AMOEBAE documentation

Lael D. Barlow

Version of May 25, 2020

Contents

1 Introduction

1.1 What is AMOEBAE?

Analysis of MOlecular Evolution with BAtch Entry (AMOEBAE) is a bioinformatics software toolkit composed primarily of scripts written in the Python3 language. AMOEBAE scripts use existing Python packages including Biopython (Cock *et al.*, 2009), the Environment for Tree Exploration (ETE3) (Huerta-Cepas *et al.*, 2016), pandas, and Matplotlib (Hunter, 2007) for setting up, running, and summarizing analyses of molecular evolution using bioinformatics software packages including MUSCLE (Edgar, 2004), BLAST+ (Camacho *et al.*, 2009), HMMer3 (Eddy, 1998), and IQ-Tree (Nguyen *et al.*, 2015). Applications include identifying and classifying predicted peptide sequences according to their evolutionary relationships with homologues. All dependencies are freely available, and AMOEBAE code is open-source (see subsection 1.9) and available on GitHub (`https://github.com/laelbarlow/amoebae`).

1.2 Why use AMOEBAE?

Webservices such as those provided by NCBI (`https://blast.ncbi.nlm.nih.gov/Blast.cgi`) (Camacho *et al.*, 2009) provide a means to investigate the evolution of one or a few genes via similarity searching, and automated pipelines such as orthoMCL (Li, 2003) attempt to rapidly perform orthology prediction for all genes in several genomes. AMOEBAE addresses mid-scale analyses which are too cumbersome to be done via webservices and yet require a level of detail and flexibility not offered by automated pipelines. AMOEBAE may be useful for analyzing the distribution of orthologues of up to perhaps 30 genes/proteins among a sampling of no more than approximately 100 eukaryotic genomes. However, you may need to carefully define the scope of your analysis depending on what additional steps you may find necessary beyond those that may be performed using AMOEBAE (30 queries and 100 genomes may in fact be unmanageable). AMOEBAE provides many options which can be tailored to the specific genes/proteins being analyzed, and allow analyses using complex sets of customized criteria to be reproduced more practically.

1.3 Key features

The core functionality is to run sequence similarity searches with multiple algorithms, multiple queries, and multiple databases simultaneously and to allow highly customizable implementation of reciprocal-best-hit search strategies. The output includes detailed summaries of results in the form of a spreadsheet and plots.

A particular advantage of AMOEBAE over other tools is the functionality for parsing results of TBLASTN (searching in nucleotide sequences with peptide sequence queries) search results. This allows rapid identification of High-scoring Segment Pair (HSP) clusters at separate gene loci (identified according to user-defined criteria), automatic checking of those loci against information in genome annotation files, and systematic use of Exonerate (Slater and Birney, 2005) where possible for obtaining better exon predictions.

1.4 A word of caution

AMOEBAE is not optimized for ease of use, but is meant to be highly configurable. The many options available to AMOEBAE users inevitably provide many opportunities for errors in specifying search criteria, and errors in interpreting output files. Some prior experience with similarity searching and with running software using the command line is essential for using AMOEBAE, and experience writing scripts in Bash and Python would be highly advantageous. Moreover, AMOEBAE is still under active development, so some features may not yet be thoroughly tested.

1.5 User support

For specific issues with the code, please use the issue tracker on the GitHub webpage here: `https://github.com/laelbarlow/amoebae/issues`.

If you have general questions regarding AMOEBAE, please email the author at lael (at) ualberta.ca.

1.6 Documentation

This document provides an overview of AMOEBAE and describes the functionality of the various commands/scripts. For a tutorial which includes a working example of a similarity search analysis run using AMOEBAE, see the Jupyter Notebook: amoebae/notebooks/similarity_search_tutorial_2.ipynb. For code documentation, please see the html file(s), which can be opened with your web browser: `amoebae/doc/code_documentation/html/index.html`.

1.7 How to cite AMOEBAE

Please cite the GitHub webpage `https://github.com/laelbarlow/amoebae` (or alternative permanent repositories if relevant). Also, the first publication to make use of a version of AMOEBAE was an analysis of Adaptor Protein subunits in embryophytes by Larson *et al.* (2019).

Also, you may wish to cite the software packages which are key dependencies of AMOEBAE, since AMOEBAE would not work without these (see subsection 2.2).

1.8 Acknowledgments

AMOEBAE was initially developed at the Dacks Laboratory at the University of Alberta, and was supported by National Sciences and Engineering Council of Canada (NSERC) Discovery grants RES0021028, RES0043758, and RES0046091 awarded to Joel B. Dacks, as well as an NSERC Postgraduate Scholarship-Doctoral awarded to Lael D. Barlow.

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG).

Also, help with testing AMOEBAE has been kindly provided by Raegan T. Larson, Shweta V. Pipalya, Kira More, and Kristína Záhonová.

1.9 License

2 How to start using AMOEBAE

2.1 System requirements

Please note that the commands shown likely only work on macOS or Linux operating systems (you may have trouble running AMOEBAE directly on Windows).

2.2 Dependencies

All dependencies are free and open-source, and can be automatically installed in a virtual environment (see subsection 2.3).

These are the main depencencies of AMOEBAE:

- Python3 (the Anaconda distribution works well).
- Biopython, a Python package for bioinformatics (Cock *et al.*, 2009).
- The Environment for Tree Exploration 3 (ETE3), a Python package for working with phylogenetic trees (Huerta-Cepas *et al.*, 2016).
- Matplotlib, a Python package for generating plots (Hunter, 2007).
- (gffutils).
- NCBI BLAST+, a software package for sequence similarity searching (Camacho *et al.*, 2009).
- HMMer3, a software package for profile sequence similarity searching (Eddy, 1998).
- MUSCLE, for multiple sequence alignment (Edgar, 2004).
- IQ-TREE, for phylogenetic analysis (Nguyen *et al.*, 2015).

2.3 Setting up an environment for AMOEBAE using Singularity

Follow the steps below to set up AMOEBAE on your personal computer. This setup process should take approximately 20 minutes to complete. Additional instructions for setting up AMOEBAE on a remote server will soon be added as well.

1. Ensure that Git is installed on your computer If you do not already have git installed, then your computer will prompt you with instructions for how to install it when you type git into the command line. If you have a newer version of macOS it may prompt you to install developer tools, which may take up a considerable amount of storage space. Documentation for Git is available here: `https://git-scm.com/doc`. You can check which version you have (or whether it is installed at all) by running the command below. Please note: Here ">>>" is used to indicate that the following text in the line is to be entered in you terminal command prompt.

   ```
   >>> git --version
   ```

2. Clone the AMOEBAE repository using Git. If you simply download the code from GitHub, instead of cloning the repository, then AMOEBAE cannot record specifically what version of the code you use, and will not run properly. Make sure to use the appropriate directory path (the path shown is just an example). Also, replace the path shown below with the path to the directory on your system where you wish to put the main AMOEBAE directory.

   ```
   >>> cd /path/to/directory/where/you/keep/files
   >>> git clone https://github.com/laelbarlow/amoebae.git
   ```

3. Set up AMOEBAE. This performs several steps including checking for whether singularity is installed and attempting to use VirtualBox and Vagrant to run Singularity in a pre-built Ubuntu virtual machine with Singularity installed. This is because Singularity does not run on MacOS (or Windows), and installation of Singularity on Linux is complex, as several dependencies are required. This script downloads a prebuilt singularity container, which was built using the singularity.recipe file, and provided on the Singularity Library (`https://cloud.sylabs.io/library/_container/5e8ca8fff0f8eb90a8a7b60d`).

   ```
   >>> cd amoebae
   >>> bash setup.sh
   ```

4. If you are setting up AMOEBAE on a high performance computing cluster, then you will not be able to install Singularity yourself, and may need to use specific procedures to load Singularity prior to use.

2.4 Running AMOEBAE using Jupyter notebooks

1. After setting up AMOEBAE according to the instructions above, the easiest way to start running analyses using AMOEBAE is via the tutorials, which are in the form of Jupyter notebooks (`https://jupyter.org/`). These Jupyter notebooks can be run using the installation of Jupyter in the Singularity container, and can be accessed using your browser (on a personal computer). To start a Jupyter server, run the bash script as indicated below (assuming your current working directory is the main amoebae directory that you cloned with Git).

   ```
   >>> bash singularity_jupyter.sh
   ```

2. Copy and past the resulting URL (the one at the bottom of the output) into the address bar of your web browser (either Firefox, Chrome, or Safari will work). This will open Jupyter to the notebooks subdirectory, which contains several tutorial and example notebooks (.ipynb files). These files are the files on your regular (host) filesystem, as the amoebae directory is synced with the Singularity container. Thus changes to files will persist after you shut down the Jupyter server and the Singularity container. Documentation on Jupyter is available here: `https://jupyter-notebook.readthedocs.io/en/stable/`.

3. Click on one of the tutorial files (.ipynb). These Jupyter notebooks include information on how to use them once opened. The first tutorial (amoebae_tutorial_1.ipynb) provides a simple example of similarity searching with BLASTP using a Jupyter notebook. The second tutorial (amoebae_tutorial_2.ipynb) provides an example using most of the similarity searching functionality that AMOEBAE provides.

4. To shut down the Jupyter server, click the logout button in the jupyter browser tab(s), and then go to the terminal window that you used to startup the Jupyter server, and press CTRL-C to kill the Jupyter kernel. This will close the Jupyter notebooks, but the

analysis output files will remain, because they are saved to your amoebae/notebooks folder which is on your host machine and accessed from within the container.

5. Working with the Jupyter notebooks interactively in this manner on high-performance computing clusters is likely possible but inconvenient, and procedures will vary. Also, running the tutorial notebooks would require access to the internet from compute nodes (as opposed to login nodes) which may not be supported. Therefore, it is recommended that you run the tutorials on a personal laptop/desktop computer if possible. To run your own notebooks on a cluster, you will need to write a job submission script that will be specific to the cluster, the job scheduler it uses, and your account details. Please refer to documentation provided by your system administrators for this. For an example script that writes a script for running a notebook as a job to a SLURM job scheduler see `https://github.com/laelbarlow/amoebae/blob/master/notebooks/write_notebook_slurm_script.sh`.

2.5 Running AMOEBAE via the command line

1. The easiest way to access AMOEBAE dependencies via the command line is to use the bash script provided. If you are running singularity in a virtual machine (*e.g.*, on MacOS), then only one shell session may be opened at once (and these cannot be opened at the same time as the singularity_jupyter.sh script is running Singularity in a virtual machine). Running the script as indicated below will open a shell session in the Singularity container, with the amoebae directory being the only one accessible. Also, the amoebae executable script is added to the $PATH in the container, so you can run amoebae commands from any directory.

   ```
   >>> bash singularity_shell.sh
   ```

2. You may find it useful to explore and test the environment using the following commands.

   - Print the paths included in the $PATH variable in the container.

     ```
     >>> tr ':' '\n' <<< "$PATH"
     ```

   - Check the location of the amoebae executable being run from within the container.

     ```
     >>> command -v amoebae
     ```

   - Check that the amoebae executable script can be run (print the help message).

     ```
     >>> amoebae -h
     ```

   - Check that all modules can be imported in all python files in the AMOEBAE code.

     ```
     >>> amoebae check_imports
     ```

   - Check that key dependencies such as BLASTP can be accessed (they are installed in the Singularity container).

```
        >>> amoebae check_depend
```

3. Again, running AMOEBAE commands on high-performance computing clusters will require you to write custom job submission scripts. Please refer to documentation provided by your system administrator(s) regarding details specific to your cluster, including the job scheduler used. Also, refer to the Singularity documentation for formulating Singularity commands (`https://sylabs.io/docs/`).

3 Command reference

Documentation for each AMOEBAE command and the various options may be accessed from the command line via the "-h" options. The following command reference information is the output of running amoebae (and each command) with the "-h" option.

3.1 amoebae

```
usage: amoebae <command> [<args>]

Commands for setting up data structure:
    mkdatadir        Make a directory with subdirectories and CSV files for
                     storing sequence data, etc.

Commands for similarity searching:
   setup_hmmdb       Construct an HMM database (with hmmpress).
   add_to_dbs        Format and add a file to a formatted directory.
   list_dbs          Print a list of all usable database files in the database
                     directory as defined in the settings file.
   add_to_queries    Add a query file to a formatted directory.
   list_queries      Print a list of all usable query files in the query
                     directory as defined in the settings file.
   get_redun_hits    Run searches with queries to find redundant hits in
                     databases (for interpreting results).
   setup_fwd_srch    Make directory in which to perform forward searches.
   run_fwd_srch      Perform searches with given queries into given dbs.
   sum_fwd_srch      Append information about forward searches to csv summary
                     file (this is used to organize reverse searches).
   setup_rev_srch    Make a directory in which to perform reverse searches.
   run_rev_srch      Perform searches with given forward search hits into given db.
   sum_rev_srch      Append information about reverse searches to csv summary
                     file.
   interp_srchs      Interpret search results based on summary.
   find_redun_seqs   Identify sequences likely encoded on redundant loci
                     predicted for the same species.
   plot              Plot search results.
```

```

Commands for phylogenetic analysis using a reference tree:
   add_to_models     Add an alignment, tree, substitution model, names of
                     clade-defining sequences to a directory with other models.
   list_models       Print a list of all usable model/reference tree names in
                     the models directory as defined in the settings file.
   get_alt_topos     Take a tree and make copies with every alternative
                     topology for the branches connecting the clades of
                     interest.

Commands for phylogenetic analysis without a reference tree:
   prune             Identify sequences in a tree, and remove them from a
                     given alignment for further phylogenetic analysis.
   auto_prune        Automatically identify sequences in a tree, and remove
                     them from a given alignment for further phylogenetic
                     analysis.
   reduce_tree       Remove terminal nodes from a given tree if there are
                     not any sequences with the same name in a given multiple
                     sequence alignment file.
   constrain_mb      Add constraint commands to MrBayes input file based on a
                     given tree topology.
   visualize_tree    Parse phylogenetic analysis output files for a single
                     alignment in a given directory, and write human-readable
                     tree figures to PDF files.
   replace_seqs      Replace sequences in an alignment with their top hits in a
                     given fasta file (useful if genomes or taxon selection has
                     been updated).

Miscellaneous commands:
   csv_to_fasta      Generate a fasta file from sequences detailed in a
                     spreadsheet of similarity search results.
   check_depend      Check that all the dependencies are properly installed and
                     useable.
   check_imports     Check that all the import statements used in the AMOEBAE
                     repository run without error.

positional arguments:
  command     Specify one of the functionalities of amoebae.

optional arguments:
  -h, --help  show this help message and exit

Copyright 2018 Lael D. Barlow Licensed under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with the
License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law
or agreed to in writing, software distributed under the License is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
```

```
permissions and limitations under the License.
```

3.2 amoebae mkdatadir

```
usage: amoebae [-h] new_dir_path

Make a directory with subdirectories and CSV files for storing sequence data,
etc.

positional arguments:
  new_dir_path  Specify the full file path that you want the new directory to
                have.

optional arguments:
  -h, --help    show this help message and exit
```

3.3 amoebae setup_hmmdb

```
usage: amoebae [-h] indirpath

Construct an HMM database (with hmmpress). This is for later sorting of given
sequences into categories based on which HMM the score highest against.

positional arguments:
  indirpath   Path to directory containing amino acid sequence alignment
              file(s) to be constructed into an HMM database using hmmpress
              from the HMMer3 software package.

optional arguments:
  -h, --help  show this help message and exit
```

3.4 amoebae add_to_dbs

```
usage: amoebae [-h] [--split_char SPLIT_CHAR] [--split_pos SPLIT_POS]
               [--skip_header_reformat] [--auto_extract_accs]
               new_file

Format and add a file to a formatted directory.

positional arguments:
  new_file              Can be a fasta file (prot or nucl) or HMM databases,
                        generated using the hmmpress program in the HMMer
                        software package. Or a GFF3 annotation file.

optional arguments:
  -h, --help            show this help message and exit
  --split_char SPLIT_CHAR
```

```
                        Character to split the header string on for extracting
                        the accession. (default: )
  --split_pos SPLIT_POS
                        Position that the accession will be in after
                        splitting. (default: 0)
  --skip_header_reformat
                        Skip reformatting of header lines in input fasta file.
                        (default: False)
  --auto_extract_accs   Automatically identify accessions/IDs in sequence
                        headers (overrides split_char and split_pos options
                        above). (default: False)
```

3.5 amoebae list_dbs

```
usage: amoebae [-h]

Print a list of all usable query files in the query directory as defined in
the settings file.

optional arguments:
  -h, --help  show this help message and exit
```

3.6 amoebae add_to_queries

```
usage: amoebae [-h] query_file

Add a query file to a formatted directory. This command adds a given sequence
file to the directory with the path that you have specified in the settings.py
file, and appends a corresponding line to the CSV file that you specified
(e.g., '0_query_info.csv') to indicate the query title, etc.

positional arguments:
  query_file  Path to a sequence file in FASTA format that can be used as a
              similarity search query file. Or path to a directory containing
              only files for addition to the queries. Note: By default, the
              portion of the input filename preceding the first underscore
              character will be recorded as the "query title", the remaining
              substring preceding the second underscore character will be
              recorded as the taxon (e.g., "Hsapiens"), and the rest of the
              filename preceding the filename extension will be recorded as
              the sequence ID. So the filename might look like this:
              "QUERYTITLE_HSAPIENS_SEQUENCEID.fa". However, the relevant
              information can be revised in the "Queries/0_query_info.csv"
              file afterward if necessary.

optional arguments:
  -h, --help  show this help message and exit
```

3.7 amoebae list_queries

```
usage: amoebae [-h]

Print a list of all usable query files in the query directory as defined in
the settings file.

optional arguments:
  -h, --help  show this help message and exit
```

3.8 amoebae get_redun_hits

```
usage: amoebae [-h] [--csv_file CSV_FILE] [--query_name QUERY_NAME]
               [--query_list_file QUERY_LIST_FILE] [--db_name DB_NAME]
               [--db_list_file DB_LIST_FILE] [--query_title QUERY_TITLE]
               [--outdir OUTDIR]
               [--blast_report_evalue_cutoff BLAST_REPORT_EVALUE_CUTOFF]
               [--blast_max_target_seqs BLAST_MAX_TARGET_SEQS]
               [--hmmer_report_evalue_cutoff HMMER_REPORT_EVALUE_CUTOFF]
               [--hmmer_report_score_cutoff HMMER_REPORT_SCORE_CUTOFF]
               [--max_number_of_hits_to_summarize MAX_NUMBER_OF_HITS_TO_SUMMARIZE]
               [--num_threads_similarity_searching NUM_THREADS_SIMILARITY_SEARCHING
    ]
               [--predict_redun_hit_selection]
               srch_dir

Run searches with queries to find redundant hits in databases (for
interpreting results).

positional arguments:
  srch_dir              Path to directory that will contain output directory
                        as a subdirectory.

optional arguments:
  -h, --help            show this help message and exit
  --csv_file CSV_FILE   Path to spreadsheet to append summary of result to for
                        manual annotation. (default: None)
  --query_name QUERY_NAME
                        Query filename to use (not full path). (default: None)
  --query_list_file QUERY_LIST_FILE
                        Path to file containing a list of query files to use,
                        if no query_name is specified (or all queries by
                        default). (default: None)
  --db_name DB_NAME     Name of database file in the database directory in
                        which to do searches (not full path). (default: None)
  --db_list_file DB_LIST_FILE
                        Path to file containing a list of database files to
                        use (if no db_name specified). (default: None)
  --query_title QUERY_TITLE
```

```
                        Name to be assigned to hits in databases that may be
                        considered redundant with a search query to which the
                        same title is assigned, otherwise it is taken from the
                        query info spreadsheet specified in the settings.py
                        file ('query_info_csv'). (default: None)
  --outdir OUTDIR       Path to directory to write search results to.
                        (default: None)
  --blast_report_evalue_cutoff BLAST_REPORT_EVALUE_CUTOFF
                        Maximum E-value for reporting BLAST hits. (default:
                        0.05)
  --blast_max_target_seqs BLAST_MAX_TARGET_SEQS
                        Maximum BLAST target sequences to consider. (default:
                        500)
  --hmmer_report_evalue_cutoff HMMER_REPORT_EVALUE_CUTOFF
                        Maximum E-value for reporting HMMer hits. (default:
                        0.05)
  --hmmer_report_score_cutoff HMMER_REPORT_SCORE_CUTOFF
                        Minimum sequence score for reporting HMMer hits.
                        (default: 5)
  --max_number_of_hits_to_summarize MAX_NUMBER_OF_HITS_TO_SUMMARIZE
                        Absolute maximum number of hits (BLAST, HMMer, etc) to
                        summarize in the output spreadsheet. This is important
                        when working with sequences with WD40 domains, for
                        example. (default: 50)
  --num_threads_similarity_searching NUM_THREADS_SIMILARITY_SEARCHING
                        Number of threads to use for running searches.
                        (default: 4)
  --predict_redun_hit_selection
                        Write a copy of the output spreadsheet with '+' in
                        rows for hits that may be specific to each query
                        title, due to not being retrieved as top hits by
                        queries associated with different query titles.
                        (default: False)

Recommendation: For most analyses, use the --query_name option and the
--db_name option, and run the get_redun_hits command for each query
separately. Otherwise, there will be redundant information in the output
spreadsheet(s).
```

3.9 amoebae setup_fwd_srch

```
usage: amoebae [-h] [--outdir OUTDIR] srch_dir query_list_file db_list_file

Make a directory in which to write output files from similarity searches.

positional arguments:
  srch_dir         Path to directory that will contain output directory as a
                   subdirectory.
  query_list_file  Path to file with list of queries to search with.
  db_list_file     Path to file with list of databases to search with.

optional arguments:
  -h, --help       show this help message and exit
  --outdir OUTDIR  Path to directory to put search results into (so that this
                   step can be piped together with other commands). (default:
                   None)

Note: Use the bash script to run forward searches on a remote server.
```

3.10 amoebae run_fwd_srch

```
usage: amoebae [-h] [--blast_report_evalue_cutoff BLAST_REPORT_EVALUE_CUTOFF]
               [--blast_max_target_seqs BLAST_MAX_TARGET_SEQS]
               [--hmmer_report_evalue_cutoff HMMER_REPORT_EVALUE_CUTOFF]
               [--hmmer_report_score_cutoff HMMER_REPORT_SCORE_CUTOFF]
               [--num_threads_similarity_searching NUM_THREADS_SIMILARITY_SEARCHING
   ]
               fwd_srch_dir

Perform searches with original queries into subject databases.

positional arguments:
  fwd_srch_dir          Path to directory that will contain forward search
                        output files.

optional arguments:
  -h, --help            show this help message and exit
  --blast_report_evalue_cutoff BLAST_REPORT_EVALUE_CUTOFF
                        Maximum E-value for reporting BLAST hits. (default:
                        0.05)
  --blast_max_target_seqs BLAST_MAX_TARGET_SEQS
                        Maximum BLAST target sequences to consider. (default:
                        500)
  --hmmer_report_evalue_cutoff HMMER_REPORT_EVALUE_CUTOFF
                        Maximum E-value for reporting HMMer hits. (default:
                        0.05)
  --hmmer_report_score_cutoff HMMER_REPORT_SCORE_CUTOFF
                        Minimum sequence score for reporting HMMer hits.
                        (default: 5)
  --num_threads_similarity_searching NUM_THREADS_SIMILARITY_SEARCHING
                        Number of threads to use for running searches.
                        (default: 4)
```

3.11 amoebae sum_fwd_srch

```
usage: amoebae [-h] [--max_evalue MAX_EVALUE]
               [--max_gap_between_tblastn_hsps MAX_GAP_BETWEEN_TBLASTN_HSPS]
               [--do_not_use_exonerate]
               [--exonerate_score_threshold EXONERATE_SCORE_THRESHOLD]
               [--max_hits_to_sum MAX_HITS_TO_SUM]
               [--max_length_diff MAX_LENGTH_DIFF]
               fwd_srch_out csv_file

Append information about forward searches to csv summary file (this is used to
organize reverse searches). For TBLASTN searches (protein queries, nucleotide
target sequences), HSPs are clustered into groups that are close enough within
the target sequence to potentially represent exons from the same coding
sequence. The nucleotide subsequences in which these clusters of HSPs are
found are then analyzed using exonerate to identify and translate potential
exons, in "protein2genome" mode, because exonerate, unlike TBLASTN, attempts
to identify exon boundaries, yielding translations that are less likely to
include translations of non-coding regions outside exons (which might include
apparent stop codons).

positional arguments:
  fwd_srch_out          Path to directory where forward search results were
                        written.
  csv_file              Path to summary spreadsheet (CSV) file, which may
                        already contain search summaries, or may not exist
                        yet.

optional arguments:
  -h, --help            show this help message and exit
  --max_evalue MAX_EVALUE
                        Maximum E-value threshold for reporting forward search
                        hits. (default: 0.0005)
  --max_gap_between_tblastn_hsps MAX_GAP_BETWEEN_TBLASTN_HSPS
                        Maximum number of nucleotide bases between TBLASTN
                        HSPs to be considered part of the same gene locus.
                        This is important, because it will be assumed that HSP
                        separated by more than this number of nucleotide bases
                        are not part of the same gene or TBLASTN "hit".
                        (default: 10000)
  --do_not_use_exonerate
                        Override the default use of exonerate to identify
                        coding sequences and translations, and just use
                        TBLASTN instead. This option is provided because
                        concatenated TBLASTN HSPs may be more inclusive of
                        sequences within the target sequence, and the results
                        of TBLASTN and exonerate may need to be compared.
                        Also, note that HSPs identified by TBLASTN but for
                        which exonerate yields no alignments will be ignored
                        if exonerate is used. (default: False)
  --exonerate_score_threshold EXONERATE_SCORE_THRESHOLD
                        Set score threshold to be applied when running
                        exonerate on nucleotide sequences identified by
                        TBLASTN. The default for setting of exonerate is 100,
                        but a lower score is set as default here, because
                        otherwise exonerate cannot identify some of the
                        seqeunces identified by TBLASTN. This option is only
                        relevant if using exonerate. (default: 10)
  --max_hits_to_sum MAX_HITS_TO_SUM
                        Maximum number of forward search hits to list in the
                        summary spreadsheet. If zero, then reverse searches
                        will be performed for all hits. (default: 0)
  --max_length_diff MAX_LENGTH_DIFF
                        Maximum number of amino acid residues length
                        difference allowed between the original query and the
                        forward hit sequence. If -1, then a maximum length
                        cutoff will not be applied. (default: -1)
```

3.12 amoebae setup_rev_srch

```
usage: amoebae [-h] [--outdir OUTDIR] [--aasubseq] [--nafullseq]
               srch_dir csv_file databases

Make directory in which to write results of reverse searches.

positional arguments:
  srch_dir         Path to directory that will contain output directory as a
                   subdirectory.
  csv_file         Path to summary spreadsheet (CSV) file, which contains a
                   summary of forward search(es).
  databases        Database filename (in database directory) or path to file
                   with list of database filenames. Note that filenames are
                   needed, not file paths.

optional arguments:
  -h, --help       show this help message and exit
  --outdir OUTDIR  Path to directory to put search results into (so that this
                   step can be piped together with other commands). (default:
                   None)
  --aasubseq       Use only the portion of each (amino acid) forward hit
                   sequence that aligns to the original query used (top HSP
                   subject sequence). This is default for nucleotide hits.
                   (default: False)
  --nafullseq      Use the full (nucleic acid) forward hit sequence. This is
                   default for amino acid hits. (default: False)
```

3.13 amoebae run_rev_srch

```
usage: amoebae [-h] [--blast_report_evalue_cutoff BLAST_REPORT_EVALUE_CUTOFF]
               [--blast_max_target_seqs BLAST_MAX_TARGET_SEQS]
               [--hmmer_report_evalue_cutoff HMMER_REPORT_EVALUE_CUTOFF]
               [--hmmer_report_score_cutoff HMMER_REPORT_SCORE_CUTOFF]
               [--num_threads_similarity_searching NUM_THREADS_SIMILARITY_SEARCHING
    ]
               rev_srch_dir

Perform searches with forward search hit sequences as queries into the
original query databases.

positional arguments:
  rev_srch_dir          Path to directory that will contain output of
                        searches.

optional arguments:
  -h, --help            show this help message and exit
  --blast_report_evalue_cutoff BLAST_REPORT_EVALUE_CUTOFF
                        Maximum E-value for reporting BLAST hits. (default:
                        0.05)
  --blast_max_target_seqs BLAST_MAX_TARGET_SEQS
                        Maximum BLAST target sequences to consider. (default:
                        500)
  --hmmer_report_evalue_cutoff HMMER_REPORT_EVALUE_CUTOFF
                        Maximum E-value for reporting HMMer hits. (default:
                        0.05)
  --hmmer_report_score_cutoff HMMER_REPORT_SCORE_CUTOFF
                        Minimum sequence score for reporting HMMer hits.
                        (default: 5)
  --num_threads_similarity_searching NUM_THREADS_SIMILARITY_SEARCHING
                        Number of threads to use for running searches.
                        (default: 4)
```

3.14 amoebae sum_rev_srch

```
usage: amoebae [-h] [--redun_hit_csv REDUN_HIT_CSV]
               [--min_evaldiff MIN_EVALDIFF] [--aasubseq] [--nafullseq]
               [--max_rev_srchs MAX_REV_SRCHS]
               csv_file rev_srch_out

Append information about reverse searches to csv summary file. Use information
from redundant hit csv file to interpret results.

positional arguments:
  csv_file              Path to summary spreadsheet (CSV) file, which may
                        already contain reverse search summaries.
  rev_srch_out          Path to directory where reverse search results were
                        written.

optional arguments:
  -h, --help            show this help message and exit
  --redun_hit_csv REDUN_HIT_CSV
```

```
                        Path to spreadsheet (CSV) file, which specifies which
                        hits are redundant positive hits for a given query
                        (query title) in a given database. If this is not
                        provided, then it is assumed that any and all reverse
                        search hits are equivalent to/redundant with the
                        original query. (default: None)
  --min_evaldiff MIN_EVALDIFF
                        Minimum difference in E-value order of magnitude
                        between top reverse search hit and first reverse
                        search hit that is not redundant with the original
                        query. (default: 5)
  --aasubseq            Use only the portion of each (amino acid) forward hit
                        sequence that aligns to the original query used (top
                        HSP subject sequence). This is default for nucleotide
                        hits. Must be selected if selected when the
                        setup_rev_srch command was run. (default: False)
  --nafullseq           Use the full (nucleic acid) forward hit sequence. This
                        is default for amino acid hits. Must be selected if
                        selected when the setup_rev_srch command was run.
                        (default: False)
  --max_rev_srchs MAX_REV_SRCHS
                        Maximum number of forward search hits to perform
                        reverse searches for per query database. If zero, then
                        reverse searches will be performed for all hits.
                        (default: 0)
```

3.15 amoebae interp_srchs

```
usage: amoebae [-h] [--fwd_only] [--fwd_evalue_cutoff FWD_EVALUE_CUTOFF]
               [--rev_evalue_cutoff REV_EVALUE_CUTOFF]
               [--hmmer_cutoff HMMER_CUTOFF] [--redun_hits]
               [--out_csv_path OUT_CSV_PATH]
               csv_file

Interpret search results based on final summary, which provides a basis for
further analyses of positive hits.

positional arguments:
  csv_file              Path to spreadsheet with forward and reverse search
                        results.

optional arguments:
  -h, --help            show this help message and exit
  --fwd_only            Interpret forward searches based on score (HMMer)
                        cutoff. (default: False)
  --fwd_evalue_cutoff FWD_EVALUE_CUTOFF
                        Specify an (more stringent) E-value cutoff for forward
                        search results. (default: None)
  --rev_evalue_cutoff REV_EVALUE_CUTOFF
```

```
                        Specify an (more stringent) E-value cutoff for reverse
                        search results. (default: None)
  --hmmer_cutoff HMMER_CUTOFF
                        Specify a score that hits must exceed to be included.
                        (default: 20)
  --redun_hits          Interpret which hits are redundant in output of
                        get_redun_hits command. (default: False)
  --out_csv_path OUT_CSV_PATH
                        Optionally specify an output file path, so that this
                        command can be piped together with others. (default:
                        None)
```

3.16 amoebae find_redun_seqs

```
usage: amoebae [-h] [--out_csv_path OUT_CSV_PATH]
               [--remove_tblastn_hits_at_annotated_loci]
               [--just_look_for_genes_in_gff3] [--ignore_gff3]
               [--allow_internal_stops ALLOW_INTERNAL_STOPS]
               [--min_length MIN_LENGTH]
               [--min_percent_length MIN_PERCENT_LENGTH]
               [--min_percent_query_cover MIN_PERCENT_QUERY_COVER]
               [--overlap_required] [--max_percent_ident MAX_PERCENT_IDENT]
               [--min_alig_res_in_overlap MIN_ALIG_RES_IN_OVERLAP]
               [--min_ident_res_in_overlap MIN_IDENT_RES_IN_OVERLAP]
               [--min_sim_res_in_overlap MIN_SIM_RES_IN_OVERLAP]
               [--min_ident_span_len MIN_IDENT_SPAN_LEN]
               [--min_sim_span_len MIN_SIM_SPAN_LEN]
               [--min_percent_ident_in_overlap MIN_PERCENT_IDENT_IN_OVERLAP]
               [--min_percent_sim_in_overlap MIN_PERCENT_SIM_IN_OVERLAP]
               [--min_percent_overlap MIN_PERCENT_OVERLAP]
               [--plot_hit_exclusion] [--add_ali_col]
               csv_file

Identify hit sequences likely encoded by the same gene loci in the genome of a
given species, or otherwise not representing paralogous genes. Criteria are
applied in this order: 1. Peptide hits with the same ID as higher-ranking hits
for the same query (query title) are excluded. 2. Nucleotide hits for the same
loci as peptide sequence hits are excluded. 3. Sequences with internal stop
codons are excluded, as these are potentially pseudogenes. 4. Sequences are
excluded if they do not meet several minimum length criteria: Absolute minimum
length (in amino acids) and percent query cover. 5. Sequences are excluded if
they do not overlap to a specified degree with all included higher-ranking
hits for the same query (query title) in sequence data for the same
species/genome. This is determined by algorithmically comparing pairs of
sequences aligned to a reference alignment of homologues, and several minimum
measures of alignment overlap may be specified. 6. Secondary hit sequences are
excluded if they do not meet a specified maximum percent identity threshold.
Highly identical sequences may result from false segmental duplications in the
genome assembly, may represent alleles, etc. Note: Applying these criteria
```

```
requires a column to be manually added to the input csv file prior to running
with the header "Alignment for sequence comparison" and filled with the
appropriate alignment name to use (one for each query title, as listed in the
"Query title" column). Alternatively, you can run this command with the
--add_ali_col option to automatically identify appropriate alignments among
your aligned FASTA queries used for running HMMer searches. If no alignment
(.afaa) file can be found, then the first single sequence query file (.faa)
that appears in the summary CSV file will be used instead.

positional arguments:
  csv_file              Path to spreadsheet with interpreted search results
                        outputted by the interp_srchs command.

optional arguments:
  -h, --help            show this help message and exit
  --out_csv_path OUT_CSV_PATH
                        Optionally specify an output file path, so that this
                        command can be piped together with others. (default:
                        None)
  --remove_tblastn_hits_at_annotated_loci
                        Ignore tblastn hits that overlap with any previously
                        annotated loci. The rationale for this would be that
                        the corresponding protein sequences should have been
                        retrieved if the tblastn hit were a true positive
                        anyway. If this option is not specified, then
                        sequences will still be excluded if they specifically
                        correspond to the same loci as do higher-ranking hits.
                        (default: False)
  --just_look_for_genes_in_gff3
                        When looking for records in GFF3 annotation files that
                        overlap with subsequences identified by similarity
                        searching (TBLASTN), ignore records that are not
                        explicitly "gene" (for example, "CDS", "mRNA", and
                        "exon"). This option should probably not be selected,
                        because in some GFF3 annotation files do not include
                        "gene" records, but do include predicted coding
                        sequences for genes. (default: False)
  --ignore_gff3         Disregard any information regarding redundancy of
                        identified nucleotide sequences with identified
                        protein sequences that may be found in GFF3 annotation
                        files. (default: False)
  --allow_internal_stops ALLOW_INTERNAL_STOPS
                        Include sequences that have internal stop codons
                        (anywhere other than the N-terminal position).
                        (default: True)
  --min_length MIN_LENGTH
                        Absolute minimum length (in AA) of a hit sequence to
                        be considered a potential distinct paralogue.
                        (default: 55)
  --min_percent_length MIN_PERCENT_LENGTH
                        Minimum length (in AA) of a hit sequence as a
                        percentage of query length for the hit to be
                        considered a potential distinct paralogue. (default:
                        15)
  --min_percent_query_cover MIN_PERCENT_QUERY_COVER
                        Minimum number of residues aligning with the original
                        query as a percentage of the original query sequence
                        length. (default: 0)
  --overlap_required    True if hits must overlap with a higher-ranking hit to
                        be considered potential unique paralogues. (default:
                        False)
  --max_percent_ident MAX_PERCENT_IDENT
                        Maximum percent identity (among aligning residues) for
                        evaluating whether two sequences are redundant or not
                        (secondary hits showing a percent identity with a
                        higher-ranking hit exceeding this value will be
                        excluded). (default: 98.0)
  --min_alig_res_in_overlap MIN_ALIG_RES_IN_OVERLAP
                        Minimum number of residues which must align for two
                        sequences to be considered as potentially distinct
                        hits. This is only relevant if the overlap_required
                        option is specified. (default: 20)
  --min_ident_res_in_overlap MIN_IDENT_RES_IN_OVERLAP
                        Minimum number of aligning residues which must be
                        identical for two sequences to be considered as
                        potentially distinct hits. This is only relevant if
                        the overlap_required option is specified. (default:
                        10)
  --min_sim_res_in_overlap MIN_SIM_RES_IN_OVERLAP
                        Minimum number of aligning residues which must be
                        similar for two sequences to be considered as
                        potentially distinct hits. This is only relevant if
                        the overlap_required option is specified. (default:
                        15)
  --min_ident_span_len MIN_IDENT_SPAN_LEN
                        Minimum number of aligning residues which are
                        identical that must exist in at least one continuous
                        span for two sequences to be considered as potentially
                        distinct hits (not counting positions where both
                        sequences have gaps). This is only relevant if the
                        overlap_required option is specified. (default: 0)
  --min_sim_span_len MIN_SIM_SPAN_LEN
                        Minimum number of aligning residues which are similar
                        (or identical) that must exist in at least one
                        continuous span for two sequences to be considered as
                        potentially distinct hits (not counting positions
                        where both sequences have gaps). This is only relevant
                        if the overlap_required option is specified. (default:
```

```
                        0)
  --min_percent_ident_in_overlap MIN_PERCENT_IDENT_IN_OVERLAP
                        Minimum percent identity between the two sequences of
                        interest in the alignment.This is only relevant if the
                        overlap_required option is specified. (default: 0)
  --min_percent_sim_in_overlap MIN_PERCENT_SIM_IN_OVERLAP
                        Minimum percent similarity (including identity)
                        between the two sequences of interest in the
                        alignment.This is only relevant if the
                        overlap_required option is specified. (default: 0)
  --min_percent_overlap MIN_PERCENT_OVERLAP
                        Minimum number of aligning residues between the two
                        sequences of interest as a percentage of the length of
                        the second sequence (the last sequence in the
                        alignment), not including gaps, for the two sequences
                        to be considered as potentially distinct hits. This is
                        only relevant if the overlap_required option is
                        specified. (default: 0)
  --plot_hit_exclusion  Plot number of hits excluded by the various criteria
                        applied. (default: False)
  --add_ali_col         Add a column to the csv file listing which alignment
                        file in the queries directory to use for comparing
                        sequences. Aligned FASTA queries are selected that
                        match the query titles of the original queries used to
                        retrieve each of the relevant hits listed in the csv
                        file. No other options need to be specified in this
                        case. (default: False)
```

3.17 amoebae plot

```
usage: amoebae [-h] [--csv_file2 CSV_FILE2] [--complex_info COMPLEX_INFO]
               [--row_order ROW_ORDER] [--out_pdf OUT_PDF]
               csv_file

Plot results of similarity search and sequence classification analyses. The
outputs are PDF files.

positional arguments:
  csv_file              Path to a spreadsheet with the relevant results to be
                        plotted. This can be either a CSV file output of the
                        sum_rev_srch command or from the find_redun_seqs
                        command. If the output of the sum_rev_srch command is
                        used, however, redundant hits will be counted (e.g.,
                        BLASTP and TBLASTN hits corresponding to the same or
                        highly identical genomic loci).

optional arguments:
  -h, --help            show this help message and exit
  --csv_file2 CSV_FILE2
```

```
                        Path to a second spreadsheet with relevant results to
                        be compared to the first and plotted. (default: None)
  --complex_info COMPLEX_INFO
                        Path to file that specifies which query titles
                        represent components of which protein complexes (or
                        otherwise grouped proteins). (default: None)
  --row_order ROW_ORDER
                        Path to file that specifies the order in which data
                        for each species will be displayed. (default: None)
  --out_pdf OUT_PDF     Path to output pdf file. (default: None)
```

3.18 amoebae add_to_models

```
usage: amoebae [-h]
               model_name alignment tree_topology subs_model type_seqs taxon

Add a phylogenetic model for relationships between members of a gene family
(sequence_data matrix, data type, tree topology, type sequence defining each
clade of interest, and substitution model) to a directory for use in
classifying sequence (via the 'phylo_class' command.

positional arguments:
  model_name     An arbitrary name for the model (which will refer to the
                 alignment, tree, substitution model, etc. collectively).
  alignment      A multiple amino acid sequence alignment in nexus format.
  tree_topology  Text file containing a tree (identified previously using
                 MrBayes, etc) containing the names of all the sequences in
                 the alignment, in newick format.
  subs_model     The name of the substitution model used to recover the
                 provided topology (chosen with ModelFinder or similar
                 software).
  type_seqs      Names of sequences (sequence headers) that are to be used to
                 define clades of interest. A csv file with seq names in one
                 column and clade names in the next column.
  taxon          Taxonomic group represented in the model (e.g., "Eukaryotes",
                 or "Amorphea").

optional arguments:
  -h, --help     show this help message and exit
```

3.19 amoebae list_models

```
usage: amoebae [-h]

Print a list of all usable model/reference tree names in the models directory
as defined in the settings file.

optional arguments:
  -h, --help  show this help message and exit
```

3.20 amoebae get_alt_topos

```
usage: amoebae [-h] [--polytomy] [--not_polytomy_clades]
               [--keep_original_backbone] [--iqtree_au_test]
               model_name out_dir_path

Take a tree and make copies with every alternative topology for the branches
connecting the clades of interest. Output as additional models in the Models
directory.

positional arguments:
  model_name            Name of model/backbone tree to modify (other info
                        provided in the model info csv file).
  out_dir_path          Path to directory in which output directory will be
                        written.

optional arguments:
  -h, --help            show this help message and exit
  --polytomy            Just make one big polytomy connecting the clades of
                        interest intead of making alternative bifurcating
                        trees. (default: False)
  --not_polytomy_clades
                        Do not make subtrees/clades of interest polytomies in
                        output topologies. (default: False)
  --keep_original_backbone
                        Keep the original backbone topology instead of
                        generating a polytomy or alternative resolved
                        topologies. (default: False)
  --iqtree_au_test      Test all the relevant alternative topologies against
                        each other using Approximately Unbiased (AU) test with
                        IQ-tree. (default: False)
```

3.21 amoebae prune

```
usage: amoebae [-h] [--include_seqs] [--output_file OUTPUT_FILE]
               tree_file alignment name_replace_table

Identify sequences in a tree, and remove them from a given alignment for
further phylogenetic analysis.

positional arguments:
  tree_file             Tree in newick format (coded names, because ETE3
                        cannot parse taxon names with space characters without
                        quotation marks around them).
  alignment             Dataset used to make the tree (nexus alignment)
                        (original alignment with original taxon names either
                        trimmed or untrimmed).
  name_replace_table    File for decoding names in input tree file.

optional arguments:
  -h, --help            show this help message and exit
  --include_seqs        Include only listed sequences/nodes instead of
                        removing them. (default: False)
  --output_file OUTPUT_FILE
                        Path to output file. (default: None)
```

3.22 amoebae auto_prune

```
usage: amoebae [-h]
               [--max_bl_iqr_above_third_quartile MAX_BL_IQR_ABOVE_THIRD_QUARTILE]
               [--remove_redun_seqs REMOVE_REDUN_SEQS]
               [--remove_redun_seqs_threshold REMOVE_REDUN_SEQS_THRESHOLD]
               [--output_file OUTPUT_FILE]
               in_dir

Automatically identify sequences in a tree, and remove them from a given
alignment for further phylogenetic analysis.

positional arguments:
  in_dir                Path to directory that contains the phylogenetic
                        analysis output files (sequence name conversion table
                        file and original nexus alignment file can be in the
                        parent directory to this directory as long as their
                        names are mostly identical.

optional arguments:
  -h, --help            show this help message and exit
  --max_bl_iqr_above_third_quartile MAX_BL_IQR_ABOVE_THIRD_QUARTILE
                        Inclusion threshold for number of interquartile ranges
                        above the third quartile of terminal branch lengths
                        the length of a terminal branch can be before it is
                        considered an outlier (length is total distance from
                        root node after rooting on midpoint, or the longest
                        terminal branch on either side of the midpoint).
                        (default: 1.5)
  --remove_redun_seqs REMOVE_REDUN_SEQS
                        Remove taxonomically redundant sequences (longest
                        branch of two sister branches when both are sequences
                        from the same species. (default: True)
  --remove_redun_seqs_threshold REMOVE_REDUN_SEQS_THRESHOLD
                        Minimum support required to consider one of two sister
                        branches/sequences taxonomically redundant. Note: only
                        used if the remove_redun_seqs option is specified.
                        (default: 0.95)
  --output_file OUTPUT_FILE
```

```
                        Path to output file. (default: None)
```

3.23 amoebae reduce_tree

```
usage: amoebae [-h] [--output_file OUTPUT_FILE] alignment tree_file

Remove terminal nodes from a given tree if there are not any sequences with
the same name in a given alignment.

positional arguments:
  alignment             Alignment in nexus format with sequences representing
                        a subset of those represented in the input tree.
  tree_file             Tree in newick format.

optional arguments:
  -h, --help            show this help message and exit
  --output_file OUTPUT_FILE
                        Path to output file. (default: None)
```

3.24 amoebae constrain_mb

```
usage: amoebae [-h] [--out_alignment OUT_ALIGNMENT] alignment tree

Add constraint commands to MrBayes input file.

positional arguments:
  alignment             Nexus alignment for input to Mrbayes (without any
                        constraint commands).
  tree                  Tree in newick format with same taxon names as in
                        alignment. To be used as a topology constraint (all
                        nodes).

optional arguments:
  -h, --help            show this help message and exit
  --out_alignment OUT_ALIGNMENT
                        Path to nexus alignment for input to Mrbayes with
                        constraints added. (default: None)
```

3.25 amoebae visualize_tree

```
usage: amoebae [-h] [--root_taxon ROOT_TAXON] [--highlight_paralogues]
              [--add_clade_names_from_file]
              input_directory method

Parse phylogenetic analysis output files in a given directory, and write
human-readable tree figures to PDF files.

```

```
positional arguments:
  input_directory       Path to directory containing input files (must contain
                        a .table file for decoding taxon names.
  method                Name of tree searching program used. Either iqtree,
                        raxml, or mrbayes accepted.

optional arguments:
  -h, --help            show this help message and exit
  --root_taxon ROOT_TAXON
                        Name of species to root on (e.g.,
                        "Klebsormidium_nitens").
  --highlight_paralogues
                        Highlight clades that contain paralogues found in at
                        least one other clade in the tree.
  --add_clade_names_from_file
                        Use a file in the parent directory with clade names
                        corresponding to representative sequences to add clade
                        names to all the taxon names in the output trees.
```

3.26 amoebae replace_seqs

```
usage: amoebae [-h] [--fasta_file FASTA_FILE] alignment

Replace sequences in an alignment the full-length sequences from the relevant
file(s) in the Genomes directory, or with their top hits in a given fasta
file. And, align, mask, and trim the identified sequences to the input
alignment

positional arguments:
  alignment             Path to multiple sequence alignment file in nexus
                        format (trimmed alignment).

optional arguments:
  -h, --help            show this help message and exit
  --fasta_file FASTA_FILE
                        Path to file containing sequences with which to
                        replace sequences in the alignment. If this option is
                        not specified, then full-length sequences will be
                        retrieved from files in the Genomes directory.
```

3.27 amoebae csv_to_fasta

```
usage: amoebae [-h] [--output_dir OUTPUT_DIR] [--abbrev] [--paralogue_names]
               [--only_descr] [--subseq] [--all_hits] [--split_by_query_title]
               [--split_by_top_rev_srch_hit SPLIT_BY_TOP_REV_SRCH_HIT]
               [--split_to_query_fastas]
               csv_file

```

```
Extract sequences described in a spreadsheet output by AMOEBAE, and write to a
file in FASTA format.

positional arguments:
  csv_file              Path to csv file listing sequences.

optional arguments:
  -h, --help            show this help message and exit
  --output_dir OUTPUT_DIR
                        Path for output directory to contain FASTA files.
                        (default: None)
  --abbrev              Add species name instead of sequence description from
                        fasta header. Applicable when the output file is to be
                        used for alignment and phylogenetic analysis.
                        (default: False)
  --paralogue_names     Use species name, query title, and paralogue number
                        instead of sequence description from fasta header.
                        Applicable when the output file is to be used for
                        alignment and phylogenetic analysis. Does not work if
                        the abbrev option is specified. (default: False)
  --only_descr          Use the description but not the ID as the new fasta
                        sequence header. Does not work if the abbrev option is
                        specified. (default: False)
  --subseq              Write subsequences that aligned to forward search
                        query, instead of the full sequences. (default: False)
  --all_hits            Write all forward hits listed in the input csv file.
                        (default: False)
  --split_by_query_title
                        Write sequences to files according to the query title
                        of the query which retrieved them in a similarity
                        search. (default: False)
  --split_by_top_rev_srch_hit SPLIT_BY_TOP_REV_SRCH_HIT
                        Write sequences to files according to the top hit that
                        they retrieve in a reverse search, for each sequence
                        that meets the reverse search criteria. (Provide the
                        reverse search identifier, eg,
                        "rev_srch_20180924122402-1") (default: None)
  --split_to_query_fastas
                        Write sequences to separate files with filenames that
                        can be easily parsed for loading the the files as
                        queries using the add_to_queries command. (default:
                        False)
```

3.28 amoebae check_depend

```
usage: amoebae [-h]

Check that all the dependencies (other than python modules) are properly
installed and useable.
```

```

optional arguments:
  -h, --help  show this help message and exit
```

3.29 amoebae check_imports

```
usage: amoebae [-h]

Check that all the import statements used in the AMOEBAE repository run
without error.

optional arguments:
  -h, --help  show this help message and exit
```

4 Miscellaneous scripts

Several scripts of less general applicablity than the amoebae commands descibed above are included in the AMOEBAE toolkit. See the amoebae/misc_scripts directory (`https://github.com/laelbarlow/amoebae/tree/master/misc_scripts`). Most scripts have information regarding usage in the files themselves. More detailed information regarding some of these scripts may be added to this documentation in the future.

5 References

Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T.L. (2009). BLAST+: Architecture and applications. *BMC Bioinformatics*, 10(1):421. doi:10.1186/1471-2105-10-421.

Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., and de Hoon, M.J.L. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423. doi:10.1093/bioinformatics/btp163.

Eddy, S.R. (1998). Profile hidden Markov models. *Bioinformatics*, 14(9):755–763. doi:10.1093/bioinformatics/14.9.755.

Edgar, R.C. (2004). MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797. doi:10.1093/nar/gkh340.

Huerta-Cepas, J., Serra, F., and Bork, P. (2016). ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. *Molecular Biology and Evolution*, 33(6):1635–1638. doi:10.1093/molbev/msw046.

Hunter, J.D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science Engineering*, 9(3):90–95. doi:10.1109/MCSE.2007.55.

Larson, R.T., Dacks, J.B., and Barlow, L.D. (2019). Recent gene duplications dominate evolutionary dynamics of adaptor protein complex subunits in embryophytes. *Traffic*, page tra.12698. doi:10.1111/tra.12698.

Li, L. (2003). OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes. *Genome Research*, 13(9):2178–2189. doi:10.1101/gr.1224503.

Nguyen, L.T., Schmidt, H.A., von Haeseler, A., and Minh, B.Q. (2015). IQ-TREE: A Fast and Effective Stochastic Algorithm for Estimating Maximum-Likelihood Phylogenies. *Molecular Biology and Evolution*, 32(1):268–274. doi:10.1093/molbev/msu300.

Slater, G. and Birney, E. (2005). [No title found]. *BMC Bioinformatics*, 6(1):31. doi:10.1186/1471-2105-6-31.