COGtools documentation

Petra Polakovicova and Karel Sedlar
Python version: 3.10.0

August 25, 2022

1 Introduction

COGtools is a package for improving the functional annotation of bacterial genomes, classification of protein-coding sequences into clusters of orthologous groups (COGs), and visualization of the final annotated genome if a complete genome is available. The package uses the outputs of the tools that assign COGs to protein-coding sequences, namely eggNOG-mapper [1], Operon-mapper [2], and Batch CD-Search [3]. The COGtools includesfunctions to process these outputs and improve the annotation. It outputs a new processed file in a suitable format that is ready to be visualized in DNAPlotter program [4]. In the case of draft genomes, a text file is generated with assigned cogs and their categories. When annotating multiple genomes, the COGtools offers the function to compare the relative abundance of categories in individual bacteria with barplots.

The best results can be achieved by combining all three resources, but COGtools can use a single resource to process the data and adjust the assigned functional categories according to the latest COG database.

The tools for a genome annotation are available at:
eggNOG-mapper: http://eggnog-mapper.embl.de/
Operon-mapper: https://biocomputo.ibt.unam.mx/operon_mapper/
Batch CD-Search:https://www.ncbi.nlm.nih.gov/Structure/bwrpsb/bwrpsb.cgi

2 Installation

- the COGtools is freely available at https://github.com/xpolak37/COGtools

The following packages are required to run COGtools properly:
'pandas', 'Biopython', 'seaborn', 'pillow'

3 Usage

To use this tool you need the following files:

1. If you have a complete genome:
   - Bacterial genome in FASTA format
   - Protein-coding sequences in FASTA format
   - Features in GFF3 format
   - Annotation by eggNOG-mapper (decorated.gff)
   - Annotation by Operon-mapper (ORF_coordinates.txt and predicted_COGs.txt)
   - Annotation by Batch CD-Search (hitdata.txt)
2. If you have a draft genome:
   - Contigs in FASTA format
   - Proteins in FASTA format
   - Features in GFF3 format (optional)
   - Annotation by eggNOG-mapper (decorated.gff)

- Annotation by Operon-mapper (predicted_protein_sequences.txt and predicted_COGs.txt)
- Annotation by Batch CD-Search (hitdata.txt)

The COGtools includes 13 functions, their usage is explained in the next sections. You can also run the whole process via the command line:

```
usage: cogtools.py [-n] [-i] [-o] [-c] [-ch] [-t] [-p] [-d] [-g]
arguments:
  -n,  --name           organism name
  -i,  --inputs         path to the input directory
  -o,  --outputs        path to the output directory
  -c                    neglect other orthologous groups than COGs
  -ch, --choice         select the option to assign a category (0-4)
  -t,  --track          create track template with a legend
  -p                    use palette from COG database
  -d,  --draft          work with the draft genome
  -g,  --gff            gff3 file used in Operon-mapper (with draft genomes)

example for complete genome:
py cogtools.py -n aneurinibacillus_thermoaerophilus
               -i C:/Users/ppola/genomes/aneurinibacilus/inputs
               -o C:/Users/ppola/genomes/aneurinibacilus/outputs -t -c -ch 1

example for draft genome:
py cogtools.py -n pseudomonas_P2653
               -i C:/Users/ppola/genomes/pseudomonas_P2653/inputs
               -o C:/Users/ppola/genomes/pseudomonas_P2653/outputs -ch 1 -d -g
```

If the group belongs to more than one category, you have 5 options to choose:

0. maintain all categories
1. the first assigned category
2. the randomly assigned category
3. the category that is the most abundant in the given genome
4. the category that is the least abundant in the given genome

When using COGtools via command line, please keep in mind that the files have to be named as follows:

1. If you have a complete genome:
   - organism_name.fasta
   - organism_name_cds.txt
   - organism_name.gff3
   - organism_name_eggnog.gff
   - organism_name_orf_operon.txt
   - organism_name_cogs_operon.txt
   - organism_name_batch.txt
2. if you have a draft genome:
   - organism_name_proteins.fsa_aa
   - organism_name_eggnog.gff
   - organism_name_proteins_operon.txt
   - organism_name_cogs_operon.txt
   - organism_name_batch.txt

3.1 em_processor

Processes the output file (decorated.gff) from eggNOG-mapper tool into more structured COGtools-data. This function can be used for complete genomes only. The output of this function is a file in GFF format that contains a suitable header with information about CDSs with assigned COGs by eggNOG-mapper.

```
usage: em_processor(organism_name, em_file, cds_file, cogs_only = False,
                    output_dir=output_path)

arguments:
  organism_name         organism name
  em_file               path to the eggNOG-mapper output file
  cds_file              path to the eggNOG-mapper input file (CDS file)
  cogs_only             neglect other orthologous groups than COGs
  output_dir            path to the output directory (optional)

example: em_processor("aneurinibacilus", "decorated.gff", "cds.fasta", cogs_only=True)
```

3.2 em_processor_draft

Process the output file (decorated.gff) from the eggNOG-mapper tool into more structured COGtools-data. This function is recommended to be used for draft genomes. The output of this function is a file in txt format that contains a suitable header with assigned COGs and their categories.

```
usage: em_processor_draft(organism_name, em_file, cogs_only = False, output_dir=output_path)

arguments:
  organism_name         organism name
  em_file               path to the eggNOG-mapper output file
  cogs_only             neglect other orthologous groups than COGs
  output_dir            path to the output directory (optional)

example: em_processor_draft("aneurinibacilus", "decorated.gff", cogs_only = True)
```

3.3 om_processor

Processes the outputs files (ORF_coordinates.txt and predicted_COGs.txt) from Operon-mapper into more structured COGtools-data. This function can be used for complete genomes only. The output of this function is a file in GFF format that contains a suitable header with information about all predicted features.

```
usage: om_processor(organism_name, orf_file, cog_file, output_dir=output_path)

arguments:
  organism_name         organism name
  orf_file              path to the Operon-mapper outputs file ORFs_coordinates.txt
  cog_file              path to the Operon-mapper outputs file predicted_COGs.txt
  output_dir            path to the output directory (optional)

example: om_processor("aneurinibacilus", "ORFs_coordinates.txt", "predicted_COGs.txt")
```

3.4 om_processor_drat

Process the outputs files (predicted_protein_sequences.txt and predicted_COGs.txt) from Operon-mapper into more structured COGtools-data. The output of this function is a file in txt format that contains a suitable header with assigned COGs and their categories.

```
usage: om_processor_draft(organism_name, proteins, operon_proteins, operon_cogs,
gff_included=True, output_dir=output_path):

arguments:
  organism_name         organism name
```

```
  proteins              path to downloaded proteins
  operon_proteins       path to Operon-mapper's output file
                        predicted_protein_sequences.txt
  operon_cogs           path to Operon-mapper's output file predicted_COGs.txt
  gff_included          a gff file was used in the Operon-mapper*
  output_dir            path to the output directory (optional)

example: om_processor("aneurinibacilus", "proteins.fsa_aa",
                      "predicted_protein_sequences.txt" "predicted_COGs.txt")

*Note: When using the GFF file in Operon-mapper, the predicted proteins match
the downloaded ones. Otherwise, this function works based on the local
alignment of the proteins.
```

3.5 batch_processor

Processes the outputs file (hitdata.txt) from the Batch CD-Search tool into more structured COGtools-data. The output of this function is a file in GFF format that contains a suitable header with information about CDSs with assigned COG by Batch CD-Search.

```
usage: batch_processor(organism_name, batch_file, output_dir=output_path)

arguments:
  organism_name         organism name
  batch_file            path to Batch CD-Search outputs file hitdata.txt
  output_dir            path to the output directory (optional)

example: batch_processor("aneurinibacilus", "hitdata.txt")
```

3.6 batch_processor_draft

Processes the outputs file (hitdata.txt) from the Batch CD-Search tool into more structured COGtools-data. This function is recommended to be used for draft genomes. The output of this function is a file in txt format that contains a suitable header with assigned COGs and their categories.

```
usage: batch_processor_draft(organism_name, batch_file, output_dir=output_path)

arguments:
  organism_name         organism name
  batch_file            path to the Batch CD-Search output file hitdata.txt
  output_dir            path to the output directory (optional)

example: batch_processor_draft("aneurinibacilus", "hitdata.txt")
```

3.7 batch_splitter

A function that splits a CDS file if it contains more than 1000 sequences.

```
usage: batch_splitter(organism_name, gene_file, output_dir=output_path)

arguments:
  organism_name         organism name
  gene_file             path to CDS file
  output_dir            path to the output directory (optional)

example: batch_splitter(aneurinibacilus, "CDS.fasta")
```

3.8 batch_merger

A function that merges annotation files from Batch CD-Search's output.

```
usage: batch_merger(organism_name, files, output_dir=output_path)

arguments:
  organism_name         organism name
  files                 path to the annotation files paths to annotated files in
                        list
  output_dir            path to the output directory (optional)

example: batch_merger(aneurinibacilus, ("hitdata1.txt", "hitdata2.txt"))
```

3.9 consensus

This function improves the functional annotation of the complete bacterial genome using a consensus of three programs: eggNOG-mapper, Operon-mapper. and Batch CD-Search. The function saves all predicted features and COG assignments and prepares the outputs file for visualization with DNAPlotter. This function can be used for complete genomes only.

```
usage: consensus(organism_name, em_file, om_file, batch_file, fasta_file,
                 get_pseudo=False, get_ncrna=False, gff_file=None,
                 cat_choice=1, output_dir=output_path)

arguments:
  organism_name     organism name
  em_file           path to Eggnog-mapper processed file (em_processor's output)
  om_file           path to Operon-mapper processed file (om_processor's output)
  batch_file        path to Batch CD-Search processed file (batch_processor's output)
  get_pseudo        add pseudogenes to the final annotation file (optional)
  get_ncrna         add ncRNA to final annotation file (optional)
  gff_file          path to the features in GFF3 format* (optional)
  cat_choice        select the option to assign a category (0-4)
  output_dir        path to the output directory (optional)

example: consensus("aneurinibacilus", em_aneurinibacilus.gff",
                   "om_aneurinibacilus.gff","batch_aneurinibacilus.gff",
                   get_pseudo=True, get_ncrna=True, gff_file="features.gff",
                   cat_choice=3)

*Note: add only if get_pseudo or get_ncrna is True
```

3.10 consensus_draft

This function improves the functional annotation of the draft bacterial genome using a consensus of three programs: eggNOG-mapper, Operon-mapper, and Batch CD-Search.

```
usage: consensus_draft(organism_name, proteins, em_file, om_file,
                       batch_file, cat_choice = 1, output_dir=output_path):

arguments:
  organism_name     organism name
  proteins          path to downloaded proteins
  em_file           path to Eggnog-mapper processed file (em_processor's output)
  om_file           path to Operon-mapper processed file (om_processor's output)
  batch_file        path to Batch CD-Search processed file (batch_processor's output)
  cat_choice        select the option to assign a category (0-4)
  output_dir        path to the output directory (optional)

example: consensus_draft("aneurinibacilus","em_aneurinibacilus.txt",
                         "om_aneurinibacilus.txt","batch_aneurinibacilus.txt",
                         cat_choice=3)
```

3.11 get_track_template

Generates the file for the Track Manager option in DNAPlotter.

```
usage: get_track_template(pos_track=(0.95, 0.90, 0.85, 0.80), size=10.0,
                          cogs_palette=True, output_dir=output_path)

arguments:
  pos_track         the positions for plotting features (CDS forward strand,
                    CDS reverse strand, pseudogenes, RNA genes) (optional)
  size              the size of a track (optional)
  cogs_palette      use palette from COG database*
  output_dir        path to the output directory (optional)

example: get_track_template()

*Note: if False, the function uses the author's subjectively chosen colours to
distinguish categories properly, you can overwrite them in the last column in
the COGtools-data: fun-20.tab.txt file.
```

3.12 get_legend

Creates a legend for the genome map.

```
usage: get_legend(font="arial.ttf", cogs_palette=True, output_dir=output_path)

arguments:
  font              the used font
  cogs_palette      use palette from COG database
  output_dir        path to the output directory (optional)

example: get_legend()
```

3.13 categories_barplot

If you have annotated multiple genomes using COGtools, you can visualize the relative abundance of COG categories in the given genomes using barplots (the whole genomes or draft).

```
usage: categories_barplot(path_to_data, names=None, draft=False, cogs_palette=True,
include_unknown=True)

arguments:
  path_to_data          the path to annotated genomes
  names                 the names of annotated genomes
  draft                 work with the draft genome
  cogs_palette          use palette from COG database
  include_unknown       include unknown COGs

example: categories_barplot("data_for_barplot",
                            names=['E. coli','A. thermoae-\nrophilus','C. beijerinckii',
                            'C. diolis', 'R. rubrum','S. thermode-\npolymerans',
                            'T. taiwanensis'],
                            draft=False, cogs_palette=False)
```

The result from the example above is shown in Figure 1.

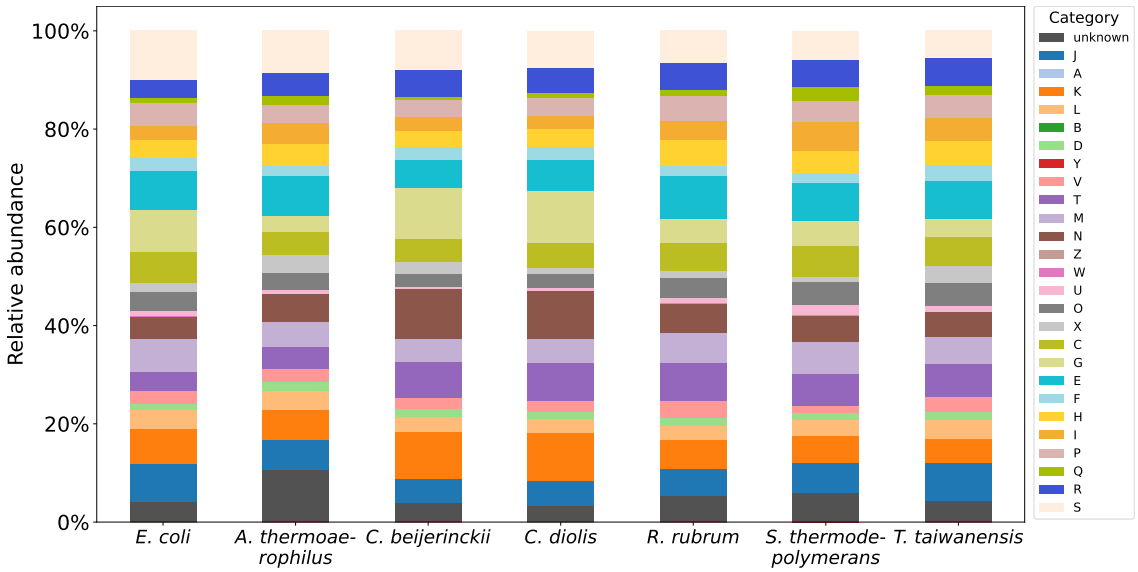


Figure 1: Distribution of COG categories in annotated bacteria.

4 The whole genome annotation (tutorial)

In this tutorial, the whole process of annotation and visualization of the chromosome of *Aneurinibacillus thermoaerophilus* CCM 8960 will be shown.

4.1 Input data

Please download the necessary files on the NCBI website: https://www.ncbi.nlm.nih.gov/nuccore/NZ_CP080764.1. How to download these files shows Figure 11.

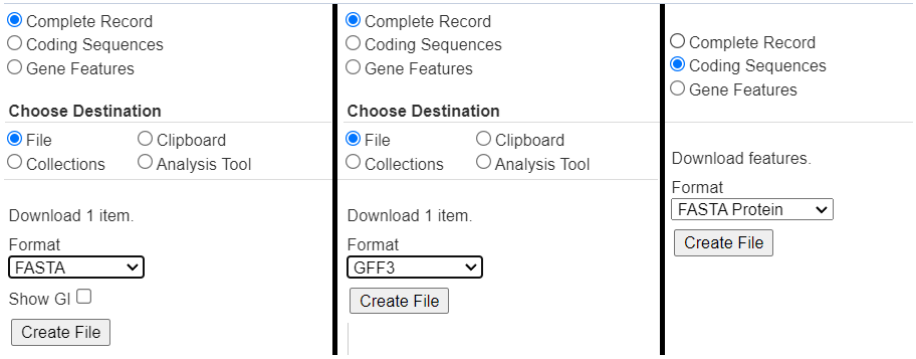


Figure 2: How to download the necessary files, from the left: sequence in FASTA, features in GFF3, CDS in FASTA.

4.2 eggNOG-mapper

On the website http://eggnog-mapper.embl.de/, please upload the downloaded CDS file in the Proteins option and fill in the necessary data. After annotation, download the decorated.gff file (see Figures 3 and 4).

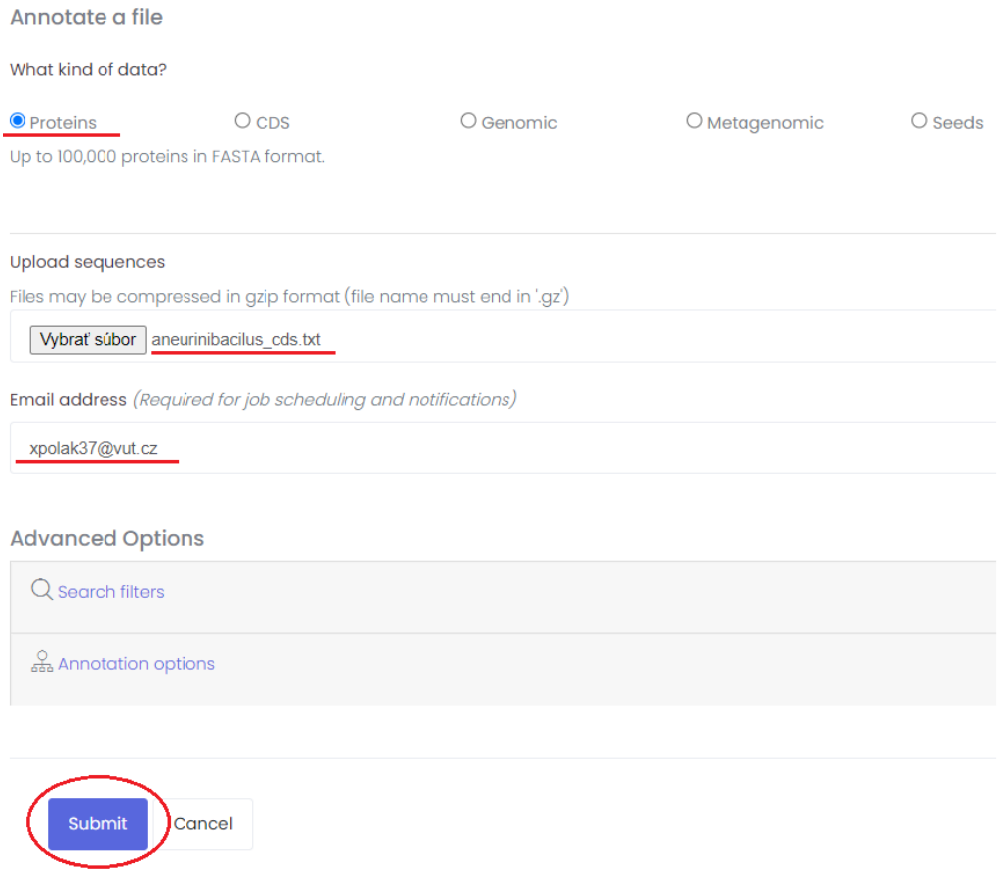


Figure 3: Settings of the eggNOG-mapper.

Figure 4: Required output file of the eggNOG-mapper.

4.3 Operon-mapper

On the website https://biocomputo.ibt.unam.mx/operon_mapper/, please upload the downloaded FASTA file with bacterial genome and GFF file with its features. After annotation, download the ORF coordinates and COGs assignations (see Figures 5 and 6).

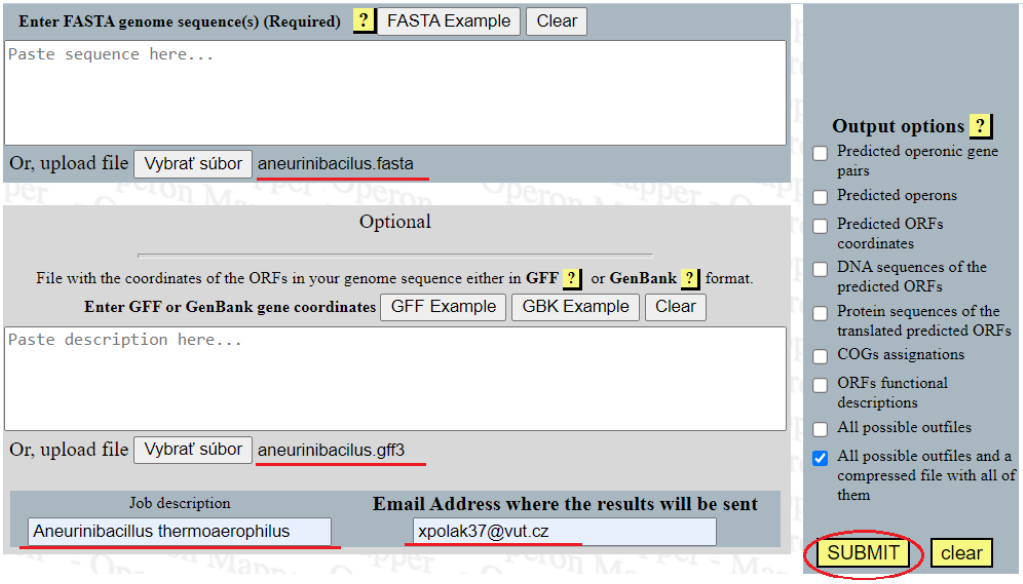


Figure 5: Settings of the Operon-mapper.

Output files		
1	Predicted operonic gene pairs	Download
2	Predicted operons	Download
3	Predicted ORFs coordinates	Download
4	DNA sequences of the predicted ORFs	Download
5	Protein sequences of the translated predicted ORFs	Download
6	COGs assignations	Download
7	ORFs functional descriptions	Download
8	Compressed file with all the above **3598509.tar.gz**	Download

Figure 6: Required output files of the Operon-mapper.

4.4 Batch CD-Search

On the website `https://www.ncbi.nlm.nih.gov/Structure/bwrpsb/bwrpsb.cgi`, please upload the downloaded CDS file and choose COG database in the Search against database option. After annotation, download the hitdata file (see Figures 7 and 8).

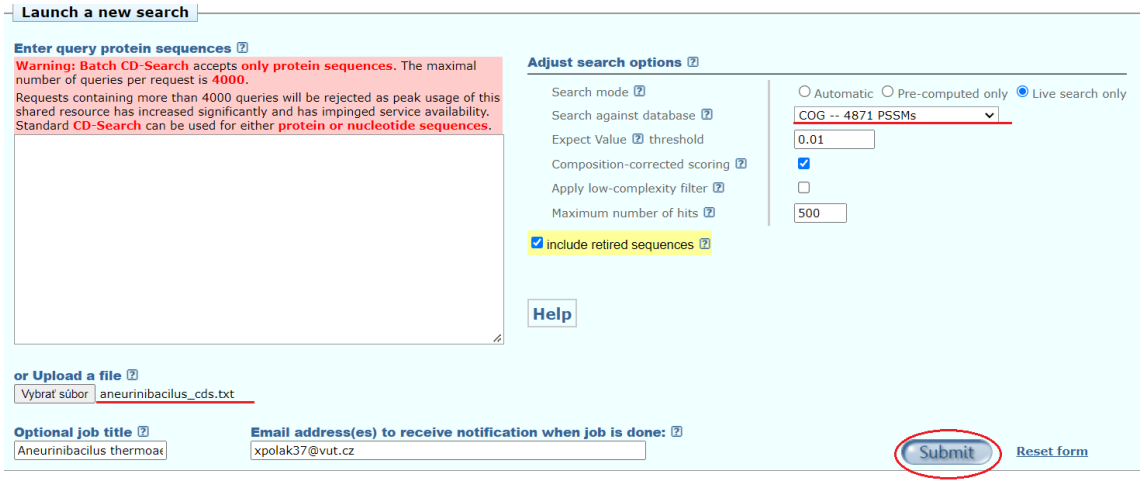


Figure 7: Settings of the Batch CD-Search.

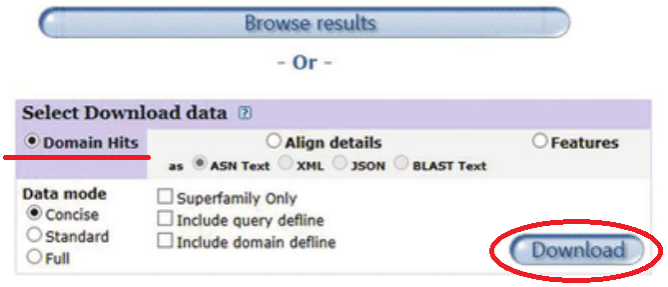


Figure 8: Required output file of the Batch CD-Search.

4.5 COGtools

You have two ways to use the COGtools. Either you use its functions in your script, or you call the whole process from the command line.

1. Using COGtools functions:

```
import cogtools

organism_name = "aneurinibacilus"
eggnog_file = "inputs/aneurinibacilus_eggnog.gff"
operon_file_gff = "inputs/aneurinibacilus_orf_operon.txt"
operon_file_cog = "inputs/aneurinibaciluss_cog_operon.txt"
batch_file = "inputs/aneurinibacilus_batch.txt"

fasta_file = "inputs/aneurinibacilus.fasta"
CDS_file = "inputs/aneurinibacilus_cds.txt"
gff_file = "inputs/aneurinibacilus.gff3"

cogtools.em_processor(organism_name, em_file=eggnog_file, cds_file=CDS_file,
                      output_dir = "outputs")
cogtools.om_processor(organism_name, operon_file_gff, operon_file_cog,
                      output_dir = "outputs")
cogtools.batch_processor(organism_name, batch_file, output_dir = "outputs")
cogtools.consensus(organism_name, output_dir + "/em_" + organism_name + ".gff",
                   output_dir + "/om_" + organism_name + ".gff",
                   output_dir + "/batch_" + organism_name + ".gff",
                   fasta_file=fasta_file, get_pseudo=True,
                   gff_file=gff_file, get_ncrna=True, cat_choice=1,
                   output_dir = "outputs")

cogtools.get_track_template(output_dir="outputs", cog_palette=False)
cogtools.get_legend(cog_palette=False)
```

2. Using COGtools via command line:

   To use this option, please make sure you have the files named according to the pattern in Section 2. Then just use the command like this (with your own paths):

```
py cogtools.py    -n aneurinibacillus
                -i C:/Users/ppola/genomes/aneurinibacilus/inputs
                -o C:/Users/ppola/genomes/aneurinibacilus/outputs -t -ch 1
```

This will generate 6 files (3 edited annotations, 1 final annotation, a track template, and a legend).

4.6 DNAPlotter

Use the DNAPlotter tool to visualize the obtained COG annotation.

1. Open the DNAPlotter.

2. Read in sequence file (file_to_plot.txt).
3. Go to Options - Track Manager - File - Import Track Template.
4. Read in track_template file.
5. Update tracks.

Following these steps, you will get a visualization similar to the one in Figure 9.

Figure 9: Chromosomal map of *Aneurinibacilus thermoaerophilus* CCM 8960.

You can use the generated legend along with this visualization (see Figure 10).

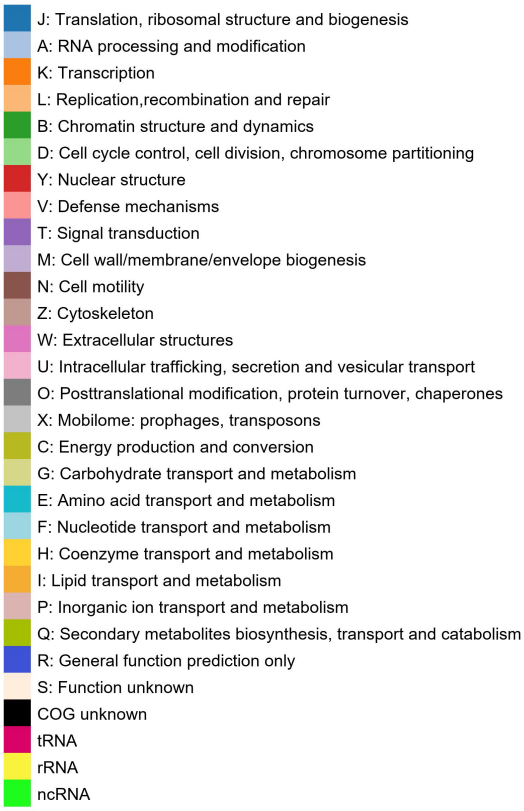


Figure 10: Generated legend of get_legend function.

5 The draft genome annotation (tutorial)

In this tutorial, the whole process of annotation draft genome of *Pseudomonas* sp. P2653 will be shown.

5.1 Input data

Please download the necessary files on the NCBI website: https://www.ncbi.nlm.nih.gov/Traces/wgs/JAKJXE01?display=contigs&page=1. How to download these files shows Figure 11.

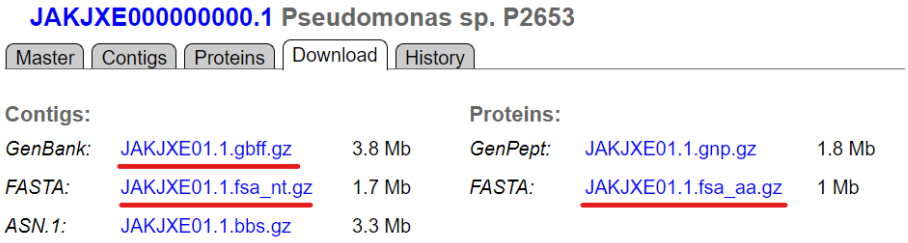


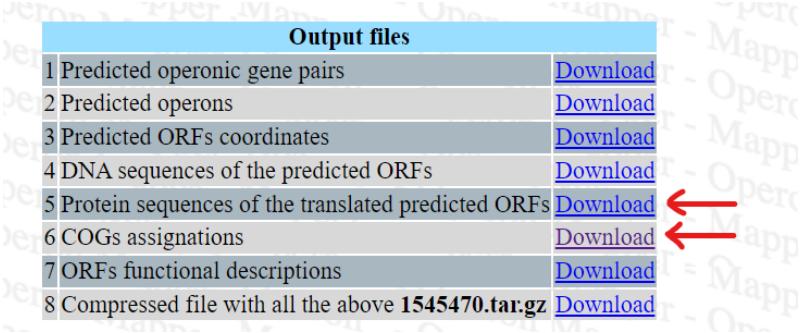
Figure 11: How to download the necessary files, contigs in FASTA, proteins in FASTA. You can also get contigs in Genbank, which you can use for Operon-mapper's annotation.

5.2 eggNOG-mapper

On the website http://eggnog-mapper.embl.de/, please upload the downloaded proteins in FASTA in the Proteins option and fill in the necessary data. After annotation, download the decorated.gff file (same as in section 4.2).

5.3 Operon-mapper

On the website https://biocomputo.ibt.unam.mx/operon_mapper/, please upload the downloaded FASTA file contigs and GFF file with its features (optional). If you do not use GFF, Operon-mapper will predict the proteins, so the results will be evaluated based on local alignment. After annotation, download the ORF coordinates and COGs assignations (see Figures 5 and 6).



Output files		
1	Predicted operonic gene pairs	Download
2	Predicted operons	Download
3	Predicted ORFs coordinates	Download
4	DNA sequences of the predicted ORFs	Download
5	Protein sequences of the translated predicted ORFs	Download
6	COGs assignations	Download
7	ORFs functional descriptions	Download
8	Compressed file with all the above **1545470.tar.gz**	Download

Figure 12: Required output files of the Operon-mapper.

5.4 Batch CD-Search

On the website https://www.ncbi.nlm.nih.gov/Structure/bwrpsb/bwrpsb.cgi, please upload the downloaded contigs and choose COG database in the Search against database option. After annotation, download the hitdata file (same as in section 4.4).

5.5 COGtools

1. Using COGtools functions:

```
import cogtools

organism_name = "pseudomonas"
eggnog_file = "inputs/pseudomonas_eggnog.gff"
operon_file_proteins = "inputs/pseudomonas_proteins_operon.txt"
operon_file_cogs = "inputs/pseudomonas_cogs_operon.txt"
batch_file = "inputs/pseudomonas_batch.txt"

contigs = "inputs/pseudomonas.fsa_nt"
proteins = "inputs/pseudomonas.fsa_aa"
gff_file = "inputs/pseudomonas.gff3"

cogtools.em_processor(organism_name, em_file=eggnog_file, output_dir= "outputs")
cogtools.om_processor(organism_name, proteins, operon_file_proteins,
                      operon_file_cogs,output_dir = "outputs")
cogtools.batch_processor(organism_name, batch_file, output_dir = "outputs")
cogtools.consensus(organism_name, proteins,
                   output_dir + "/em_" + organism_name + ".txt",
                   output_dir + "/om_" + organism_name + ".txt",
                   output_dir + "/batch_" + organism_name + ".txt",
                   cat_choice=1, output_dir = "outputs")
```

2. Using COGtools via command line:

```
py cogtools.py -n pseudomonas
-i C:/Users/ppola/genomes/pseudomonas/inputs
-o C:/Users/ppola/genomes/pseudomonas/outputs -d -ch 1 -g
```

References

[1] HUERTA-CEPAS, Jaime, Damian SZKLARCZYK, Davide HELLER, et al., 2019. EggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. *Nucleic Acids Research* [online]. **47**(D1), D309-D314. ISSN 0305-1048. Available at: https://doi.org/10.1093/nar/gky1085

[2] TABOADA, Blanca, Karel ESTRADA, Ricardo CIRIA, Enrique MERINO and John HANCOCK, 2018. Operon-mapper: a web server for precise operon identification in bacterial and archaeal genomes. *Bioinformatics* [online]. **34**(23), 4118-4120. ISSN 1367-4803. Available at: https://doi.org/10.1093/bioinformatics/bty496

[3] MARCHLER-BAUER, Aron and Stephen H. BRYANT, 2004. CD-Search: protein domain annotations on the fly. *Nucleic Acids Research* [online]. **32**(Web Server), W327-W331. ISSN 0305-1048. Available at: https://doi.org/10.1093/nar/gkh454

[4] CARVER, T., N. THOMSON, A. BLEASBY, M. BERRIMAN and J. PARKHILL. 2008. DNAPlotter: circular and linear interactive genome visualization. *Bioinformatics* [online]. **25**(1), 119-120. Available at: https://doi.org/10.1093/bioinformatics/btn578