eggNOG mapper v2.1.5 to v2.1.12

Overview

EggNOG-mapper (a.k.a. `emapper.py` or just *emapper*) is a tool for fast functional annotation of novel sequences. It uses precomputed orthologous groups (OGs) and phylogenies from the eggNOG database (http://eggnogdb.embl.de/) to transfer functional information from fine-grained orthologs only.
Common uses of eggNOG-mapper include the annotation of novel genomes, transcriptomes or even metagenomic gene catalogs.
The use of orthology predictions for functional annotation permits a higher precision than traditional homology searches (i.e. BLAST searches), as it avoids transferring annotations from close paralogs (duplicate genes with a higher chance of being involved in functional divergence).
Benchmarks comparing different eggNOG-mapper options against BLAST and InterProScan are available at
https://github.com/jhcepas/emapper-benchmark/blob/master/benchmark_analysis.ipynb.
EggNOG-mapper is also available as a public online resource:
http://eggnog-mapper.embl.de

What's new in eggNOG-mapper v2

development branches

- Fix #454 issue. Prokka GFF files have FASTA sequences after a `##FASTA` header, causing the parsing of GFF fields to crash. As for now, all lines below the `##FASTA` header are ignored.

v2.1.12

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.12

- Fix bug parsing annotations files, which affects gff decoration when `--resume` is used and `.emapper.annotations` file already exists.
- Updated novel families DB version and added novel families .pkl DB for annotation, which can be downloaded from http://eggnog6.embl.de/download/novel_fams-1.0.1 or using `download_eggnog_data.py -D -F`

- Using the novel families search `-m novel_fams` now provides new annotation fields in `.emapper.annotations` output file.

v2.1.11

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.11

- Bug on issue #432
- Bug when running `--decorate_gff yes` along with `--resume`.

v2.1.10

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.10

- Bug fixes (see Pull Requests), including compatibility with python 3.11.
- HMMER database downloaded using `download_eggnog_data.py` can be saved with a user specified name (see Pull Requests).
- Paths of input and output files should now accept white spaces.
- `--genepred prodigal` should accept now gzipped files as input, fixing issue #439.
- Added `--timeout_load_server` to control the number of attempts made to fire up a hmmpgmd server, also used with the `--usemem` option.

v2.1.9

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.9

- Added the option to search queries against the novel families identified at https://doi.org/10.1101/2022.01.26.477801

1 - Download the novel families diamond database: `download_eggnog_data.py -F`
2 - Search your queries against the novel families: `emapper.py -m novel_fams`

v2.1.8

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.8

- Bug fixes related with hmmer searches using hmmpgmd servers (--usemem option, etc). Fixes #390

v2.1.7

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.7

- Changes in GFF decoration to (hopefully) fix issue #367 , in which positions were relative to the ORF (1-end) instead of relative to the contig which contains the ORF. Now, 2 GFF files will be created when using gene prediction and gff decoration: one ".genepred.gff" from either Prodigal or blastx-like hits, one ".decorated.gff" with the annotations added as attributes.

- Update the `--resume` procedure. Now, most files are created from scratch, being resumed the search (if a ".emapper.hits" exists) and annotations (if a ".emapper.hits" exists, or ".emapper.seed_orthologs" if using `--annotate_hits_table` option; and if a partial ".emapper.annotations" file exists).
- Now it should be possible to use Biopython 1.78 (python 3.9) which removed the BioAlphabet module.
- emapper.py's `--temp_dir` option is now linked to Diamond's `--tmpdir` option.
- Update / Bug fix (issue #328) regarding report of orthologs, where the seed ortholog was listed in every orthology category. Now the seed ortholog should be tagged as "seed" in the "orth_type" column.
- Help description and default values of parameters of `emapper.py` are now provided more consistently (PR #346 addressing issue #344, thanks @nick-youngblut). Also done by @nick-youngblut for all the other main scripts (PR #347).
- `--version` now reports the Diamond/MMseqs2 version found (either from PATH or from eggNOG-mapper bundled bin).
- Fixed a bug when transferring PFAM annotations from denovo hmmpgmd search and number of queries was large (>15000).

v2.1.6

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.6

- Updated citation of eggNOG-mapper.
- Updated citation of Diamond.
- Diamond's `--iterate` mode is now used by default. To disabled it when running `emapper.py`, use `--dmnd_iterate no`.
- Added `--dmnd_ignore_warnings` option, to activate Diamond's `--ignore-warnings` option.
- Added `--dmnd_algo` option, to control Diamond's `--algo` option. It can be used to perform a much faster search with small input sets of sequences using `--dmnd_algo ctg`.
- Added `--mp_start_method` option, to control the Python's multprocessing start method. Only use it when the default is not working properly in your OS.
- Minor changes related with `hmmpgmd` searches.

v2.1.5

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.5

- New Diamond version 2.0.11 bundled.
- Added new options for diamond mode (require using Diamond version 2.0.11): `--dmnd_iterate` (Diamond's `--iterate` option), `--sensmode fast`, and `--sensmode default`.

- Bug fix (issues #313 and #319), when parsing GFF files with "." in the "score" field.
- `--dmnd_frameshift INT` option to control Diamond's `--frameshift/-F` option.
- Bug fix / try to address the `main __spec__` bug (issue #299). For a fix in version 2.1.4, check pre-release https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.4-main_spec
- Minor progress report text fix.

v2.1.4

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.4

- `--excel` option to output annotations in .xlsx format.
- Recalibration of PFAM annotation from orthologs.

v2.1.3

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.3

- Bug fix (issue #301), when running `--pfam_realign realign` and there are seeds without annotations.
- Bug fix (issue #303), when running `--pfam_realign denovo` and there are 100 queries or more (so that `hmmpgmd` servers are used).
- Other minor changes (see issue #302).

v2.1.2

https://github.com/eggnogdb/eggnog-mapper/releases/tag/2.1.2

- Orthologs retrieved bottom-up from narrowest OG to max annotation level OG.
- Changes in format of annotations and gff output files (check Output format).

Requirements

Software Requirements

- Python 3.7 (or greater)
- BioPython 1.76 (*python package*) (*BioPython 1.78 should work since emapper version 2.1.7*)
- psutil 5.7.0 (*python package*)
- xlsxwriter 1.4.3 (*python package*), only if using the `--excel` option

- wget (*linux command*, required for downloading the eggNOG-mapper databases with `download_eggnog_data.py`, and to create new Diamond/MMseqs2 databases with `create_dbs.py`)
- sqlite (>=3.8.2)

Check also the Optional tools section below.

Storage Requirements

- ~40 GB for the eggNOG annotation databases (*eggnog.db* and *eggnog.taxa.db*)
- ~9 GB for Diamond database of eggNOG sequences (required if using `-m diamond`, which is the default search mode).
- ~11 GB for MMseqs2 database of eggNOG sequences (~86 GB if MMseqs2 index is created) (required if using `-m mmseqs`).
- ~3 GB for PFAM database (required if using `--pfam_realign` options for realignment of queries to PFAM domains).
- The size of eggNOG diamond/mmseqs databases created with `create_dbs.py` is highly variable, depending on the size of the chosen taxonomic groups.

Databases for specific taxonomic ranges can be downloaded (for HMMER) or created (for Diamond and MMseqs2). The size of these databases is highly variable. For the size of HMMER databases, check http://eggnog5.embl.de/#/app/downloads. For Diamond and MMseqs2 databases, DB size will depend on the number of proteins which are from those taxonomic ranges. Also, these proteins need to be downloaded to create the databases, and can be removed afterwards.

Other Requirements

- Using `--dbmem` loads the whole *eggnog.db* sqlite3 annotation database during the annotation step, and therefore requires ~44 GB of memory.
- Using the `--num_servers` option when running HMMER in server mode (a.k.a. `hmmgpmd`, which is used for `-m hmmer --usemem`, `--pfam_realign denovo` or `hmm_server.py`) loads the HMM database as many times as specified in the argument (e.g. `--pfam_realign denovo --num_servers 2` loads the PFAM database into memory twice, with up to roughly 2 GB per instance).

Installation

Pypi version

https://pypi.org/project/eggnog-mapper/

```
pip install eggnog-mapper
```

Conda (bioconda channel) version

https://anaconda.org/bioconda/eggnog-mapper

```
conda install -c bioconda -c conda-forge eggnog-mapper
```

GitHub release

https://github.com/eggnogdb/eggnog-mapper/releases

- Download the latest version of eggnog-mapper from the next link: https://github.com/eggnogdb/eggnog-mapper/releases/latest
- Decompress the `.tar.gz` or `.zip` file
- Enter the decompressed directory and install the dependencies, either with:
    - setuptools: `python setup.py install`
    - pip: `pip install -r requirements.txt`
    - conda: `conda install --file requirements.txt`

Cloning a GitHub repository

- Download (clone) the repository: `git clone https://github.com/eggnogdb/eggnog-mapper.git`
- Enter the repository directory and install the dependencies, either with:
    - setuptools: `python setup.py install`
    - pip: `pip install -r requirements.txt`
    - conda: `conda install --file requirements.txt`

Setup

If you want to be sure that eggNOG-mapper is using the bundled binaries for external tools (prodigal, hmmer, diamond, mmseqs), it may help adding the *emapper* scripts and binaries to your environment or PATH variable. If, for example, your eggnog-mapper path was `/home/user/eggnog-mapper`:

```
export
PATH=/home/user/eggnog-mapper:/home/user/eggnog-mapper/eggnogmapper/bin:"$PATH"
```

Also, if you want to store eggNOG-mapper databases in a specific directory, you may wish to create an environment variable to avoid using `--data_dir` in all your commands. For example:

```
export EGGNOG_DATA_DIR=/home/user/eggnog-mapper-data
```

Next step would be downloading the eggNOG-mapper databases, running the next script:

```
download_eggnog_data.py
```

This will download the eggNOG annotation database (along with the taxa databases), and the database of eggNOG proteins for Diamond searches.
If no `EGGNOG_DATA_DIR` variable was defined and no `--data_dir` option was given to `download_eggnog_data.py`, the latter will try to download the files to the `data/` directory within your eggnog-mapper directory.
Note that the Diamond DB is optional. You may decide to perform instead HMMER or MMseqs2 searches, or you may wish to create a Diamond or MMseqs2 DB specific of a taxonomic range. To do so, check `download_eggnog_data.py --help` (for downloading HMMER databases or whole Diamond or MMseqs2 databases), and check `create_dbs.py --help` (to create taxa-specific Diamond/MMseqs2 databases). Also, you may wish to download the PFAM database to be able to run PFAM realignemnts with `--pfam_realign realign` or `--pfam_realign denovo`.
For example, some options of `download_eggnog_data.py` are:

- The `-P` flag is required to download the PFAM database.
- The `-M` flag is required to download the MMseqs2 database. The whole MMseqs2 database includes eggNOG proteins which do not belong to any eggNOG Orthologous Group (OG), whereas the Diamond database only includes those which belong to an OG. Also, note that no MMseqs2 index is provided. To create it, you could use the `mmseqs createindex "$EGGNOG_DATA_DIR"/mmseqs tmp` command (see https://mmseqs.com/latest/userguide.pdf for more details).
- The `-H -d taxID` flag is required to download a HMMER database for a given *taxID* (check list of databases at http://eggnog5.embl.de/#/app/downloads).

Similarly, use `create_dbs.py`. For example, to create a diamond database for Bacteria only:

```
create_dbs.py -m diamond --dbname bacteria --taxa Bacteria
```

This will create a `bacteria.dmnd` diamond database in the default data directory or the one specified in `EGGNOG_DATA_DIR` environment variable. Such database can be used with `emapper.py --dmnd_db bacteria.dmnd`. Note that the first time `create_dbs.py` is used it will take time to download the eggNOG proteins and create the Diamond or MMseqs2 database. Next calls to `create_dbs.py` (to the same data directory pointed by `EGGNOG_DATA_DIR`, or `--data_dir`, or `data/` by default) will not need to download the eggnog5 proteins again. If no more databases are going to be created, the proteins can be removed. For further info, check `create_dbs.py --help`.

Optional tools

Depending on the workflow being used with eggNOG-mapper you will need different external tools. Nonetheless, all of them are actually included, bundled, along with eggNOG-mapper. If you are running eggNOG-mapper fine, you may not need to install anything else.

However, the bundled tools are compiled binaries and could cause trouble in some systems, or could not be the most optimized compiled binaries for your system. In such cases, you may wish to install some or all of these tools independently. The tools are:

- Prodigal: required if using `--itype genome` or `--itype metagenome` along with the option `--genepred prodigal`. Current bundled version is V2.6.3: February, 2016.
- Diamond: required to run the search steps with `-m diamond`. Current bundled version is 2.0.11.
- MMseqs2: required to run the search steps with `-m mmseqs`. Current bundled version is 113e3212c137d026e297c7540e1fcd039f6812b1.
- HMMER: required to run the search steps with `-m hmmer`, to run the HMMER based scripts (`hmm_mapper.py`, `hmm_server.py`, `hmm_worker.py`), and to perform realignments to PFAM with `--pfam_realign realign` or `--pfam_realign denovo`. Current bundled version is HMMER 3.1b2 (February 2015).

Basically, whether eggNOG-mapper uses the one you installed or the bundled one will depend on what tool is found in your environment (PATH) first. If none are found in the environment, eggNOG-mapper will try to use the bundled ones.

Basic usage

To start an annotation job, provide a FASTA file containing your query sequences (`-i` option), specify a project name which will be used as a prefix for all the output files (`-o` option), and run *emapper.py*

```
emapper.py -i FASTA_FILE_PROTEINS -o test
```

This basic example will run a `diamond blastp` search, and for those queries with hits to eggNOG proteins, will carry out functional annotation.

A few recipes

- Run search and annotation, using Diamond in blastx mode

```
emapper.py -m diamond --itype CDS -i FASTA_FILE_NTS -o test
```

- Run search and annotation, using MMseqs after translating input CDS to proteins. Add the search and annotation results to the attributes of an existing GFF file (*GFF decoration*), using the *GeneID* field to link features from the GFF to the annotation results.

```
emapper.py -m mmseqs --itype CDS --translate -i FASTA_FILE_CDS -o test \
--decorate_gff MY_GFF_FILE --decorate_gff_ID_field GeneID
```

- Run search and annotation for assembled contigs, using MMseqs2 "blastx" hits for gene prediction

```
emapper.py -m mmseqs --itype metagenome -i FASTA_FILE_NTS -o test
```

- Run search and annotation for a genome, using Diamond search on proteins predicted by Prodigal, changing the output directory.

```
emapper.py -m diamond --itype genome --genepred prodigal \
-i FASTA_FILE_NTS -o test --output_dir /home/me/mydir
```

- Run gene prediction using a genome to train Prodigal

```
emapper.py -m mmseqs --itype genome --genepred prodigal -i FASTA_FILE_NTS -o test \
--training_genome FASTA_FILE --training_file OUT_TRAIN_FILE
```

- Perform a 2-step (search + annotation) run, using Diamond in *more-sensitive* mode and loading the annotation DB into memory (--dbmem; requires ~44 GB free mem)

```
emapper.py -m diamond --sensmode more-sensitive --no_annot -i FASTA_FILE_PROTS -o
test

emapper.py -m no_search --annotate_hits_table test.emapper.seed_orthologs -o
test_annot_1 --dbmem
```

- Repeat the annotation step, using specific taxa as target and reporting the one-to-one orthologs found

```
emapper.py -m no_search --annotate_hits_table test.emapper.seed_orthologs -o
test_annot_2 --dbmem \
--report_orthologs --target_orthologs one2one --target_taxa 72274,1123487
```

- Use HMMER to search a database of bacterial proteins, using current directory for temporary files, and using a "scratch" directory to write output on a different hard drive than the one used to read. Once emapper.py finishes, output files in the scratch dir will be moved to the actual output dir, and the scratch dir will be removed.

```
emapper.py -m hmmer -d bact -i FASTA_FILE_PROTS -o test --scratch_dir /scratch/test
--temp_dir .
```

- Perform a Diamond search, and annotation also realigning queries to the PFAM domains found on the Orthologous Groups

```
emapper.py -i FASTA_FILE_PROTS -o test --pfam_realign realign
```

- Perform a Diamond search, and annotation also realigning queries to the whole PFAM database

```
emapper.py -i FASTA_FILE_PROTS -o test --pfam_realign denovo
```

- Perform a Diamond search and annotation, constraining the Orthologous Groups from which to retrieve annotations to the *Bacteria* taxon:

```
emapper.py -i FASTA_FILE_PROTS -o test --tax_scope Bacteria
```

- Perform a Diamond search and annotation, constraining the Orthologous Groups from which to retrieve annotations to a predefined tax scope (defined in the file `eggnogmapper/annotation/tax_scopes/bacteria`) for Bacteria and its descendants. The difference with the previous example is that here the *bacteria* file used as tax scope contains a list of taxa instead of only *Bacteria*. If several OGs intersect with the scope, for a given protein, the narrowest will be used to transfer annotations to the query:

`emapper.py -i FASTA_FILE_PROTS -o test --tax_scope bacteria`

- Perform a Diamond search and annotation, constraining the Orthologous Groups from which to retrieve annotations to the *Gammaproteobacteria* tax scope. Also, from those OGs, force to retrieve annotations at the *Bacteria* level:

```
emapper.py -i FASTA_FILE_PROTS -o test --tax_scope Gammaproteobacteria
--tax_scope_mode Bacteria
```

Parameters emapper.py

General Options

- `--version`

show version and exit.

- `--list_taxa`

List available taxonomic names and IDs for OGs and exit. These are valid tax names and IDs for `--tax_scope/--tax_scope_mode`

Execution Options

- `--cpu NUM_CPU`

number of CPUs to be used whenever possible (diamond, annotation tasks, etc). `--cpu 0` to run with all available CPUs. Default is 1.

- `--resume`

resumes a previous emapper run, skipping results in existing output files.

- `--override`

overwrites output files if they exist. By default, execution is aborted if conflicting files are detected.

- `--mp_start_method [fork,spawn,forkserver]`

used to control the Python multiprocessing start method. Use it only if multiprocessing is not running correctly in your OS. Default: spawn. *(since version 2.1.6)*

Input Data Options

- `-i FILE`

input FASTA file containing query sequences (proteins by default; see `--itype` and `--translate`). Required unless `-m no_search`.

- `--itype INPUT_TYPE`

Type of sequences in the input (`-i`) file. Default is `--itype proteins`. All *INPUT_TYPE* options are shown the next table:

INPUT_TYPE	**description**
proteins	Input sequences are used directly for the search step.
CDS	Input sequences are used directly for Diamond and MMseqs2, which will run in "blastx" mode and will report the best hit, unless `--translate` is used, in which case input sequences will be translated to proteins and searched "blastp" mode. For HMMER, the input CDS are always translated to proteins (`--translate` is automatically activated).
genome	Input sequences are considered chromosomes/contigs, and gene prediction will be performed (see `--genepred`).
metagenome	The same as *genome*, except that when `--genepred prodigal` Prodigal will be run in a different mode.

- `--translate`

When `--itype CDS --translate`, input sequences will be translated to proteins before search. If `-m hmmer` and `--itype CDS`, input sequences will be translated to proteins, as if `--translate` was automatically activated. If `-m diamond` or `-m mmseqs` and `--itype CDS` but no `--translate` option is given, searches will be performed in "blastx" mode, and the best hit will be annotated. Note that this is different than using `--itype genome` or `--itype metagenome`, where more than one hit per query could be annotated.

- `--annotate_hits_table FILE`

annotate TSV formatted table with at least 4 fields: query, hit, evalue, score. Usually, FILE is a *emapper.seed_orthologs* file from a previous eggNOG-mapper run. Required if `-m no_search`.

- `-c FILE`, `--cache FILE`

Annotations file with md5 checksums of sequences. Required if `-m cache`.

- `--data_dir DIR`

Specify a path to the eggNOG-mapper databases. By default, `data/` folder or the one specified by the `EGGNOG_DATA_DIR` environment variable. The annotation DBs must be always present in this directory, whereas HMMER, Diamond and MMseqs2 DBs could be in a different directory when using `-d/--database`, `--dmnd_db` and `--mmseqs_db`, respectively.

Gene Prediction Options

- `--genepred search|prodigal`

When `--itype genome` or `--itype metagenome` is used, gene prediction is carried out. There are 2 gene prediction modes:

  - The default is `--genepred search`, which means that either Diamond or MMseqs2 (depending on `-m` argument) is run in *blastx* mode. As of now, we cannot recommend using Diamond for complete genomes, unless the assembly is rather fragmented and/or contigs are not very large. MMseqs2 is faster than Diamond for assembled genomes, and it is the recommended one for large contigs.
  - If `--genepred prodigal` is specified, Prodigal is run for gene prediction, and the proteins predicted by Prodigal are used in the subsequent search and annotation steps. Prodigal will be run in a different mode depending whether `--itype genome` or `--itype metagenome` is used.

- `--trans_table TRANS_TABLE_CODE`

Option to change the translation table used for CDS translation and gene prediction. It corresponds with Diamond's `--query-gencode`, MMseqs2's `--translation-table` and Prodigal's `-g/--trans-table`). Usually the value is an integer corresponding to a specific translation table (e.g. https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi). Check each program's documentation for more info. It is also applied when `--translate` is used, both for input and output sequences.

- `--training_genome FASTA_FILE`

FASTA file of the genome to be used for Prodigal's training mode. Requires `--itype genome --genepred prodigal` and also requires `--training_file FILE`. Note training will be run only if the training file does NOT exist. If the training file already exists, the latter will be used directly for gene prediction, and training will be skipped.

- `--training_file FILE`

Training file to be created and/or used by Prodigal. If the training file does not exist, the training genome (`--training_genome` option) will be used to create a training file, and then immediately perform gene prediction from such training file. If the training file already exists, the training is skipped, the `--training_genome` option is ignored, and gene prediction is performed using the existing training file.

- `--allow_overlaps none|strand|diff_frame|all`

When `--genepred search --itype genome/metagenome`, this option controls whether overlapping blastx hits are annotated or not. By default, `--allow_overlaps none` best hit is annotated among overlapping hits. If `--allow_overlaps strand`, best non-overlapping hit from each strand is annotated. If `--allow_overlaps diff_frame`, best non-overlapping hit from each frame in each strand is annotated. If `--allow_overlaps all`, all hits are annotated.

- `--overlap_tol FLOAT`

Overlap tolerance (0.0 - 1.0). Two hits are considered to overlap if (overlap_size / hit length) > overlap_tol for at least one of the hits. By default, 0.0, any overlap is considered as such. In contrast, `--overlap_tol 1.0` means that one of the hits must overlap entirely to consider that hits do overlap to each other.

Search Options

- `-m MODE`

how input queries will be searched against eggNOG sequences. Default is `-m diamond`. All *MODE* options are shown in the next table:

MODE	**description**	**Notes**
diamond	search queries against eggNOG sequences using Diamond.	Requires `-i FILE`.
hmmer	search sequences/hmm against sequences/hmm using HMMER.	Requires `-i FILE` and `-d DB_NAME`.
mmseqs	search queries against eggNOG sequences using MMseqs2.	Requires `-i FILE`.
cache	annotate queries using a previously annotated file (`-c`) which includes md5 hashes of the annotated sequences.	Requires `-i FILE` and `-c FILE`.
no_search	skip search stage. Annotate an existing *emapper.seed_orthologs* file.	Requires `--annotate_hits_table FILE`, unless `--no_annot` is used.

Search filtering common options

- `--pident FLOAT`

report only alignments equal or above this percentage of identity threshold (0.0 - 100.0). Default *None*. No effect if `-m hmmer`.

- `--evalue FLOAT`

report only alignments equal or above this e-value threshold. Default *0.001*.

- `--score FLOAT`

report only alignments equal or above this bit score threshold. Default *None*.

- `--query_cover FLOAT`

report only alignments equal or above this query coverage fraction threshold (0.0 - 100.0). Default *None*. No effect if `-m hmmer`.

- `--subject_cover FLOAT`

report only alignments equal or above this target (eggNOG sequence) coverage fraction threshold (0.0 - 100.0). Default *None*. No effect if `-m hmmer`.

Diamond search options

- `--dmnd_algo [auto, 0, 1, ctg]`

Diamond's *--algo* option. Use `--dmnd_algo ctg` to perform a faster search when the input set of sequences is small. Default is Diamond's default. *(since version 2.1.6)*.

- `--dmnd_db FILE`

path to diamond-compatible database. Useful to specify a location different than `data/`, `EGGNOG_DATA_DIR` or `--data_dir`.

- `--sensmode DIAMOND_SENS_MODE`

either *default*, *fast*, *mid-sensitive*, *sensitive*, *more-sensitive*, *very-sensitive* or *ultra-sensitive*. Default is *sensitive* (to be sure, check the default for your version in `emapper.py --help`).

- `--dmnd_iterate [yes, no]`

Activates/disables the `--iterate` option of Diamond for iterative searches, from faster, less sensitive modes, up to the sensitivity specified with `--sensmode`. It requires using Diamond version >= 2.0.11. In version 2.1.5 use `--dmnd_iterate` to activate it. In version 2.1.6 it is activated by default; use `--dmnd_iterate no` to disable it.

- `--matrix MATRIX_NAME`

which substitution matrix to be used by diamond, among BLOSUM62,BLOSUM90,BLOSUM80,BLOSUM50,BLOSUM45,PAM250,PAM70,PAM30 .

- `--dmnd_frameshift INT`

controls Diamond's `--frameshift/-F` option. Default is Diamond's default (disabled).

- `--gapopen INT`

gap open penalty used by diamond. Default is Diamond's default.

- `--gapextend INT`

gap extend penalty used by diamond. Default is Diamond's default.

- `--block_size FLOAT`

Diamond's *-b/--block-size* option. Default is Diamond's default.

- `--index_chunks INT`

Diamond's *-c/--index-chunks* option. Default is Diamond's default.

- `--outfmt_short`

Diamond will produce only the query, subject, evalue and score fields in its output, and seed_orthologs file will have only those fields also. This option could be useful to obtain better performance when no thresholds for pident, and query and subject coverages are used (for more info, see Diamond docs about traceback).

- `--dmnd_ignore_warnings`

Diamond's *--ignore-warnings* option. Use it to avoid Diamond stopping execution due to warnings (for example, when queries include proteins with ATGC aminoacids only. *(since version 2.1.6)*.

MMseqs2 search options

- `--mmseqs_db FILE`

path to MMseqs2-compatible database. Useful to specify a location different than *data/*, *EGGNOG_DATA_DIR*, or *--data_dir*.

- `--start_sens FLOAT`

Starting sensitivity for MMseqs2 iterative searches. Default 3.

- `--sens_steps INT`

Number of iterative searches with different sensitivities for MMseqs2. Default 3.

- `--final_sens FLOAT`

Final sensitivity for MMseqs2 iterative searches. Default 7.

- `--mmseqs_sub_mat MMSEQS_SUB_MAT`

Matrix to be used for *--sub-mat* option of MMseqs2. Default: the default one used by MMseqs2.

HMMER search options

- `-d DB_NAME`, `--database DB_NAME`

specify the target database for sequence searches. DB_NAME should be the name of a database downloaded using `download_eggnog_data.py -H -d taxID`, or such a database loaded in a server (e.g. `db.hmm:host:port`; see `hmm_server.py --help`)

- `--servers_list FILE`

A FILE with a list of remote hmmpgmd servers. Each row in the file represents a server, in the format *host:port*. If `--servers_list` is specified, host and port from `-d` option will be ignored.

- `--qtype QUERY_TYPE`

hmm or seq. Type of input data (`-i`).

- `--dbtype DB_TYPE`

hmmdb or seqdb. Type of data in DB (`-db`).

- `--usemem`

Use this option to allocate the whole HMMER database in memory. If `--dbtype hmmdb`, the database must be a *hmmpress*-ed database. If `--dbtype seqdb`, the database must be a HMMER-format database created with *esl-reformat*. Database will be unloaded after execution. Note that `--usemem` is different than `--dbmem`, which is used to load into memory the annotation DB.

- `-p INT`, `--port INT`

Port used to setup HMM server, when `--usemem`. Also used for `--pfam_realign` modes.

- `--end_port PORT`

Last port to be used to setup HMM server, when `--usemem`. Also used for `--pfam_realign` modes.

- `--num_servers INT`

When using `--usemem`, specify the number of servers to fire up. By default, only 1 server is used. Note that cpus specified with `--cpu` will be distributed among servers and workers. Also used for `--pfam_realign` modes. It is important to consider that, for each server instance, the HMM database will be loaded into memory, and therefore memory consumption will grow as `--num_servers` increases.

- `--num_workers INT`

When using `--usemem`, specify the number of workers per server to fire up. By default, cpus specified with `--cpu` will be distributed among servers and workers. Also used for `--pfam_realign` modes. In our tests `--num_workers` has not the expected impact on performance, and increasing `--num_servers` is required to get an actual speed boost, although the memory requirements must be met. However, this could be different in other systems, compilations, or versions of *hmmpgmd*.

- `--hmm_maxhits INT`

Max number of hits to report (0 to report all). Default=1.

- `--report_no_hits`

Whether queries without hits should be included in the output table.

- `--hmm_maxseqlen INT`

Ignore query sequences larger than this length. Default=5000"

- `--Z FLOAT`

Fixed database size used in *phmmer*/*hmmscan* allows comparing e-values among databases. Default=40,000,000

- `--cut_ga`

Adds the *--cut_ga* to HMMER commands (useful for Pfam mappings, for example). See HMMER documentation for more info.

- `--clean_overlaps CLEAN_OVERLAPS_MODE`

Removes hits which overlap, keeping only the one with best e-value. Default *none*. Use the *all* and *clans* options when performing a *hmmscan* type search (i.e. domains are in the database). Use the *hmmsearch_all* and *hmmsearch_clans* options when using a *hmmsearch* type search (i.e. domains are the queries). The *clans* and *hmmsearch_clans* options will only have effect for hits to/from PFAM.'

Annotation Options

- `--no_annot`

perform only the search stage and skip functional annotation, reporting only seed orthologs (*emapper.seed_orthologs* file).

- `--dbmem`

Store the whole eggNOG sqlite DB into memory before retrieving the annotations. This requires ~44 GB of RAM memory available, but can increase annotation speed considerably. Database will be unloaded after execution.

- `--seed_ortholog_evalue FLOAT`

min e-value expected when searching for seed eggNOG ortholog. Queries not having a significant seed orthologs will not be annotated. Default=0.001

- `--seed_ortholog_score FLOAT`

min bit score expected when searching for seed eggNOG ortholog. Queries not having a significant seed orthologs will not be annotated. Default=60

- `--tax_scope FILE|PREDEFINED_FILENAME|LIST_OF_TAXA|none`

fix the taxonomic scope used for annotation, so only speciation events from a particular clade are used for functional transfer. More specifically, the `--tax_scope` list is intersected with the seed orthologs clades (i.e. the taxa from seed orthologs' OGs), and the resulting clades are used for annotation based on `--tax_scope_mode`. Note that those seed orthologs without clades intersecting with `--tax_scope` will be filtered out, and won't be annotated. Possible arguments for `--tax_scope` are shown in the following table. Default is *auto*, which is a *PREDEFINED_FILENAME*.

tax_scope	**description**
FILE	A path to a file which contains a list of tax IDs and/or tax names.
PREDEFINED_FILENAME	The filename of a file stored in *eggnogmapper/annotation/tax_scopes/*, and which contains a list of tax IDs and/or tax names. Available ones are: *auto* (synonym *all*), *auto_broad* (synonym *all_broad*), *all_narrow*, *archaea*, *bacteria*, *bacteria_broad*, *eukaryota*, *eukaryota_broad*, and *prokaryota_broad*.
LIST_OF_TAXA	A comma-separated list of tax IDs and/or tax names. For example, `--tax_scope 2759,2157,2,1` for Eukaryota, Archaea, Bacteria and root.
none	Do not filter out annotations based on taxonomic scope.

- `--tax_scope_mode broadest|inner_broadest|inner_narrowest|narrowest|FILE|PREDEFINED_FILENAME|LIST_OF_TAXA`

A second layer to control taxonomic scope from which to retrieve annotations. If `--tax_scope` acts as a filter, `--tax_scope_mode` determines which taxonomic level (i.e. which Orthologous Group) from the seed ortholog is used to retrieve annotations. Default is *inner_narrowest*. Possible arguments are:

tax_scope_mode	**description**

broadest	Use the OG with the broadest clade.
inner_broadest	From the OG which intersect `--tax_scope`, use the one with the broadest clade.
inner_narrowest	From the OG which intersect `--tax_scope`, use the one with the narrowest clade.
narrowest	Use the OG with the narrowest clade.
FILE, *PREDEFINED_FILENAME*, *LIST_OF_TAXA*	The same values accepted by `--tax_scope`, except *none*. If this option is used, the taxa from OGs is intersected first with `--tax_scope`, to filter out seed orthologs outside the taxonomic range. Secondly, taxa from OGs are intersected with `--tax_scope_mode`. From the remaining OGs, the one with the narrowest clade is used (as in *inner_narrowest*). This is useful when we want to filter out queries using a different taxonomic scope than the one to use for annotation. For example, we want to annotate only queries within the *Gammaproteobacteria* range, but use the *Bacteria* level OGs for annotation of those queries: `--tax_scope Gammaproteobacteria --tax_scope_mode Bacteria` should work.

- `--target_orthologs one2one|many2one|one2many|many2many|all`

defines what type of orthologs (in relation to the seed ortholog) should be used for functional transfer. Default: *all*

- `--target_taxa all|LIST_OF_TAX_IDS`

Comma-separated list of taxa to be used for annotation. Note that this option interacts with the Ortholgous Group chosen due to the *--tax_scope/--tax_scope_mode* options. First, speciation events are identified among the OGs based on *--tax_scope/--tax_scope_mode*. Then, annotation will be transferred from the orthologs found within those speciation events: from all the orthologs if *--target_taxa all* (default), or only from the taxa from *LIST_OF_TAX_IDS*, when `--target_taxa LIST_OF_TAX_IDS`.

- `--excluded_taxa none|LIST_OF_TAX_IDS`

the opposite behaviour than `--target_taxa`. Default is *none* (no species are excluded for annotation).

- `--report_orthologs`

as a first step in functional annotation, eggNOG-mapper identifies the orthologs of each query, using seed orthologs from the search stage as an anchoring or starting point. A list of these orthologs is not reported by default. Use this option get the list of orthologs found for each query ('emapper.orthologs' file).

- `--go_evidence experimental|non-electronic|all`

defines what type of GO terms should be used for annotation. *experimental* = Use only terms inferred from experimental evidence. *non-electronic* (default) = Use only non-electronically curated terms. *all* = all GO terms will be retrieved.

- `--pfam_realign none|realign|denovo`

Defines how PFAM annotation will be performed.

- *none*

A list of PFAMs, directly transferred from orthologs, will be reported.

- *realign*

PFAMs from orthologs will be realigned to the query, and a list of PFAMs and their positions on the query will be reported.

- *denovo*

Each query will be realigned to the whole PFAM DB, and a list of PFAMs and their positions on the query will be reported.

- `--md5`

Adds a column with the md5 hash of the query sequences in the annotations output file. An annotations output file created this way can be used as cache file (`-c CACHE_FILE`) for the `-m cache` mode.

Output options

- `--output,-o FILE_PREFIX`

base name for output files

- `--output_dir DIR`

where output files should be written. default is current working directory.

- `--scratch_dir DIR`

write output files in a temporary scratch dir, move them to the final output dir when finished. Speed up large computations using network file systems.

- `--temp_dir DIR`

where temporary files are created. Better if this is a local disk.

- `--no_file_comments`

no header lines nor stats are included in the output files

- `--decorate_gff no|yes|FILE`

Option to create/decorate a GFF file with emapper hits and/or annotations. Default is *no*.

- *no*: no GFF decoration will be performed. If running gene prediction with Prodigal, its GFF will be among the output files anyway. If running blastx-based gene prediction (`--genepred search`), the GFF with CDS of hits will be among output files anyway.
- *yes*: a new GFF will be created, and decorated with hits and/or annotations. Since emapper version 2.1.7 this is a different GFF file than the one created during gene prediction.
- *FILE*: a new GFF will be created, adding hits and/or annotations to the attributes already existing in the specified FILE. `--decorate_gff_ID_field` can be used to specify a GFF attribute (see `--decorate_gff_ID_field`). Since emapper version 2.1.7 this is a different GFF file than the one created during gene prediction.
- `--decorate_gff_ID_field ID`

ID or name of the GFF attribute to be used to link GFF features and annotation results, when using `--decorate_gff FILE`. Default is *ID*.

- `--excel`

Annotations will be output also in Excel (.xlsx) format.

Output format

Output files

- Search hits (*prefix*.emapper.hits)

A file with the results from the search phase, from HMMER, Diamond or MMseqs2.

- Seed orthologs (*prefix*.emapper.seed_orthologs)

A file with the results from parsing the hits. Each row links a query with a seed ortholog. This file has the same format independently of which searcher was used, except that it can be in short format (4 fields), or full.

- Annotations (*prefix*.emapper.annotations)

A file with the results from the annotation phase. Therefore, each row represents the annotation reported for a given query.

- Excel (*prefix*.emapper.annotations.xlsx)

Annotations in .xlsx format. Created only when using the `--excel` option.

- Orthologs (*prefix*.emapper.orthologs)

A file with the list of orthologs found for each query. This file is created only if using the `--report_orthologs` option.

- Sequences of predicted CDS (*prefix*.emapper.genepred.fasta)

A FASTA file with the sequences of the predicted CDS. It is generated when gene prediction is carried out, with `--itype genome` or `--itype metagenome`.

- GFF of predicted CDS (*prefix*.emapper.gff) (*only in versions 2.1.5 and 2.1.6*)

A GFF (version 3) file, with the search and/or annotation results. See `--decorate_gff`.

- GFF of predicted CDS (*prefix*.emapper.genepred.gff) (*only in version 2.1.7*)

A GFF (version 3) file, with the gene prediction results from Prodigal or from blastx-like searches.

- GFF decorated with hits and/or annotations (*prefix*.emapper.decorated.gff) (*only in version 2.1.7*)

A GFF (version 3) file, decorated with hits and/or annotations added as attributes. See `--decorate_gff`.

- Sequences without annotation (*prefix*.emapper.no_annotations.fasta)

A FASTA file with the sequences of queries for which an existing annotation was not found using the *-m cache* mode. This file can be used as input of another eggNOG-mapper run without using the cache, trying to annotate the sequences.

- PFAM hits (*prefix*.emapper.pfam)

A file with the positions of the PFAM domains identified. Only created if `--pfam_realign realign` or `--pfam_realign denovo`.

Output fields

All files contain rows with tab-separated columns or fields.

Seed orthologs file

- *query*
- *target*

The *target* is what is also known, in eggnog-mapper, as 'seed ortholog'. It is the eggNOG sequence representing the best hit found for a given query during the search phase, and it will be used, during the annotation phase, to retrieve orthologs from which to transfer annotations.

- *e-value*
- *bit-score*

The *e-value* and *bit-score* fields are the values returned by the search tool being used (*diamond* by default, see *-m* option).

- *pident*

Percentage of identity between the query and the subject.

- *qstart*

First position of query in the alignment.

- *qend*

End position of query in the alignment.

- *sstart*

Start position of subject (a.k.a. target) in the alignment.

- *send*

End position of subject (a.k.a. target) in the alignment.

- *qcov*

Percentage of the query length which is part of the alignment.

- *scov*

Percentage of the subject (a.k.a. target) length which is part of the alignment.
The `--outfmt_short` option can be used to output only the first 4 fields of the seed orthologs file, when running searches with Diamond (see `--outfmt_short` option above).

Annotations file

Search hit fields

- query_name
- seed_eggNOG_ortholog
- seed_ortholog_evalue
- seed_ortholog_score

Orthologous Groups fields

- eggNOG_OGs

a comma-separated, clade depth-sorted (broadest to narrowest), list of Orthologous Groups (OGs) identified for this query. Note that each OG is represented in the following format: *OG@tax_id|tax_name*

- max_annot_lvl

tax_id|tax_name of the level of widest OG used to retrieve orthologs for annotations.

- COG_cat

COG category of the narrowest OG with a valid one.

- Description

Description of the narrowest OG with a valid one.

Transferred annotations fields

- Preferred_name
- GOs
- EC
- KEGG_ko
- KEGG_Pathway
- KEGG_Module

- KEGG_Reaction
- KEGG_rclass
- BRITE
- KEGG_TC
- CAZy
- BiGG_Reaction
- PFAMs
- md5 (only when `--md5` is used)

Orthologs file

- query
- orth_type

Type of orthologs in this row. See `--target_orthologs`.

- species
- comma-separated list of orthologs

If an ortholog shows a "*", such ortholog was used to transfer its annotations to the query.

HMMER hits file

- query
- hit
- evalue
- sum_score
- query length
- HMM position "from"
- HMM position "to"
- Sequence position "from"
- Sequence position "to"
- query coverage

Diamond hits file

Depends on `--outfmt_short`. See Diamond documentation for more info.

MMseqs2 hits file

See MMseqs2 documentation for more info.

Sequences of predicted CDS

If gene prediction is performed using search hits (diamond or mmseqs "blastx" hits), sequence identifiers include the identifier of the original sequence from which the CDS has been found, followed by an underscore and a number to differentiate among CDS from the same original sequence (e.g. A CDS found in >query_seq will be named

>query_seq_1. A second one will be >query_seq_2, ...). If gene prediction is performed using prodigal, this output file is the one generated by Prodigal (check Prodigal documentation for output formats).

GFF

All eggNOG-mapper attributes will be prefixed with "em_". If gene prediction is performed using Prodigal and `--decorate_gff no` is used, this output file is the one generated by Prodigal (check Prodigal documentation for more info).

Sequences without annotation

Just a FASTA file with the same identifiers as the original sequences.

PFAM hits

- query_name
- hit (i.e. PFAM domain)
- evalue
- sum_score
- query_length
- hmmfrom
- hmmto
- seqfrom
- seqto
- query_coverage

Setting up large annotation jobs

The following recommendations are based on the different experiences annotating huge genomic and metagenomic datesets (>100M proteins).
eggNOG mapper works at two phases: 1) finding seed orthologous sequences 2) expanding annotations. 1 is mainly cpu intensive, while 2 is more about disk operations. You can therefore optimize the annotation of huge files, but running each phase on different setups.

Phase 1. Homology searches

1) Split your input FASTA file into chunks, each containing a moderate number of sequences (1M seqs per file worked good in our tests). We usually work with FASTA files where sequences are in a single line, so splitting is very simple.

```
split -l 2000000 -a 3 -d input_file.faa input_file.chunk_
```

2) Use diamond mode. Each chunk can be processed independently in a cluster node, and you should tell `emapper.py` not to run the annotation phase yet. This way you can parallelize diamond searches as much as you want, even when running from a shared file system. Assuming an example with 100M proteins, the above command will generate 100 file chunks, and each should run diamond using 16 cores. The necessary commands that need to be submitted to the cluster queue can be generated with something like this:

```
# generate all the commands that should be distributed in the cluster
for f in *.chunk_*; do
echo ./emapper.py -m diamond --no_annot --no_file_comments --cpu 16 -i $f -o $f;
done
```

Phase 2. Orthology and functional annotation

The annotation phase needs to query `data/eggnog.db` intensively. This file is a sqlite3 database, so it is highly recommended that the file lives under the fastest local disk possible. For instance, we store `eggnog.db` in SSD disks or, if possible, under `/dev/shm` (memory based filesystem).

3) Concatenate all chunk_*.emapper.seed_orthologs file.

```
cat *.chunk_*.emapper.seed_orthologs > input_file.emapper.seed_orthologs
```

4) Run the orthologs search and annotation phase in a single multi core machine (10 cores in our example), reading from a fast disk.

```
emapper.py --annotate_hits_table input.emapper.seed_orthologs --no_file_comments -o
output_file --cpu 10
```

We usually annotate at a rate of 300-400 proteins per second using a 10 cpu cores and having `eggnog.db` under the `/dev/shm` disk, but you can of course run many of those instances in parallel. If you are running `emapper.py` from a conda environment, check [these](https://github.com/jhcepas/eggnog-mapper/issues/80).
and _voilà_, you got your annotations.

Even larger jobs in large memory computers

Use MMseqs for the search step. If possible, create the MMseqs2 index first.

```
# generate all the commands that should be distributed in the cluster
for f in *.chunk_*; do
echo ./emapper.py -m mmseqs --no_annot --no_file_comments --cpu 16 -i $f -o $f;
done
```

Load the annotation database into memory for the annotation step (~44 GB mem required)

```
emapper.py --annotate_hits_table input.emapper.seed_orthologs --no_file_comments -o
output_file --cpu 10 --dbmem
```

Also when running Diamond for the search step (`-m diamond`) it can benefit from using large memory computers, by tuning the `--block_size` and the `--index_chunks` options. Also `--index_chunks` could be required by diamond when running in computers with over 64GB RAM.

Exporting as .csv

Sometimes all that you need is a table that lists the orthogroup assignments of all proteins in the database. While this table is not provided by default, there is a way to extract it:

- install emapper following the instructions here
- run \`download_eggnog_data.py\` and agree to download the main EggNOG database ("eggnog.db")
- install a program that can read .db files (for instance [DB](https://sqlitebrowser.org))
- open said program and open \`eggnog.db\`
- export the \`prots\` table as CSV (in DB Browser: right click > export as CSV file)

This should result in an approx. 10Gb csv file that contains all relevant information per protein in the database.

Citation

Please cite the following two papers if you use eggNOG-mapper v2

```
[1] eggNOG-mapper v2: functional annotation, orthology assignments, and domain
      prediction at the metagenomic scale. Carlos P. Cantalapiedra,
      Ana Hernandez-Plaza, Ivica Letunic, Peer Bork, Jaime Huerta-Cepas.
      biorxiv (2021). doi: https://doi.org/10.1101/2021.06.03.446934

[2] eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated
      orthology resource based on 5090 organisms and 2502 viruses. Jaime
      Huerta-Cepas, Damian Szklarczyk, Davide Heller, Ana Hernández-Plaza, Sofia
      K Forslund, Helen Cook, Daniel R Mende, Ivica Letunic, Thomas Rattei, Lars
      J Jensen, Christian von Mering, Peer Bork Nucleic Acids Res. 2019 Jan 8;
      47(Database issue): D309–D314. doi: 10.1093/nar/gky1085
```