of the file
  - Example (defaults):
    - conserved
    - predicted
    - unnamed
    - hypothetical
    - putative
    - unidentified
    - uncharacterized
    - unknown
    - uncultured
    - uninformative
- **(- - no-check)**
  - EnTAP checks execution paths and inputs prior to annotating to prevent finding out your input was wrong until midway through a run. Using this flag will eliminate the check (not advised to use!)

3.3.4 Expression Analysis

The goal of expression filtering, or transcript quantification, is to determine the relative abundance levels of transcripts when taking into account the sequenced reads and how they map back to the assembled transcriptome and using this information to filter out suspect expression profiles possibly originated from poor or incomplete assemblies. Filtering is done through the use of the FPKM (fragments per kilobase per of million mapped reads) , or a measurable number of expression. This can be specified with the - -fpkm flag as specified above. EnTAP will use this FPKM value and remove any sequences that are below the threshold.

3.3.5 Frame Selection

Frame selection is the process of determining the coding region of a transcript. Oftentimes, due to assembly errors or other factors, a coding region may not be found for a transcript and EnTAP will remove this sequence. When a coding region is found, EnTAP will include the sequence for further annotation.

3.3.6 Taxonomic Favoring and Contaminant Filtering

Taxonomic contaminant filtering (as well as taxonomic favoring) is based upon the NCBI Taxonomy database. In saying this, all species/genus/lineage names must be contained within this database in order for it to be recognized by EnTAP.

Contaminant Filtering:

Contaminants can be introduced during collection or processing of a sample. A contaminant is essentially a species that is not of the target species you are collecting. Some common contaminants are bacteria and fungi that can sometimes be found within collected samples. If a query sequence from your transcriptome is found when matching against a similarity search database, it will be flagged as such (but NOT removed automatically). Oftentimes, researchers would like to remove these sequences from the dataset.

An example of flagging bacteria and fungi as contaminants can be seen below:

```
EnTAP --runN -i path/to/transcriptome.fasta -d path/to/database.dmnd -c fungi -c␣
↪bacteria
```

Taxonomic Favoring

During best hit selection of similarity searched results, taxonomic consideration can utilized. If a certain lineage (such as sapiens) is specified, hits closer in taxonomic lineage to this selection will be chosen. Any lineage such as species/kingdom/phylum can be utilized as long as it is contained within the Taxonomic Database. If it is not located within the database, EnTAP will stop the execution immediately and let you know!

This feature can be utilized with the taxon flag. An example command utilizing both common contaminants and a species taxon can be seen below:

```
EnTAP --runN -i path/to/transcriptome.fasta -d path/to/database.dmnd -c fungi -c␣
↪bacteria --taxon sapiens
```

3.3.7 Picking Up Where You Left Off

In order to save time and make it easier to do different analyses of data, EnTAP allows for picking up where you left off if certain stages were already ran and you'd like analyze data with different contaminant flags or taxonomic favoring. As an example, if similarity searching was ran previously you can skip aligning against the database and analyze the data to save time. However, the - - overwrite flag will not allow for this as it will remove previous runs and not recognize them.

In order to pick up and skip re-running certain stages again, the files that were ran previously **must** be in the same directories and have the same names. With an input transcriptome name of 'transcriptome' and example database of 'complete.protein':

- **Expression Filtering**
  - transcriptome.genes.results
- **Frame Selection**
  - transcriptome.fasta.faa

  - transcriptome.fasta.fnn
  - transcriptome.fasta.lst
- **Similarity Search**
  - blastp_transcriptome_complete.protein.faa.out
- **Gene Family**
  - annotation_results.emapper.annotations
  - annotation_results_no_hits.emapper.annotations

Since file naming is based on your input as well, the flags below **must** remain the same:

- (- - runN / - - runP)
- (- - ontology)
- (- - protein)
- (-i / - - input)
- (-a / - - align)
- **(-d / - - database)**
  - Does not necessarily need to remain the same. If additional databases are added, EnTAP will recognize the new ones and run similarity searching on them whilst skipping those that have already been ran
- (- - qcoverage)
- (- - tcoverage)
- (- - trim)
- (- - out-dir)

3.3.8 State Control

> **Warning:** This is experimental and certain configurations may not work. This is not needed if you'd like to run certain portions because of "picking up where you left off!"

State control of EnTAP allows you to further customize your runs. This is separate from the exclusion of - - align flag to skip expression filtering, or runP, instead of runN, to skip frame selection. You probably will never actually have to use this feature! Nonetheless, state control is based around the following stages of EnTAP:

1. Expression Filtering
2. Frame Selection
3. Transcriptome Filtering (selection of final transcriptome)
4. Similarity Search
5. Gene Ontology / Gene Families

With this functionality of EnTAP, you can execute whatever states you would like with certain commands. Using a '+' will execute from that state to the end, while using a 'x' will stop at that state. These basic commands can be combined to execute whatever you would like. It's easier if I lay out some examples:

- **(- - state 1+)**

  - This will start at expression filtering and continue to the end of the pipeline

- **(- - state 1+4x)**
  - This will start at expression filtering and stop after similarity search
- **(- - state 4x)**
  - This will just execute similarity search and stop
- **(- - state 1+3x5)**
  - This will essentially execute every stage besides similarity searching

The default 'state' of EnTAP is merely '+'. This executes every stage of the pipeline (or attempts to if the correct commands are in place).

CHAPTER 4

Interpreting the Results

EnTAP provides many output files at each stage of execution to better see how the data is being managed throughout the pipeline:

1. *Final Annotation Results*
2. *Log File / Statistics*
3. *Transcriptomes*
4. *Expression Filtering*
5. *Frame Selection*
6. *Similarity Searching*
7. *Orthologous Groups/Ontology*
8. *Protein Families* (optional)

The two files to check out first are the *final annotations* and *log file*. These files are the most important and contain a summary of all the information collected at each stage, including statistical analyses. The remaining files are there for a more in depth look at each stage.

4.1 Final Annotations

The final EnTAP annotations are contained within the */final_results* directory. These files are the summation of each stage of the pipeline and contain the combined information. So these can be considered the most important files!

All .tsv files in this section will have the following header information (from left to right)

- Query sequence ID
- Subject sequence ID
- Percentage of identical matches
- Alignment length

- Number of mismatches
- Number of gap openings
- Start of alignment in query
- End of alignment in query
- Start of alignment in subject
- End of alignment in subject
- Expect (e) value
- Query coverage
- Subject title
- Species (DIAMOND)
- Origin Database (DIAMOND)
- ORF (GeneMarkS-T)
- Contaminant (yes/no the hit was flagged as a contaminant)
- Seed ortholog (EggNOG)
- Seed E-Value (EggNOG)
- Seed Score (EggNOG)
- Predicted Gene (EggNOG)
- Taxonomic Scope (EggNOG, tax scope that was matched)
- OGs (EggNOG, orthologous groups assigned)
- Description (EggNOG)
- KEGG Terms (EggNOG)
- Protein Domains (EggNOG)
- GO Biological (Gene Ontology normalized terms)
- GO Cellular (Gene Ontology normalized terms)
- GO Molecular (Gene Ontology normalized terms)
- —– Optional Columns If Using InterProScan —–
- IPScan GO Biological (InterPro)
- IPScan GO Cellular (InterPro)
- IPScan GO Molecular (InterPro)
- Pathways (InterPro)
- InterPro (InterPro, database entry)
- Protein Database (InterPro, database assigned. Ex: pfam)
- Protein Description (InterPro, description of database entry)
- E Value (InterPro, E-value of hit against protein database)

Gene ontology terms are normalized to levels based on the input flag from the user (or the default of 0,3,4). A level of 0 within the filename indicates that ALL GO terms will be printed to the annotation file. Normalization of GO terms to levels is generally done before enrichment analysis and is based upon the hierarchical setup of the Gene Ontology database. More information can be found at GO.

- final_annotations_lvlX.tsv
  - As mentioned above, the 'X' represents the normalized GO terms for the annotation
  - This .tsv file will have the headers as mentioned previously as a summary of the entire pipeline
- final_annotated.faa / .fnn
  - Nucleotide and protein fasta files containing all sequences that either hit databases through similarity searching or through the ontology stage
- final_unannotated.aa / .fnn
  - Nucleotide and protein fasta files containing all sequences that did not hit either through similarity searching nor through the ontology stage

4.2 Log File / Statistics

The log file contains a statistical analysis of each stage of the pipeline that you ran. I'll give a brief outline of some of the stats performed:

1. Initial Statistics
   - Transcriptome statistics: n50, n90, average gene length, longest/shortest gene
   - Summary of user flags
   - Summary of execution paths (from config file)
2. Expression analysis
   - Transcriptome statistics: n50, n90, average gene length, longest/shortest gene
   - Summary of sequences kept/removed after filtering
3. Frame Selection
   - Transcriptome statistics: n50, n90, average gene length, longest/shortest gene
   - Summary of frame selection: Partial, internal, complete genes. Genes where no frame was found
4. Similarity Searching
   - Contaminant/uninformative/informative count
   - Phylogenetic/contaminant distribution of alignments
   - Alignment distribution based upon frame results (partial/internal/complete)
   - Sequence count that did not align against a database reference
   - Statistics calculated for each individual database and final results
5. Gene Family Assignment
   - Phylogenetic distribution of gene family assignments
   - Gene Ontology level distribution (note: level 0 means all levels)
   - Gene Ontology category distribution (biological processes, molecular function, cellular component)

6. InterProScan
    - Additional statistics coming soon!
7. Final Annotation Statistics
    - Statistical summary of each stage
    - Runtime

4.3 Transcriptomes

The */transcriptomes* contains the original, processed, and final transcriptomes being used by EnTAP. The files are as follows with the 'transcriptome' tag based upon the name of your input transcriptome:

- transcriptome.fasta
    - This file is essentially a copy of your input transcriptome. The sequence ID's may be changed depending on whether you selected the 'trim' flag or otherwise.
- transcriptome_expression_filtered.fasta
    - As the name implies, this transcriptome is the resultant of the Expression Filtering stage with sequences removed that fall under the FPKM threshold you have specified.
- transcriptome_frame_selected.fasta
    - This transcriptome is the resultant of Frame Selection. Sequences in which a frame was not selected are removed and those with a frame are kept in this file. As a result, this file will always be in protein format.
- transcriptome_final.fasta
    - This is your final transcriptome following the "Transcriptome Filtering" stage of EnTAP. **This transcriptome will be used for the later stages of the pipeline** (Similarity Searching and Ontology). Depending on which methods of execution you chose (runN / runP), the result here may be either protein or nucleotide with Frame Selection and/or Expression Filtering.

4.4 Expression Filtering (RSEM)

The */expression* folder will contain all of the relevant information for this stage of the pipeline. This folder will contain the main files (results from expression analysis software), files processed from EnTAP (including graphs).

4.4.1 RSEM Files: */expression*

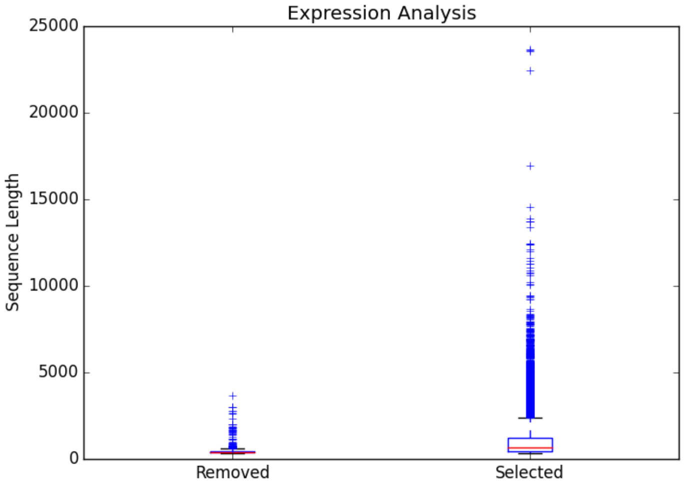
The */expression* directory will contain all of the output from RSEM including a converted BAM file (if you input a SAM) and the results of the expression analysis.

4.4.2 EnTAP Files: */processed*

This directory will contain all of the files produced from EnTAP concerning expression analysis. With a generic transcriptome input of "Species.fasta", these files will have the following format

- Species_removed.fasta
    - Fasta file of sequences that were under the specified FPKM threshold

- Species_kept.fasta
  - Fasta file of sequences that were kept after filtering (over the FPKM threshold)
- */figures*
  - Directory containing a box plot of sequence length vs the sequences that were removed and kept after expression analysis



4.5 Frame Selection (GeneMarkS-T)

The */frame_selection* folder will contain all of the relevant information for the frame selection stage of the pipeline. This folder will contain the *main files* (results from frame selection software), files *processed* from EnTAP, and *figures* generated from EnTAP.

4.5.1 GeneMarkS-T Files: */frame_selection*

The files within the root */frame_selection* directory contain the results from the frame selection portion of the pipeline. More information can be found at GeneMarkS-T. With a generic transcriptome input of "Species.fasta", these files will have the following format:

- Species.fasta.fnn
  - Nucleotide fasta formatted frame selected sequences
- Species.fasta.faa
  - Amino acid fasta formatted frame selected sequences
- Species.fasta.lst
  - Information on each sequence (partial/internal/complete/ORF length)
- .err and .out file
  - These files are will contain any error or general information produced from the GeneMarkS-T run

4.5.2 EnTAP Files: */processed*

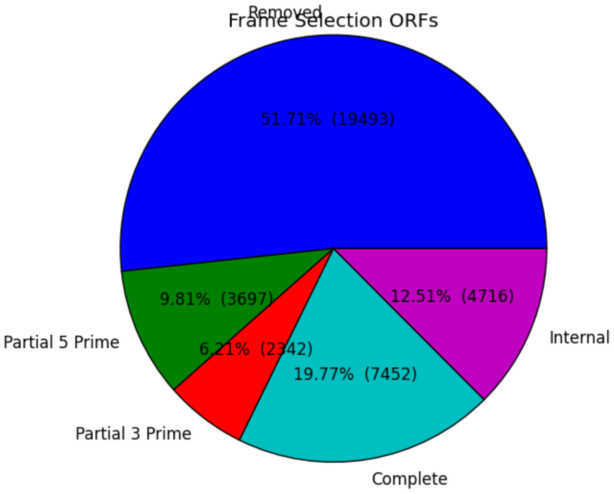
Files within the */processed* are generated by EnTAP and will contain ORF information based on the GeneMarkS-T execution.

- complete_genes.fasta
  - Amino acid sequences of complete genes from transcriptome
- partial_genes.fasta
  - Amino acid sequences of partial (5' and 3') sequences
- internal_genes.fasta
  - Amino acid sequences of internal sequences
- sequences_lost.fasta
  - Nucleotide sequences in which a frame was not found. These will not continue to the next stages of the pipeline

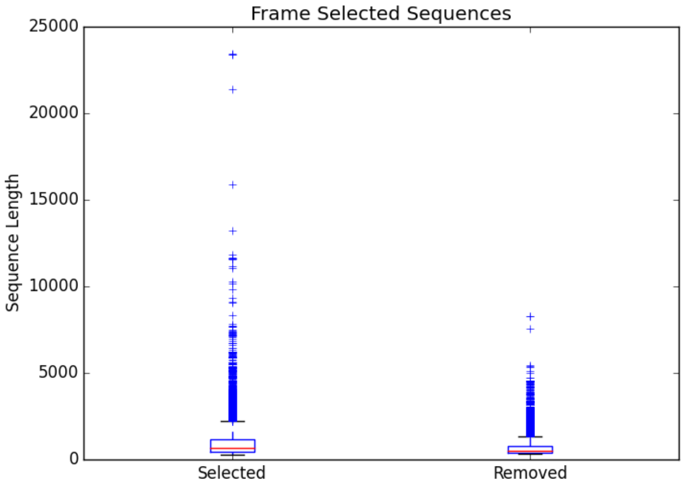
4.5.3 EnTAP Files: */figures*

In addition to files, EnTAP will generate figures within the */figures* directory. These are some useful visualizations of the information provided by GeneMarkS-T

- frame_results_pie.png
  - Pie chart representing the transcriptome (post expression filtering) showing complete/internal/partial/and sequences in which a frame was not found



- frame_selected_seq.png
  - Box plot of sequence length vs. the sequences that were lost during frame selection and the sequences in which a frame was found



4.6 Similarity Search (DIAMOND)

The */similarity_search* directory will contain all of the relevant information for the similarity searching stage of the pipeline. This folder will contain the *main files* (results from similarity search software), *files* analyzing hits from each database, *overall* results combining the information from each database, and *figures* generated from EnTAP.

4.6.1 DIAMOND Files: */similarity_search*

The files within the */similarity_search* directory contain the results from the similarity searching portion of the pipeline against each database you select. More information can be found at DIAMOND. With running _blastp (protein similarity searching), a generic transcriptome input of “Species.fasta”, with a database called “database” the files will have the following format:

- blastp_Species_database.out
  - This contains the similarity search information provided in the format from DIAMOND
  - Header information (from left to right):
    - Query Sequence ID
    - Subject Sequence ID
    - Percentage of Identical Matches
    - Alignment Length
    - Number of Mismatches
    - Number of gap openings
    - Start of alignment in query
    - End of alignment in query
    - Start of alignment in subject
    - End of alignment in subject

    - Expect (e) value
    - Bit score
    - Query Coverage
    - Subject Title (pulled from database)

- blastp_Species_database_std.err and .out
  - These files are will contain any error or general information produced from DIAMOND

4.6.2 EnTAP Files: */processed*

Files within the */processed* are generated by EnTAP and will contain information based on the hits returned from similarity searching against each database. This information contains the *best hits* (discussed previously) from each database based on e-value, coverage, informativeness, phylogenetic closeness, and contaminant status.

The files below represent a run with the same parameters as the section above:

- All the TSV files mentioned in this section will have the same header as follows (from left to right):
  - Query sequence ID
  - Subject sequence ID
  - Percentage of identical matches
  - Alignment length
  - Number of mismatches
  - Number of gap openings
  - Start of alignment in query
  - End of alignment in query
  - Start of alignment in subject
  - End of alignment in subject
  - Expect (e) value
  - Query coverage
  - Subject title
  - Species (pulled from hit)
  - Origin Database
  - ORF (taken from frame selection stage)
  - Contaminant (yes/no the hit was flagged as a contaminant)
- database/best_hits.faa and .fnn and .tsv
  - Best hits (protein and nucleotide) that were selected from this database
  - This contains ALL best hits, including any contaminants that were found as well as uninformative hits
  - The .tsv file contains the header information mentioned above of these same sequences
  - Note: Protein or nucleotide information may not be available to report depending on your type of run (these files will be empty)
- database/best_hits_contam.faa/.fnn/.tsv

  - Contaminants (protein/nucleotide) separated from the best hits file. As such, these contaminants will also be in the _best_hits.faa/.fnn.tsv files

- database/best_hits_no_contam.faa/.fnn/.tsv
  - Sequences (protein/nucleotide) that were selected as best hits and not flagged as contaminants
  - With this in mind: best_hits = best_hits_no_contam + best_hits_contam
  - These sequences are separated from the rest for convenience if you would like to examine them differently
- database/no_hits.faa/.fnn/.tsv
  - Sequences (protein/nucleotide) from the transcriptome that did not hit against this particular database.
  - This does not include sequences that were lost during expression filtering or frame selection
- database/unselected.tsv
  - Similarity searching can result in several hits for each query sequence. With only one best hit being selected, the rest are unselected and end up here
  - Unselected hits can be due to a low e-value, coverage, or other properties EnTAP takes into account when selecting hits

4.6.3 EnTAP Files: */overall_results*

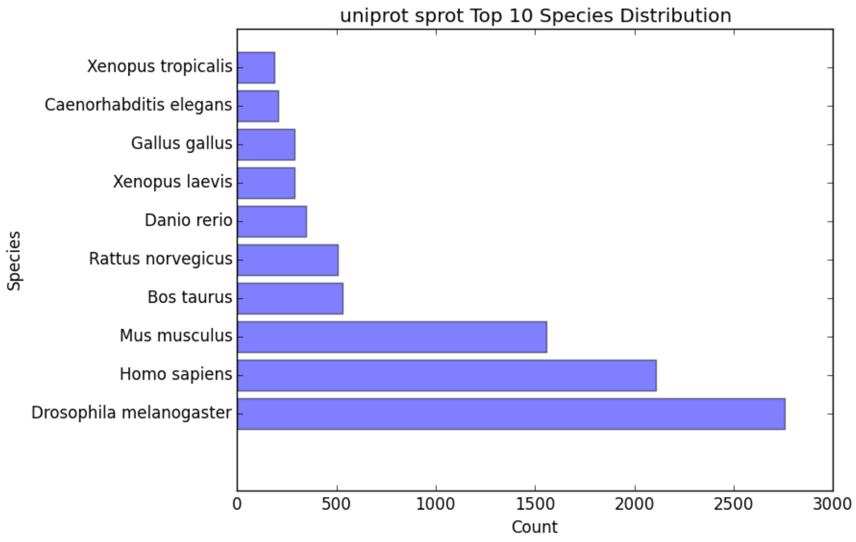
While the */processed* directory contains the best hit information from each database, the */overall_results* directory contains the overall best hits combining the hits from each database.

4.6.4 EnTAP Files: */figures*

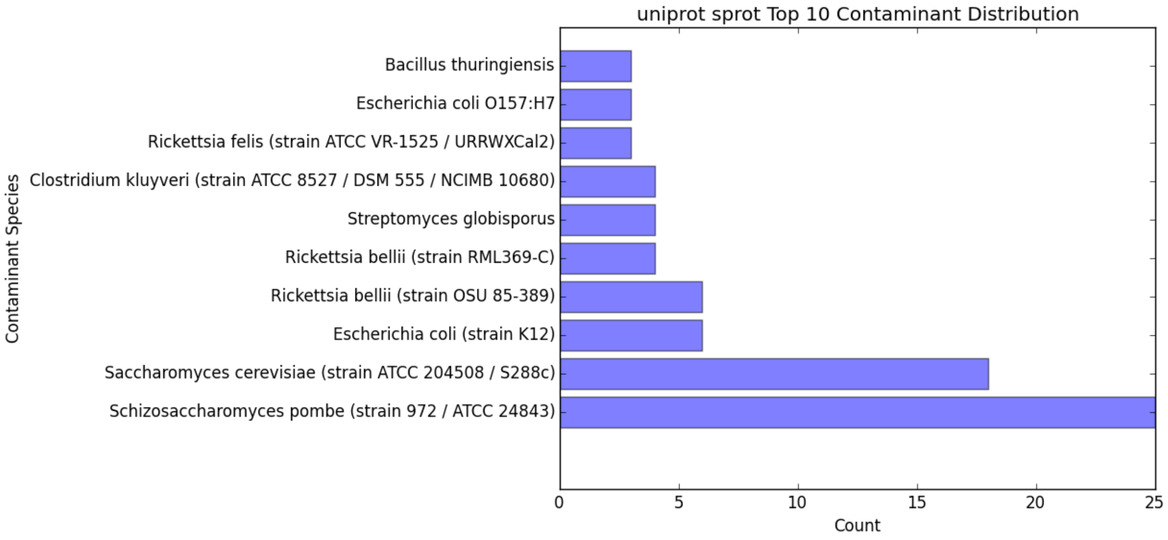
In addition to files, EnTAP will generate figures within the */figures* directory for each database. These are some useful visualizations of the information provided by similarity searching.

Here, there will be several figures:

- species_bar.png / species_bar.txt
  - Bar graph representing the top 10 species that were hit within a database
  - Text file representing the data being displayed



- contam_bar.png / contam_bar.txt
  - Bar graph representing the top 10 contaminants (within best hits) that were hit against the databast
  - Text file representing the data being displayed



4.7 Orthologous Groups/Ontology (EggNOG)

The */ontology/EggNOG* directory will contain all of the relevant information for the EggNOG stage of the pipeline. This folder will contain the *EggNOG files*, *files* analyzing the annotation from EggNOG, and *figures* generated from EnTAP.

4.7.1 EggNOG Files: */ontology/EggNOG*

Files within the */ontology/EggNOG* are generated by EggNOG and will contain information based on the hits returned from EggNOG against the orthologous databases. More information can be found at EggNOG.

- annotation_results.emapper.annotations
  - EggNOG results for sequences from the final transcriptome being used (post-processing)

4.7.2 EnTAP Files: */processed*

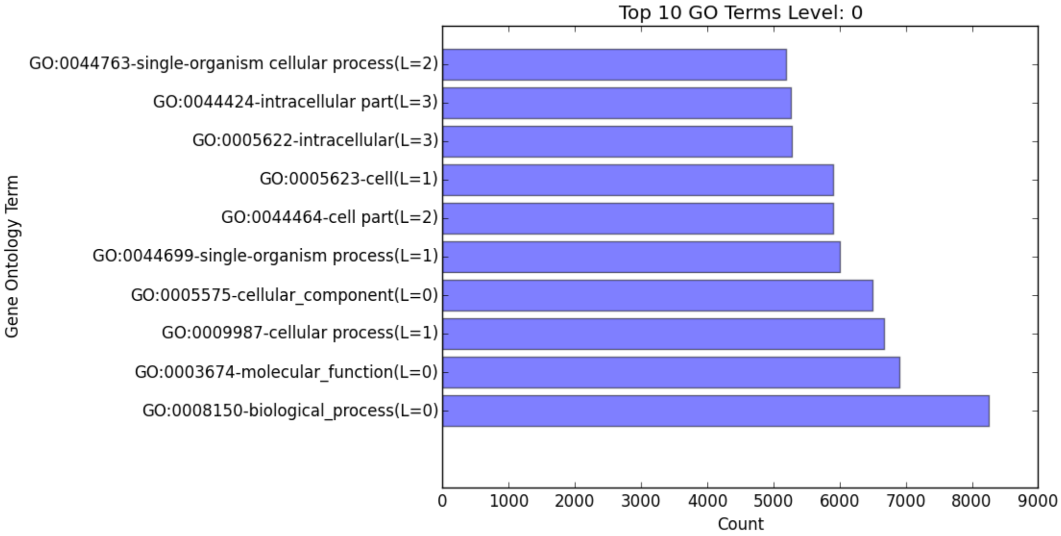
Files within the */processed* are generated by EnTAP and contain information on what sequences were annotated and which were not.

- unannotated_sequences.fnn/faa
  - Sequences where no gene family could be assigned (nucleotide/protein)
- annotated_sequences.fnn/faa
  - Sequences where a gene family could be assigned (nucleotide/protein)

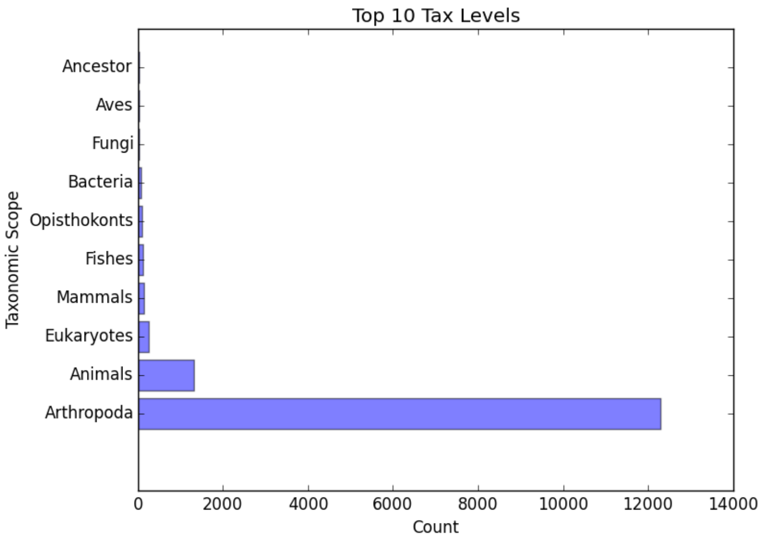
4.7.3 EnTAP Files: */figures*

The */figures* will contain figures generated by EnTAP of Gene Ontology and Taxonomic distribution of the results

- (overall/molecular_function/cellular_component/biological_process)#_go_bar_graph.png/.txt
  - Bar graph of each category of Gene Ontology terms of a specific level # (remember, level 0 signifies all levels!)



- eggnog_tax_scope.png/.txt
  - A bar graph representation of the taxonomic scope of the gene families assigned through EggNOG



4.8 Protein Families (InterProScan)

Full documentation coming soon!

CHAPTER 5

Changelog

This page contains (mostly) all of the changes that were made between each version of EnTAP. The current latest version is EnTAP Beta v0.8.1-beta

5.1 EnTAP Beta v0.8.1-beta

- Added additional error logging to provide more information when something goes wrong
- Configuration file mandatory (default place to look is current working directory)
- Changed tax database paths in config file to avoid confusion (separate text and bin). Config file must be re-downloaded/generated!
- Defaults/output during configuration changed to config file then if not found, database-out flag
- Added deletion of empty files if a certain stage failed (preventing re-reading an empty file)
- Added errors/warnings for no alignments/hits in each stage
- entap_out directory changed to transcriptomes to be more clear (holds only transcriptomic data)
- Final EnTAP output files moved from the root outfiles directory to final_results directory
- Several filename changes to add consistency in new transcriptomes directory (final transcriptome is now _final.fasta.
- Several title changes to the log file to mitigate confusion
- EggNOG no longer broken down into separate files - those that hit and those that did not hit a database. Now entire transcriptome is pushed with one output file
- 10 species/contaminants/other in similarity searching statistics has been changed to 20 to provide more information to the user
- Best hit selection state combined with similarity search
- Added ‘N’ as an accepted nucleotide

- Several behind the scenes changes
- Fixed Cmake global installation issue
- Fixed incorrect error codes
- Fixed InterPro printing bug to no hits/hits files
- Fixed Frame Selection not printing new lines for certain files

5.2 EnTAP Beta v0.8.0-beta

- Overhaul of the taxonomic/gene ontology databases
  - Faster accession/indexing
  - MUST be re-downloaded and re-indexed (or use the updated versions that come with the EnTAP distribution)
  - Taxonomic database includes thousands more entries with synonyms of many species
  - Perl is no longer a dependency, with Python being used to download the database
- Added blastx support
  - Blastx now allowed for ALL stages of annotation (similarity search + ontology)
  - –runN flag now specifies blastx (frame selection will not be ran)
  - –runP flag now specifies blastp (frame selection will be performed if nucleotide sequences are input)
- Added InterProScan support
  - Now possible to run EggNOG and/or InterProScan (with both blastx or blastp)
  - EggNOG and/or InterProScan specified with –ontology flag (0 and/or 1)
  - Full output of both will be provided in the final annotations file
- Added additional statistics to the log file for EggNOG and Expression Analysis
- Added numerous file/path/software checks to the start of an EnTAP run
  - Test runs/path checks are performed on all software that will be ran
  - Additional checks to specific flags
  - These checks can be turned off for an EnTAP run with –no-check flag (not advised!)
- –tag flag changed to –out-dir to specify output directory (not just what you'd like it named as)
  - Defaults to current directory with /outfiles folder
- –paired-end flag for Expression Filtering changed to –single-end (with paired-end being the default)
- Added contaminant and informative yes/no columns in final annotations file (among other headers)
- Added ability to input your own list of informative/uninformative terms for EnTAP to flag
- Added contaminant and none contaminant final annotation files
- Fixed a sequence id issue in Expression Filtering not mapping to BAM/SAM file
- Fixed a bug in –trim flag for sequence headers
- Fixed a bug where some systems had issues with graphing
- Debug and log files are now time stamped and not overwritten

- Fixed pathing for EnTAP configuration and made more streamlined
- Fixed several instances of older compilers complaining
- Added a lot of error messaging to help diagnose any issues easily
- Changed similarity search to have full database name, not path
- Fixed a bug in parsing input fasta file (added corrupt file checks)

5.3 EnTAP Beta v0.7.4.1-beta

- Minor changes to taxonomic database download and indexing

5.4 EnTAP Beta v0.7.4-beta

- Initial beta release!

CHAPTER 6

Troubleshooting

This page contains several useful ways to troubleshoot whenever you run into an issue with EnTAP execution or installation. If you do not come to a solution through checking inputs, paths, or the configuration file I'd review these in the order I am laying them out.

1. *Error Codes* : Provide specific information on where your run failed within the EnTAP pipeline. This could be during execution or installation. These can be as general as 'DIAMOND has failed,' or much more detailed. They will be more detailed if the failure was detected by EnTAP rather than accompanying software.
2. *.err and .out Files* : Provide information from the software that has failed. These are essentially what the software would print out to the user if an error occurs. Thus, they are out of EnTAP's control and are the most useful item when troubleshooting why a specific program failed. This should be the second thing you look for after checking the standard error code produced by EnTAP. An example would look like: 'DIAMOND was unable to finish execution due to a memory overflow issue.'
3. *FAQs* : If you're still running into trouble, check out these common issues!

6.1 Error Codes

EnTAP has quite a few error codes that are generally going to have messages attached to them when your execution fails. I'll list the error codes and possible messages/troubleshooting ideas here. This is a continually evolving section where additional codes will be added as they are created (and I find the time to update). However, pre-existing codes will NOT be changed!

0. No error!

10. There is some error in your inputs to EnTAP. This is most likely caused by your command arguments. Make sure to check these out as well as any relevant paths. EnTAP should provide the specific error attached to this code as to why the run failed.

12. Some error in parsing the configuration file attached to EnTAP. Make sure it is following the proper format as when it was first downloaded from the GitLab page. There should be no spaces or extra line breaks after the '=' for each parameter.

13. This error is called when the configuration file could not be generated. If EnTAP does not recognize a configuration file, it will attempt to generate one for you.
14. This error is called when a configuration file could not be found in either your input, or the execution directory for EnTAP. It will merely generate the file and exit execution to allow you to fill in any parameters you may want to include.
15. There was an issue in download the EnTAP Taxonomic Database. This will only be called when you are running the - -config command. Some solutions can be: check your internet connection, ensure you have the required Python libraries installed (it currently uses Python as a means of download), and try running this outside of EnTAP (with the commmand python tax_download.py -o output_file.txt). If you decide to download the text version separately, just be sure to include that in the entap_config.txt file and it will convert it to a binary when you run - - config again.

6.2 .err and .out Files

6.3 FAQs