Methods in Molecular Biology

Comparative Genomics, Volume 2

Chapter: Using protein-coding gene families in bacterial genome comparisons

Authors: Dennis Carhuaricra-Huaman and João Carlos Setubal

ABSTRACT

The identification of orthologous genes is relevant for comparative genomics, phylogenetic analysis and functional annotation. Many bioinformatics tools exist to predict orthologous using different computational strategies and web-based resources that collect orthology data available for online analysis. In this protocol method, we present a guide to infer orthologous from a dataset of ten prokaryotic proteomes using four best-know methods: Orthomcl, COGtriangles, orthofinder2 and OMA. We compare the number of orthologous groups predicted and present a brief workflow for the functional annotation and reconstruction of phylogeny from inferred single-copy orthologous genes. Furthermore, we explore two orthology databases: eggNOG6 and OrthoDB and evaluate their capability to detect remotely conserved orthologous in prokaryotes, the user-friendliness and the information they provide.

KEYWORDS: Orthology inference, orthology resources, phylogeny inference, functional annotation

1. INTRODUCTION

Homology describes an evolutionary relationship between genes or proteins. Two subtypes of homology relationships were recognized by Walter Fitch [1], orthology and paralogy. Orthology refers to genes that originated from a speciation event, whereas paralogy to genes that arise from a gene duplication event. The original motivation behind identifying orthologs was to carry out phylogenetic reconstruction, since by definition they possess the same evolutionary history as the underlying species. However, nowadays, another motivation for distinguishing between orthologs and paralogs is the prediction of the function of newly annotated genes [2, 3]. Ortholog genes are more likely to have the same function because they were the same gene in the most recent common ancestor. The identification of orthologs is essential in many applications such as comparative genomics, phylogenetics and functional annotation [4]. Currently, high-throughput sequencing methods have increased the availability of complete genomes (and their proteomes), and robust computational methods for identifying orthologs are crucial to explore the vast amount of genomic data [5].

Computational methods for identifying orthologous sequences can be mainly divided into two groups: phylogeny-based and graph-based methods [6]. Phylogeny-based methods require aligning a group of homologous sequences, computing a phylogenetic tree, and inferring the type of evolutionary event represented by each internal node in the generated tree, a process that involves reconciling the gene tree with the species tree [7]. While this method tends to be more accurate, it requires more computational work and is not as easily scalable. On the other hand, graph-based approaches rely on sequence similarity searches and consider two sequences orthologous if they are each other's best hit in their respective proteomes [6, 8]. Several software tools combine both methods using attributes of graph-based and tree-based methods in the inference of orthology relationships [9].

In this chapter, we cover the topic of computational inference of orthology from a practical point of view. There are basically two ways this can be done: a) using orthology inference programs with user-defined proteomes and executed locally; b) using existing orthology database resources in a website. We will use publicly available proteomes from ten prokaryotic species to demonstrate different methods and resources for orthologous inference and their application in phylogenetic analysis and functional annotation.

In this chapter we cover the case in which the genomes to be compared are distantly related. For closely related genomes (e.g. strains of the same species), we refer the reader to Chapter CGPROKS.

2. REQUIREMENTS AND ASSUMPTIONS

This chapter assumes basic knowledge of Unix/Linux and R. All analyses can be run on a desktop computer running Linux/Unix or Mac OSX. Most programs are run using bash shell commands. We show commands executed on the Linux shell preceded by the "$" symbol and the label **bash shell**. We also present R code, which can be executed in the RStudio environment (https://posit.co/download/rstudio-desktop/). R code sections are preceded by the label **R script**.

3. DATASETS

Orthology algorithms use protein sequences as input to predict orthologous families. In this chapter, we use sets of protein sequences (proteomes) from ten species of prokaryotes, spanning six different phyla, to present different methods and software tools. Proteomes can be found in public repositories, such as GenBank, and by GenBank convention have the extension .faa. Table 1 provides the accession numbers in GenBank from where the proteomes used in this chapter were downloaded.

Table 1. Information on the proteomes used for presenting orthologous resources

label	species	Kingdom	Phyla	Class	Biosample Accession
Bfragilis	*Bacteroides fragilis*	Bacteria	Bacteroidetes	Bacteroidia	SAMN16357367
Ecoli	*Escherichia coli*	Bacteria	Proteobacteria	Gammaproteobacteria	SAMN02604091
Hinfluenzae	*Haemophilus influenzae*	Bacteria	Proteobacteria	Gammaproteobacteria	SAMN02595602
Kgyiorum	*Kerstersia gyiorum*	Bacteria	Proteobacteria	Betaproteobacteria	SAMN10273024
Lacidophilus	*Lactobacillus acidophilus*	Bacteria	Firmicutes	Bacilli	SAMN02603216
Mgenitalium	*Mycoplasma genitalium*	Bacteria	Firmicutes	Tenericutes	SAMN02603983
Mjannaschii	*Methanococcus jannaschii*	Archea	Euryarchaeota	Methanococci	SAMN02603984
Mtuberculosis	*Mycobacterium tuberculosis*	Bacteria	Actinobacteria	Actinobacteridae	SAMEA3138326
Rradiobacter	*Rhizobium radiobacter*	Bacteria	Proteobacteria	Alphaproteobacteria	SAMN10169604
Synechocystis	*Synechocystis sp.*	Bacteria	Cyanobacteria	Cyanophyceae	SAMD00061113

4. SOFTWARE

Table 2 summarizes the software tools employed in this chapter, along with links to the corresponding websites where they can be downloaded and installed.

Table 2. List of bioinformatics tools used in this chapter

Tool	Description	Ref	Source
eggNOG-mapper	Orthologous inference and functional annotation	[10]	https://github.com/eggnogdb/eggnog-mapper
Egg-NOG v6	Orthology resource	[11]	http://eggnog6.embl.de/
GenBank	Genome database	[12]	https://www.ncbi.nlm.nih.gov/genbank/
get_homologues	Orthologous inference	[13]	https://github.com/eead-csic-compbio/get_homologues
ggtree	phylogenetic visualization	[14]	https://github.com/YuLab-SMU/ggtree
IQ-TREE2	Phylogenetic inference	[15]	https://github.com/iqtree/iqtree2
MAFFT	Sequence alignment	[16]	https://mafft.cbrc.jp/alignment/software/
OMA	Orthologous inference	[17]	https://github.com/DessimozLab/OmaStandalone
OrthoDB	Orthology resource	[18]	https://www.orthodb.org/

OrthoFinder2	Orthologous inference	[9]	https://github.com/davidemms/OrthoFinder
seqkit	Sequence concatenation	[19]	https://bioinf.shenwei.me/seqkit/
trimAL	Sequence trimming	[20]	https://github.com/inab/trimal

5. GENERATING ORTHOLOGS BY VARIOUS METHODS

We say that two genes are homologous when they share a common ancestor gene. In a somewhat simplified scenario, two genes can have a common ancestor in two situations. The first is caused by speciation. Gene *a* in species *X* is homologous to gene *b* in species *Y* when *X* and *Y* have a common ancestor *Z*, and *Z* had a gene that can be said to be the ancestor of *a* and *b* (by virtue of the fact that *Z* is the ancestor of *X* and *Y*). In this case we say that not only *a* and *b* are homologous, but they are also **orthologous**, to distinguish it from the second situation. That happens when *a* and *b* are the result of gene duplication. Duplicated genes are named **paralogs**; however, there can be two sub-cases, depending on when duplication takes place with respect to speciation. If duplication occurs *before* speciation, the resulting genes in the descendant species are called **out-paralogs**. If duplication occurs *after* speciation, the duplicated genes in a given species are called **in-paralogs**. When analyzing several genomes, as is the case in this Chapter, genes that are orthologs among all pairs of genomes are grouped together as Orthologous Groups (OGs), also called orthogroups.

The inference of OGs can be challenging due to various evolutionary events, such as gene duplication, loss, or horizontal gene transfer. There are several programs to infer orthology currently available. In this Chapter, three different graph-based algorithms are used to predict OGs in our dataset: OrthoMCL (which

uses Markov chains for clustering), the COG algorithm (which uses the BeT method for clustering), and OMA standalone (which uses 'cliques' and hierarchical clustering methods). Additionally, Orthofinder2, which combines graph-based and tree-based methods, is also used.

5.1. OrthoMCL

OrthoMCL [21] is a program for determining orthogroups in a given set of genomes. In addition to the original reference, a summary explanation of its algorithm can be found in [22]. OrthoMCL distinguishes between orthologs and paralogs. Two genes from the same species will be included in the same OG only if their reciprocal BLAST similarity is larger than the similarity to any gene not belonging to that species. Those genes are considered "recent in-paralogs".

The standalone version of OrthoMCL can be downloaded from the website https://orthomcl.org/orthomcl/app/downloads/software/ and installed locally. However, it is not a simple program to run, because its pipeline requires the installation of other programs (BLAST, MySQL, MCL). In this Chapter, we run OrthoMCL through the convenient Get_Homologues package [13], which offers three clustering algorithms, including OrthoMCL.

In our example, Get_Homologues is executed in a directory (which we named prok_proteomes) that contains the full proteome sequences of the ten prokaryote genomes that were previously downloaded from Genbank (Table 1).

Bash shell

```
$ get_homologues.pl -d ./prok_proteomes/ -M -C 30 -t 2
```

The flag -M specifies the use of OrthoMCL algorithm to compute orthologs. An output folder named 'prok_proteomes_homologues' is created after the running is finished; it includes a subfolder that contains separate FASTA files for each orthologous group. The options -C specifies the minimum prcentage of coverage in BLAST pairwise, we set it at 30%. Any OG may contain in-paralogs; the option -e should be called to exclude clusters containing in-paralogs:

```
$ get_homologues.pl -d ./prok_proteomes/ -M -C 30 -e
```

In this case, the resulting clusters will contain only single-copy genes from each taxon, a desirable property for phylogenetic reconstruction.

We ran orthoMCL on our dataset; the results are shown in Table 3.

5.2. The COG algorithm

COG stands for “Cluster of Orthologous Groups”, which is the name used by its authors [23] for orthogroups. The algorithm for determining COGs is called the COGtriangles algorithm and is briefly explained in [22]. The algorithm has a pre-processing step in which all in-paralogs are determined and converted to single vertices. The COGtriangles algorithm was refined in terms of computational complexity by the EdgeSearch algorithm [24]. EdgeSearch is also one of the options of the Get_Homologues package [13].

To run the COGtriangles algorithm in the Get_Homologues package use the option -G as follows:

Bash shell

```
$ get_homologues.pl -d ./prok_proteomes_fasta -G -C 30 -t 2
```

An output subfolder is created inside the 'prok_proteomes_homologues' folder, containing separate FASTA files for each OG. Table 3 summarizes the number of clusters predicted by this algorithm.

Get_Homologues can compare the output of the different methods implemented by the software. The script compare_clusters.pl identifies the common elements between the sets of clusters generated by orthoMCL and COG algorithms. The command is shown below.

Bash shell

```
$ compare_clusters.pl -o sample_intersection -d
prok_proteomes_homologues/*_algOMCL_*,prok_proteomes_homologues/
*_algCOG_*
```

A folder called sample_intersection is created containing the list of clusters commonly predicted by both algorithms and a consensus visualized in a Venn diagram (Figure 1).

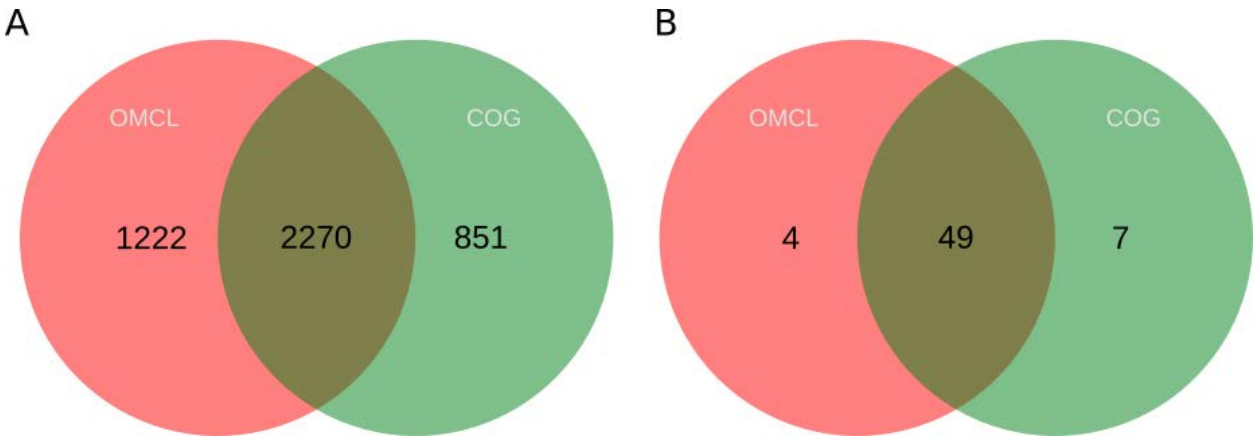


Figure 1. Venn diagram depicting the overlap of orthologous groups predicted by OrthoMCL and COG algorithms in A) OGs containing at least 2 sequences and B) single copy OGs containing genes in all ten species.

5.3. OrthoFinder2

OrthoFinder2 [9] is a hybrid method that combines graph and tree-based approaches. In the first step, OrthoFinder2 incorporates BLAST score normalization in order to predict highly accurate OGs by taking into account gene length bias [25]. In the second step, these OGs are used to infer unrooted gene trees with dendroBLAST; after that, these gene trees are used for the identification of gene duplication events. Ultimately, all of the phylogenetic data obtained is used to determine the complete set of orthologs between all species. Altogether, Orthofinder2 achieves very high ortholog inference accuracy on the Quest for Orthologs benchmarks [9, 26].

Running OrthoFinder2 is a straightforward process that only requires a collection of protein sequence files (one for each species) in FASTA format. The command is shown below.

Bash shell

```
$ orthofinder -f ./prok_proteomes_fasta
```

OrthoFinder2 generates a folder named "OrthoFinder" to store the output. The file Statistics_Overall.tsv, which contains the number of genes assigned and the number of orthogroups clustered, is located within the subfolder Comparative_Genomics_Statistics. Table 3 summarizes the results.

5.4. OMA

Orthologous Matrix (OMA) is a graph-based software that provides three different types of orthologs: pairwise orthologs, OMA Groups, and Hierarchical Orthologous Groups (HOGs), each type based on a different inference method [17, 27]. The OMA algorithm begins by inferring pairwise orthologs, which are

subsequently utilized to construct both HOGs and OMA Groups. To infer OGs, it first computes all-against-all Smith-Waterman alignments, saving only candidate pairs with sufficient score and overlap. OMA groups are clustered by identifying well-connected subgraphs ("cliques") corresponding to sequences that are strictly orthologous, excluding orthologs involved in 1-to-many and many-to-many relations.

The other strategy of clustering performed by OMA software is hierarchical clustering. Hierarchical orthologous groups (HOGs) are formed starting with the most specific taxonomic level and merging groups progressively towards the root of the species tree [28]. These HOGs may include both orthologs and in-paralogs with respect to the reference speciation.

The OMA algorithm is available as open-source software, OMA Standalone (https://omabrowser.org/standalone/#downloads), which is compatible with the public OMA Browser (https://omabrowser.org).

To run OMA, all proteomes (.faa files) should be located in a folder called DB. OMA runs with the following commands:

Bash shell

```
$ oma -p
$ oma -s
$ oma
```

All resulting files will be located in a folder called 'output'. Inside, the *OrthologousGroupsFasta* subfolder will contain all OGs; these are 1-to-1 strict orthologs. Table 3 shows the OGs predicted by OMA in our dataset.

Table 3. The number of orthologous groups predicted by four different algorithms among the proteomes of 10 prokaryotic species.

	orthoMCL	COG algorithm	OrthoFinder2	OMA
All orthologs groups (containing 2 or more homologous genes)	3492	3121	4205	4389
Single-Copy Orthogroups (SCOGs) in all ten species	53	56	53	17

5.5. Performance of orthology inference methods

Given that there are multiple orthology inference methods, it is natural to ask which one is the most accurate. This is a difficult question to answer. In what follows, we provide a simple comparison of the results we presented above, and then briefly describe a multi-group initiative to provide benchmarks for assessing the quality of orthology inference methods.

The Quest For Orthologs (QFO) consortium [29] offers an online benchmarking tool for orthology prediction (http://orthology.benchmarkservice.org). The benchmarking service has established a reference proteome dataset to enable comparisons of individual inference methods on a consistent set of species and proteins. The latest version of the database comprises 78 Uniprot reference proteomes from model organisms and species of biomedical interest (48 Eukaryotes, 23 Bacteria and 7 Archaea).

The benchmark service assesses the quality of predictions based on: species tree discordance test, agreement with reference phylogenies/orthologs and functional test. Each test measures the performance in terms of precision and recall. In simple terms, Precision measures the number of false positives whereas recall measures the number of false negatives. A false positive is a gene that was wrongly considered as belonging to family X. A false negative is a gene that should have been assigned to family X but was not. Additional details can be found in [29].

In our dataset, the number of all orthologs (containing two or more species) predicted by OMA (4,389 clusters) is higher than that predicted by orthoMCL (3,492) and COG algorithms (3,121), but similar to OrthoFinder2 (4,205). However, when we limit our observation to single-copy orthogroups in ten species, the pattern changes. Even though OMA yielded the highest number of orthologs, it inferred the fewest number of SCOGs in all ten species (17 SCOGs); whereas Orthofinder2, OrthMCL and COG predicted 53, 53 and 56 SCOGs, respectively.

These results are consistent with previous studies. For example, Altenhoff et al. compared several orthology methods on a reconstruction of the Lophotrochozoa phylogeny where OMA widely predicted more ortholog groups than OrthoMCL and OrthoFinder, but OMA produced a smaller quantity of larger groups. This difference in group size distribution is likely the result of different trade-offs in terms of precision (proportion of predicted orthologs that are correct) and recall (proportion of true orthologs that are correctly predicted). A recent benchmarking study comparing 20 different orthology prediction methods highlighted the trade-off between precision and recall [26]. The OMA Groups method ranks as one of the best in precision but lower recall than most other methods. This means that genes are often missing from OGs, or OGs are more fragmented than they should. OrthoMCL performs among those with the highest recall, though with lower accuracy than other methods. OrthoFinder2, Domanoid [30] and

Orthoinspector [31] show a good balance between precision and recall over the benchmark (https://orthology.benchmarkservice.org/).

In terms of runtime, OMA is by far the most costly of the orthology methods tested, due to its reliance on full Smith–Waterman alignments and evolutionary distance in the all-against-all phase.

6. OBTAINING PHYLOGENETIC TREES FROM ORTHOGROUPS

Phylogenetic reconstruction is one of the main goals of orthologous inference. In this subsection we compare the phylogenies produced from the sequence alignment of single-copy OGs (SCOGs) predicted by OrthoFinder2 and OMA methods.

6.1. OrthoFinder2

We use the 53 SCOG sequences predicted by OrthoFinder2 to infer a phylogenetic tree. After running OrthoFinder2 (section 5.3) the SCOGs sequences are located in the folder OrthoFinder/Single_Copy_Orthologue_Sequences.

First, We navigate into the result folder

Bash shell

```
$ cd OrthoFinder/Single_Copy_Orthologue_Sequences
```

We rename the FASTA headers in all OG files using the column label in Table 1. As all protein sequences are sorted alphabetically we use the following code:

```
$  for i in *.fa; do awk 'NR==FNR{names[NR]=$0; next}
/^>/{$1=">"names[++c]}1' label.tab $i > {i%.*}_rn.fa; done
```

Then, we align all protein sequences in OGs using MAFFT as follows:

```
$ for i in *_rn.fa; do mafft --maxiterate 1000 --localpair $i >
${i%%.*}_alig.fa; done
```

Poorly aligned regions are removed using trimAL as follows:

```
$ for i in *_alig.fa; do trimal -automated1 -in $i -out
${i%%.*}_trim.fa; done
```

A supermatrix is then constructed by concatenating all alignments using the “concat” option of seqkit tool:

```
$ seqkit concat *_trim.fa > concat_orthofinder.fa
```

The resulting concat.fa FASTA file is used to construct the phylogenomic tree with IQTREE2 software using a maximum-likelihood algorithm.

```
$ iqtree2 -s concat_orthofinder.fa -m LG -bb 1000 -o
"Mjannaschii"
```

The tree inferred can be visualized (Figure 2A) with *ggtree* package using the following commands:

R script

```
library(ggtree)

# read tree file
tree <- read.tree("concat_orthofinder.fa.treefile")
# read metadata (table 1)

metadata <- read.table("metadata.tab", sep = "\t", header = T)

# Draw the tree displaying bootstrap value
```

```
gg <- ggtree(tree, layout = "rectangular",right = T)  +
  geom_nodelab(aes(label=label), size=2,hjust = -0.3)
# add tip labels
p1 <- gg  %<+% metadata +
  geom_tiplab(aes(label=paste("bolditalic('", species,
"')~","'('~", Fila,"~')'")), parse=T,size=2.2, hjust = -0.02)
p1
```

6.2. OMA

In order to infer the phylogenetic tree of the ten species using OMA, first we need to extract all 17 SCOGs predicted by it (Section 5.4).

Bash shell

```
# We navigate into the result folder
$ cd Output/OrthologousGroupsFasta

# Create a directory to copy all 17 SCOG to there

$ mkdir OG_10species

# copy all OGs that contain 10 protein sequences into the “OG_10species” directory
$ for i in *.fa; do occurrences=$(grep -c ">" "$i"); if ((
occurrences == 10 )); then cp "$i" OG_10species; fi; done
```

We then navigate to the folder “OG_10species” to align all sequences within each OG using MAFFT, as follows:

Bash shell

```
$ cd OG_10species

$ for i in *.fa; do mafft --maxiterate 1000 --localpair $i >
${i%%.*}_alig.fa; done

# Poorly aligned regions are removed using trimAL as follows:

$ for i in *_alig.fa; do trimal -automated1 -in $i -out
```

```
${i%%.*}_trim.fa; done
```

In order to construct the phylogeny from these alignments, it is necessary to merge them into a single alignment.

Bash shell

```
$ seqkit concat *_trim.fa > concat_OMA.fa
```

The phylogenetic tree can be inferred using IQTREE2 software.

Bash shell

```
# run iqtree2 using the concatenated alignment fasta as input

# use LG as the model (-m) and 1000 replicates of bootstrap (-bb)
$ iqtree2 -s concat_OMA.fa -m LG -bb 1000 -o “Mjannaschii”
```

To draw the tree, we will use ggtree package as follows:

R script

```
library(ggtree)

# Read the treefile produced by IQTREE2
tree <- read.tree("cancat_OMA.fa.treefile")

# Read metadata (table 1)
metadata <- read.table("metadata.tab", sep = "\t", header = T)

# Draw the tree
gg <- ggtree(tree, layout = "rectangular", right = T)  +
```

```
  geom_nodelab(aes(label=label), size=2,hjust = -0.3)

# add tip labels
p2 <- gg  %<+% metadata +
  geom_tiplab(aes(label=paste("bolditalic('", species,
"')~","'('~", Fila,"~')'")), parse=T,size=2.2, hjust = -0.02)
p2
```

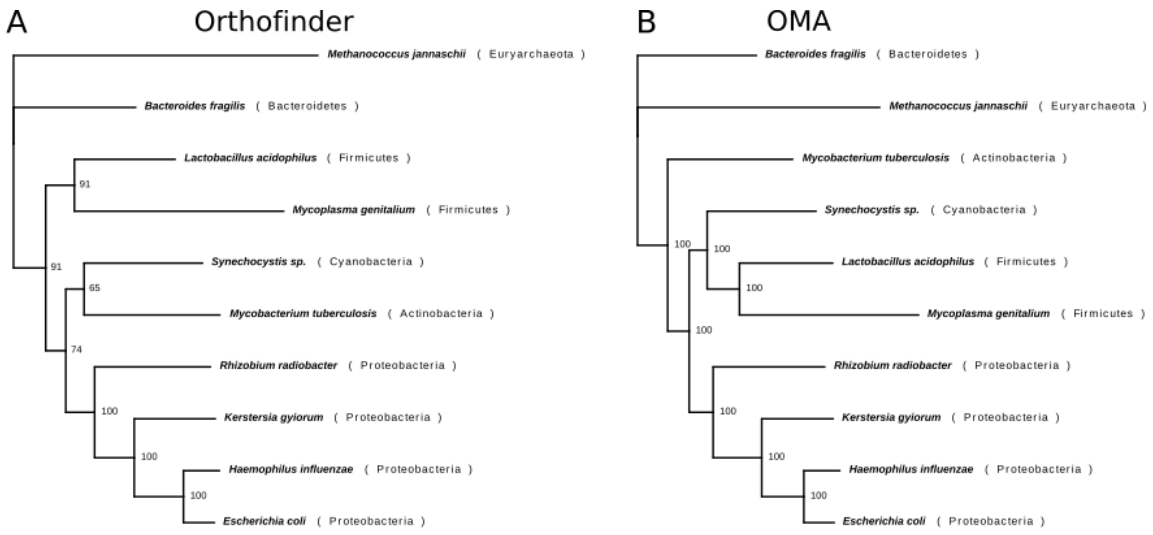


Figure 2. Phylogenetic tree constructed using maximum likelihood algorithm based on the alignment of A) 53 SCOGs predicted by OrthoFinder2 and B) 17 SCOGs predicted by OMA. The tip labels represent the scientific name of the species and the respective phylum is shown between parentheses. Bootstrap values are represented by the number on nodes.

The phylogenetic tree generated using 17 orthogroups predicted by OMA shows a topology similar to that constructed using 53 OGs predicted by OrthoFinder2. The innermost clade contains *E. coli* and *H. influenzae.* Since both belong to the class gammaproteobacteria, this was expected. *K. gyiorum* (betaproteobacteria) and *R. radiobacter* (alphaproteobacteria) are also more closely related as they

belong to the phylum proteobacteria. Likewise, the two species of the phylum firmicutes (*M. genitalium* and *L. acidophilus*) group together. On the other hand, while *Synechocystis sp.* (cyanobacteria) and Firmicutes form a well-supported cluster species in the OMA-based tree, in the OrthoFinder2 tree we observe that *Synechocystis sp.* and *Mycobacterium tuberculosis* (Actinobacteria) cluster together although with a low bootstrap support. Bootstraps values were used to estimate the branch support of the phylogenetic tree using 1000 replicates. Bootstrapping is a computational strategy to measure how strongly the sequence data support the phylogeny. When the value is closer to 100 means that the node is well-supported.

Qualitative comparison between trees generated from Orthofinder and OMA do not show remarkable differences. Another way to compare discordance between trees topology is using the Robinson-Foulds (RF) distance (Robinson and Foulds, 1981), a metric that verifies the distance between two trees. RF ranges from 0 to 1, with higher values indicating greater differences between compared trees.

7. eggNOG-mapper

eggNOG-mapper [10] is a gene functional annotation tool. It utilizes the vast eggNOG database [11] of orthologous groups, which spans thousands of bacterial, archaeal, and eukaryotic organisms. (This database is covered in more detail in Section 8; eggNOG is an acronym for evolutionary gene genealogy Non-supervised Orthologous Groups) To achieve this, the tool uses precomputed phylogenies for each OG to enhance orthology assignments, making it possible to annotate a gene by the transfer of annotations from close orthologs. eggNOG-Mapper is able to differentiate between orthologous and paralogous gene groups, which is important for functional assignment purposes [2].

eggNOG-mapper was specifically designed to annotate vast collections of sequences, primarily focusing on protein-coding genes from genomes, metagenomes, and transcriptomes. The functional attributes assigned to query sequences consist of curated COG functional categories [32], Gene Ontology terms [33],

KEGG pathways [34] and carbohydrate-active enzymes (CAZymes) terms [35]. eggNOG-mapper is available online (http://eggnog-mapper.embl.de/) and can also be used in a standalone version (https://github.com/eggnogdb/eggnog-mapper).

Three options for the initial sequence-mapping step are available: Diamond [36], MMseqs2 [37], and HMMER3 [38]. Diamond is the default mode and the best, considering memory and speed. Compared to the other two modes, HMMER3 mode is slower and requires the download of large databases. Nevertheless, utilizing HMM-based searches may be helpful in detecting distant homology relationships.

We use eggNOG-mapper to annotate single-copy OGs by four methods: OrthoMCL, COG, OrthoFinder2, and OMA (described in Sections 5.1-5.4).

Bash shell

```
# the following code is an example for OMA output

# navigate into the directory containing 17 OGs in all ten species

$ cd OG_10species

# merge all sequences in one file

$ cat *.faa > OGs_OMA.faa

# run eggNOG mapper

$ emapper.py -i OGs_OMA.faa -o eggnog_OMA
```

eggNOG-mapper by default will run diamond blastp. The option -m changes the search algorithm to MMseqs2 (-m mmseqs) or HMMER (-m hmmer).

The output consists of three files. The most important is the annotation file (outputname.emapper.annotations), which provides the functional predictions for each query (COG category, KEGG pathway, OG terms and CAZy terms) in TSV format. A comparative list of predicted single-copy OGs is displayed in Figure 3.

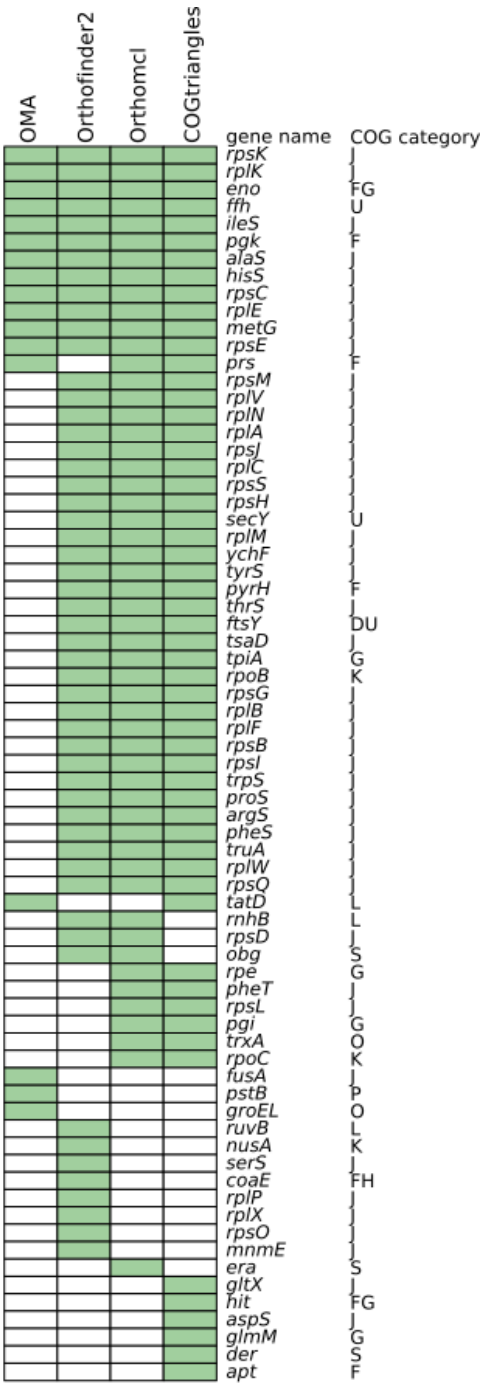


Figure 3. List of SCOGs predicted by OMA, OrthoFinder2, orthoMCL and COGtriangle algorithms including annotation using COGs and KEGG orthology (KO), inferred by eggNOG-mapper. Some SCOGs were annotated into two different COG categories.

what about OGs that were *not* annotated with COGs, such as rpsC and rpsE? the COG column is empty, but there is a COG category assigned to these genes

A total of 71 different OGs were predicted: 12 OGs were predicted by all four methods, 23 OGs were predicted by at least two methods. Most OGs were assigned to the COG category of "translation, ribosomal structure, and biogenesis" (J).

For better visualization, we plot the single-copy orthogroups predicted by each of the four orthology inference methods as a function of the COG category (Figure 4).

```
library(ggplot2)
library(dplyr)
library(tidyverse)

#read the tsv output file from eggNOG-mapper after removing all
comment lines
oma<- read.table("mapper_OMA.tab", header = T, sep = "\t")
oma_g <- as.data.frame(table(oma$COG_category))
of<- read.table("mapper_of.tab", header = T, sep = "\t")
of_g <- as.data.frame(table(of$COG_category))
omcl <- read.table("mapper_OMCL.tab", header = T, sep = "\t")
omcl_g <- as.data.frame(table(omcl$COG_category))
```

```
cog <- read.table("mapper_COG.tab", header = T, sep = "\t")
cog_g <- as.data.frame(table(cog$COG_category))

# combine all outputs in one table
combined <- rbind(oma_g,of_g, omcl_g, cog_g)


# Add a new column with the method name
combined <- transform(combined, method =
rep(c("OMA","Orthofinder", "OrthoMCL", "COG"), times =
c(nrow(oma_g),nrow(of_g), nrow(omcl_g), nrow(cog_g))))
a <- combined %>% group_by(method) %>%
    mutate(freq = Freq*100/sum(Freq))
# Set categories position on x axis
positions <- c("J", "K","L","F","G","P","FG","FH","O","U", "DU")
# Draw the barplot
ggplot(a, aes(fill=method, y=freq, x=Var1)) +
geom_bar(position = position_dodge2(width = 1.4, preserve =
"single"), stat="identity", colour="black", size=0.2)+
  scale_x_discrete(limits = positions) +
  xlab("COG category") + ylab("% of proteins")+
  theme_bw()+
  scale_fill_brewer(palette = "Set3")
```

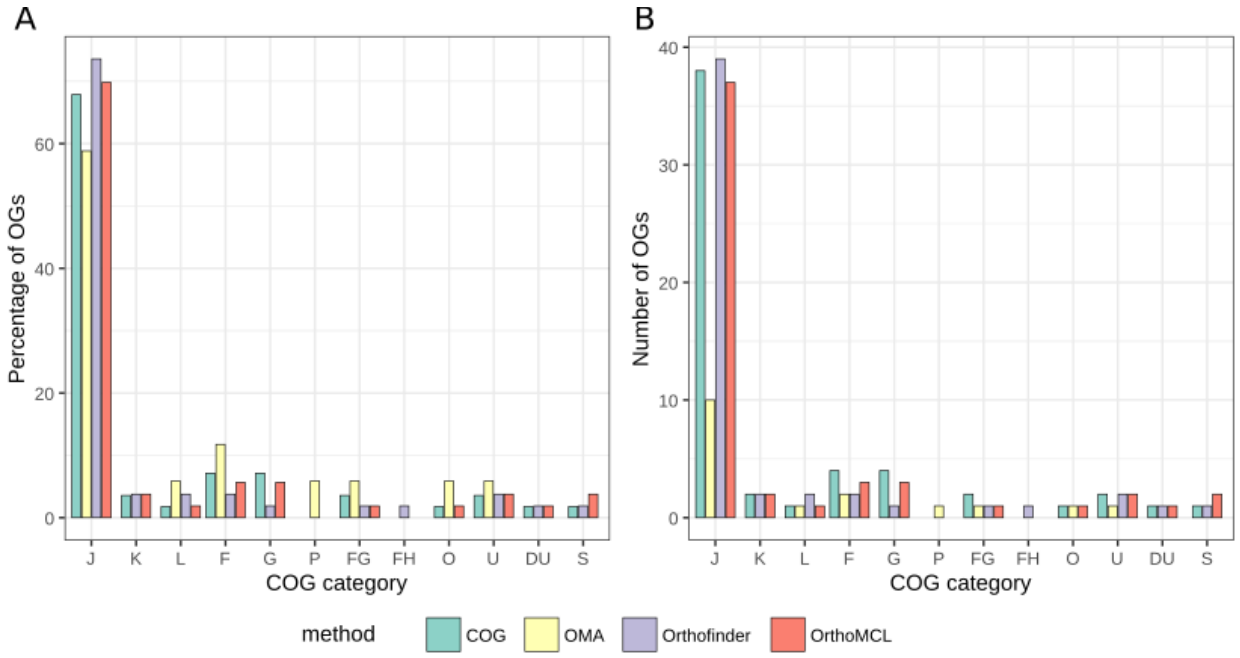


Figure 4. Histogram comparing the proportion of proteins among different COG categories annotated by eggNOG-mapper in four different orthology inference methods.

We can see in Figure 4A that most proteins (between 60 and 70 %) in OGs were annotated with COG category J (Translation, ribosomal structure and modification), encoding for ribosomal proteins and other components of the translation machinery, and this was consistent for all four orthology methods (Figure 4). This is expected for phylogenetic diverse bacteria (as is the case for our group of 10 species), because it is with genes belonging to category J that sequence conservation across widely separated prokaryotic species has been observed to occur [23, 39]. All methods predicted OGs related to replication functions (COG Category L) with the exception of OrthoMCL. OrthoFinder2 was the only one that predicted OGs with proteins associated with transcriptional functions (COG Category K: Transcription).

Although some studies analyzing genomes of bacteria and prokaryotes differ in the number and groups of predicted universal single-copy genes [40–42], it is consistent that the vast majority of OGs were

annotated within COG category J and fewer than a handful are annotated as part of categories L,G, H or O [43].

8. Obtaining orthologs from ortholog databases

In this section, we briefly describe a few resources that offer pre-computed orthologous groups. The text that follows complements and updates the contents of Setubal and Stadler (2018), section 3.2.

Orthology databases vary in the quantity and diversity of the species they represent. Some databases, like eggNOG [44] and OrthoDB [45], cover a wide range of species, including viral sequences, while others, such as TreeFam [46], are more specialized to certain clades.

Orthology resources offer diverse ways of information exploration via Web interfaces for manual inspections essential for routine use by non-experts or using programmatical access interfaces. The search can be performed using the gene name (e.g. *alaS*), GeneID (e.g. 948665) or the protein sequence. Here we will use the protein AlaS that was predicted as a single-copy orthologous group present in all ten species of our dataset to test two orthology resources: OrthoDB v.11 and eggNOG v.6.

According to the UniProtKB database (entry: P00957), AlaS is an alanyl tRNA ligase that catalyzes the attachment of L-alanine to tRNA(Ala). The Gene Ontology annotation for this protein is Aminoacyl-tRNA ligase (Molecular function, GO:0004812) and is involved in the protein biosynthesis process (Biological process, GO:0006412).

8.1. OrthoDB

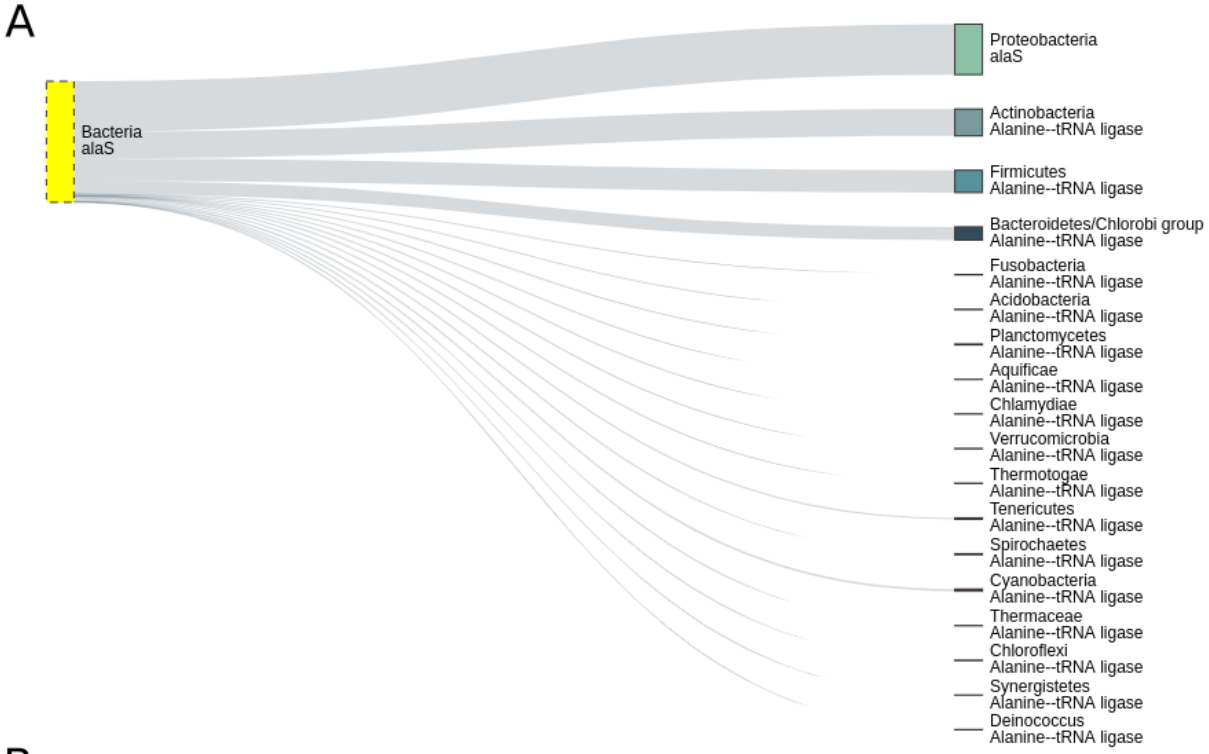
The OrthoDB database contains pre-computed orthology data at various levels of taxonomic distance. The most recent version (OrthoDB v.11) provides analysis and annotation of over 100 million genes, sampling a broad range of species diversity, including prokaryotes (18,158 genomes), eukaryotes (1,973 genomes) and viruses (7,962 genomes). OrthoDB is based on the OrthoLoger software

(https://orthologer.ezlab.org). It relies on the bidirectional best hit method, calculated using the MMseqs2 algorithm [37]. Orthologous genes are clustered following a hierarchical approach guided by a user-provided organism taxonomy. A distinctive feature of OrthoDB is that it also provides evolutionary information for orthologous groups, such as the rate of sequence divergence.

Users can search OrthoDB (https://www.orthodb.org/) by conducting a basic text search, utilizing identifiers from diverse databases, or inputting a protein sequence. When using the term “alaS”, the database yields 612 groups, which correspond to results across various taxonomic levels. At the bacterial level, the alaS group contains 17,747 genes in 17,105 species (out of 17,551 species in the database), which indicates that in-paralogs are included in OGs. The group hierarchy splits OGs into different sub-taxonomic levels, which is displayed with a Sankey flow diagram (Figure 5A). Single-copy genes were identified in 16,501 bacterial species and multi-copy genes in 604 species. We navigate into each of the nine bacterial species of our dataset and confirm the single-copy presence of AlaS in the OrthoDB database. In the search at the archaea level, OrthoDB returns the presence of 1,592 genes in 602 species (out of 607 species in the database), and just 52 species contain single-copy *alaS* gene, including *Methanococcus jannaschii.*

OrthoDB also provides a functional descriptions of the protein family including COG categories, Gene Ontology terms and InterPro domains, as well as evolutionary descriptions including the number of copies per organisms, evolutionary rate, and gene architecture (Figure 5B).

A



B

Functional descriptions

Functional Category	J: Translation, ribosomal structure and biogenesis T: Signal transduction mechanisms L: Replication, recombination and repair K: Transcription
GO Molecular Function	12216 genes with GO:0000166: nucleotide binding 12215 genes with GO:0005524: ATP binding 12214 genes with GO:0004813: alanine-tRNA ligase activity 11972 genes with GO:0016874: ligase activity 11840 genes with GO:0046872: metal ion binding 11811 genes with GO:0008270: zinc ion binding
GO Biological Process	12214 genes with GO:0006419: alanyl-tRNA aminoacylation 11949 genes with GO:0043039: tRNA aminoacylation
GO Cellular Component	12186 genes with GO:0005737: cytoplasm
InterPro Domains	12169 genes with IPR002318: Alanine-tRNA ligase, class IIc 11684 genes with IPR003156: DHHA1 domain 11845 genes with IPR009000: Translation protein, beta-barrel domain superfamily 11937 genes with IPR012947: Threonyl/alanyl tRNA synthetase, SAD 12168 genes with IPR018162: Alanine-tRNA ligase, class IIc, anti-codon-binding domain superfamily 11946 genes with IPR018163: Threonyl/alanyl tRNA synthetase, class II-like, putative editing domain superfamily 12197 genes with IPR018164: Alanyl-tRNA synthetase, class IIc, N-terminal 12194 genes with IPR018165: Alanyl-tRNA synthetase, class IIc, core domain 11912 genes with IPR023033: Alanine-tRNA ligase, eukaryota/bacteria

Evolutionary descriptions

Phyletic Profile	17747 genes in 17105 species (out of 17551) single copy in 16501 species, multi-copy in 604 species
Evolutionary Rate	4.36
Gene Architecture	Median Protein Length 879 (std. 88.1) Median Exon Count 1 (std. 1.86)

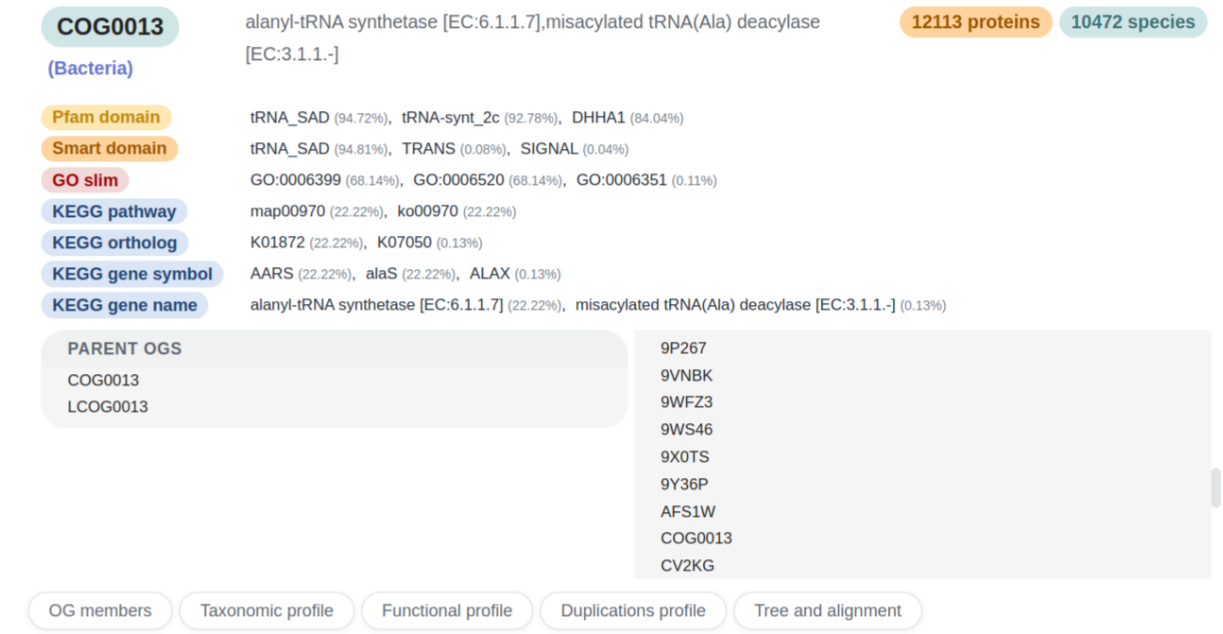
Figure 5. AlaS family at Bacteria level in OrthoDB. A. Interactive group hierarchy split into different taxonomic groups with a Sankey flow diagram. B. Functional annotations of the orthologous group, including COG categories, GO terms, InterPro domains and evolutionary information. Both figures are screenshots of the search results page for AlaS protein at the orthoDB website (https://www.orthodb.org/) [18].

8.2. eggNOG

The eggNOG database offers comprehensive functional information and orthology data for organisms across all domains of life. eggNOG computes all-against-all Smith-Waterman alignments carried out by the SIMAP project [47], and best hits are stored and indexed in a relational database. The current database version (version 6) contains information on 12,535 organisms and 17 million orthologous groups [11].

The user has to first enter a search term (e.g. AlaS or alanyl-tRNA ligase). The OGs matching the query are shown as a list of summary cards, where the most important functional annotations are displayed. Select the card in which the taxonomic level (e.g. Bacteria) the orthologous group should be searched in. The orthogroup (COG0013) of AlaS gene provided by eggNOGG contain 12113 proteins in 10472 bacterial species, which indicates that inparalogs are included in the OG (Figure 6A). The result has different information including functional annotation such as KEGG, Gene Ontology, PFAM and SMART, the list of species with multiple copies of the gene (Duplications profile), and interactive visualization of the taxonomic distribution of OGs and phylogenetic tree (Figure 6B).

A


COG0013
(Bacteria)
alanyl-tRNA synthetase [EC:6.1.1.7],misacylated tRNA(Ala) deacylase [EC:3.1.1.-]
12113 proteins
10472 species
Pfam domain
tRNA_SAD (94.72%), tRNA-synt_2c (92.78%), DHHA1 (84.04%)
Smart domain
tRNA_SAD (94.81%), TRANS (0.08%), SIGNAL (0.04%)
GO slim
GO:0006399 (68.14%), GO:0006520 (68.14%), GO:0006351 (0.11%)
KEGG pathway
map00970 (22.22%), ko00970 (22.22%)
KEGG ortholog
K01872 (22.22%), K07050 (0.13%)
KEGG gene symbol
AARS (22.22%), alaS (22.22%), ALAX (0.13%)
KEGG gene name
alanyl-tRNA synthetase [EC:6.1.1.7] (22.22%), misacylated tRNA(Ala) deacylase [EC:3.1.1.-] (0.13%)
PARENT OGS
COG0013
LCOG0013
9P267
9VNBK
9WFZ3
9WS46
9X0TS
9Y36P
AFS1W
COG0013
CV2KG
OG members
Taxonomic profile
Functional profile
Duplications profile
Tree and alignment

B

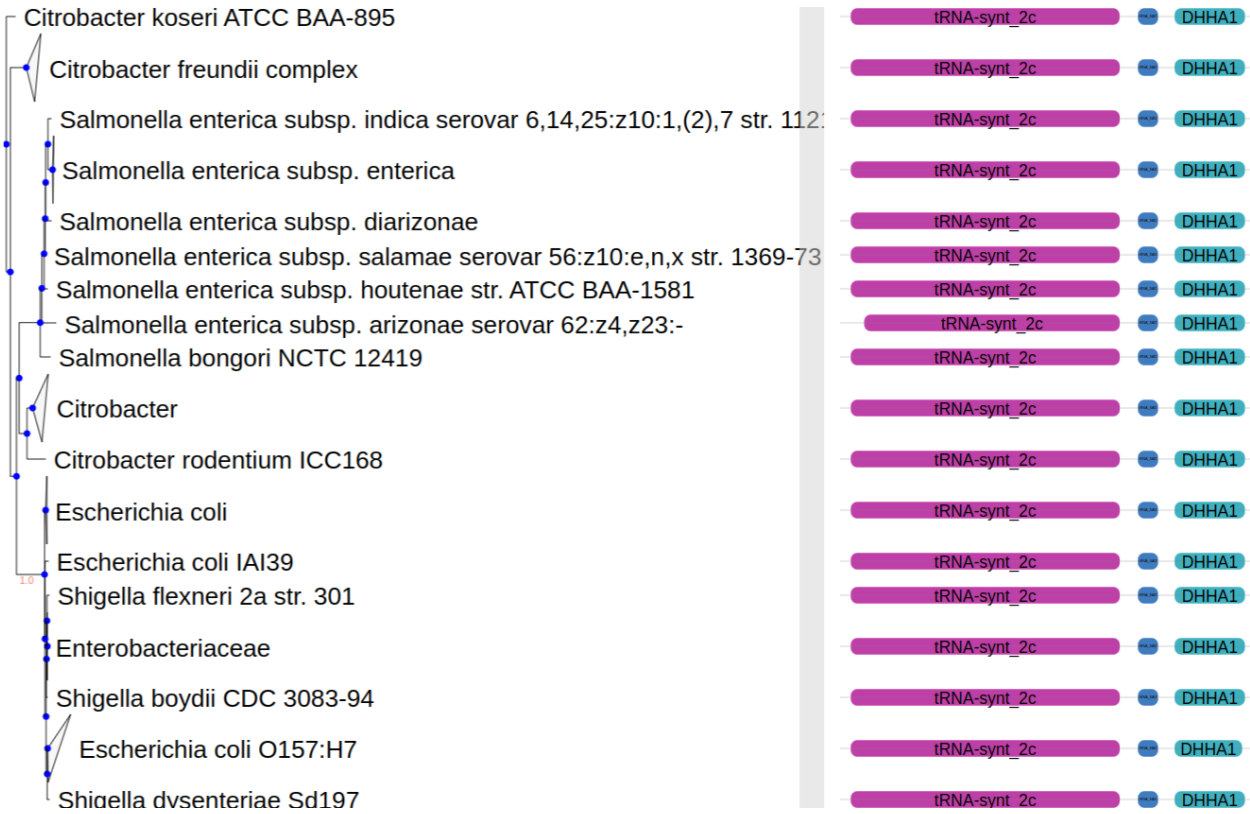

Citrobacter koseri ATCC BAA-895
Citrobacter freundii complex
Salmonella enterica subsp. indica serovar 6,14,25:z10:1,(2),7 str. 112
Salmonella enterica subsp. enterica
Salmonella enterica subsp. diarizonae
Salmonella enterica subsp. salamae serovar 56:z10:e,n,x str. 1369-73
Salmonella enterica subsp. houtenae str. ATCC BAA-1581
Salmonella enterica subsp. arizonae serovar 62:z4,z23:-
Salmonella bongori NCTC 12419
Citrobacter
Citrobacter rodentium ICC168
Escherichia coli
Escherichia coli IAI39
Shigella flexneri 2a str. 301
Enterobacteriaceae
Shigella boydii CDC 3083-94
Escherichia coli O157:H7
Shigella dysenteriae Sd197
tRNA-synt_2c
DHHA1

Figure 6. A. eggNOG summary card of the AlaS orthogroup (COG059) in bacteria domain. A list of functional annotations from different sources (KEGG, GO, etc) is summarized in cards. Expandable cards display detailed information about species members of the orthogroup, taxonomic profile, and interactive visualization tool for phylogenetic trees. B. Interactive visualization of phylogenetic tree indicating speciation or duplication events on nodes coupled to a schematic representation of the gene domain structure. Both figures are screenshots of the search results page for AlaS protein (http://eggnog6.embl.de/) [11].

9. CONCLUSIONS

REFERENCES

1. Fitch WM (1970) Distinguishing homologous from analogous proteins. Syst Zool 19:99–113
2. Altenhoff AM, Studer RA, Robinson-Rechavi M, Dessimoz C (2012) Resolving the Ortholog Conjecture: Orthologs Tend to Be Weakly, but Significantly, More Similar in Function than Paralogs. PLOS Comput Biol 8:e1002514. https://doi.org/10.1371/journal.pcbi.1002514
3. Huerta-Cepas J, Forslund K, Coelho LP, et al (2017) Fast Genome-Wide Functional Annotation through Orthology Assignment by eggNOG-Mapper. Mol Biol Evol 34:2115–2122. https://doi.org/10.1093/molbev/msx148
4. Gabaldón T, Koonin EV (2013) Functional and evolutionary implications of gene orthology. Nat Rev Genet 14:360–366. https://doi.org/10.1038/nrg3456
5. Nevers Y, Defosset A, Lecompte O (2020) Orthology: Promises and Challenges. In: Pontarotti P (ed) Evolutionary Biology—A Transdisciplinary Approach. Springer International Publishing, Cham, pp 203–228
6. Altenhoff AM, Glover NM, Dessimoz C (2019) Inferring Orthology and Paralogy. In: Anisimova M (ed) Evolutionary Genomics: Statistical and Computational Methods. Springer, New York, NY, pp 149–175
7. Fernández R, Gabaldon T, Dessimoz C (2020) Orthology: Definitions, Prediction, and Impact on Species Phylogeny Inference. 2.4:1
8. Wolf YI, Koonin EV (2012) A Tight Link between Orthologs and Bidirectional Best Hits in Bacterial and Archaeal Genomes. Genome Biol Evol 4:1286–1294. https://doi.org/10.1093/gbe/evs100
9. Emms DM, Kelly S (2019) OrthoFinder: phylogenetic orthology inference for comparative genomics. Genome Biol 20:238. https://doi.org/10.1186/s13059-019-1832-y
10. Cantalapiedra CP, Hernández-Plaza A, Letunic I, et al (2021) eggNOG-mapper v2: Functional Annotation, Orthology Assignments, and Domain Prediction at the Metagenomic Scale. Mol Biol Evol 38:5825–5829. https://doi.org/10.1093/molbev/msab293

11. Hernández-Plaza A, Szklarczyk D, Botas J, et al (2023) eggNOG 6.0: enabling comparative genomics across 12 535 organisms. Nucleic Acids Res 51:D389–D394. https://doi.org/10.1093/nar/gkac1022
12. Benson DA, Cavanaugh M, Clark K, et al (2018) GenBank. Nucleic Acids Res 46:D41–D47. https://doi.org/10.1093/nar/gkx1094
13. Contreras-Moreira B, Vinuesa P (2013) GET_HOMOLOGUES, a Versatile Software Package for Scalable and Robust Microbial Pangenome Analysis. Appl Environ Microbiol 79:7696–7701. https://doi.org/10.1128/AEM.02411-13
14. Yu G, Smith DK, Zhu H, et al (2017) ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. Methods Ecol Evol 8:28–36. https://doi.org/10.1111/2041-210X.12628
15. Minh BQ, Schmidt HA, Chernomor O, et al (2020) IQ-TREE 2: New Models and Efficient Methods for Phylogenetic Inference in the Genomic Era. Mol Biol Evol 37:1530–1534. https://doi.org/10.1093/molbev/msaa015
16. Katoh K, Standley DM (2013) MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. Mol Biol Evol 30:772–780. https://doi.org/10.1093/molbev/mst010
17. Altenhoff AM, Levy J, Zarowiecki M, et al (2019) OMA standalone: orthology inference among public and custom genomes and transcriptomes. Genome Res 29:1152–1163. https://doi.org/10.1101/gr.243212.118
18. Kuznetsov D, Tegenfeldt F, Manni M, et al (2023) OrthoDB v11: annotation of orthologs in the widest sampling of organismal diversity. Nucleic Acids Res 51:D445–D451. https://doi.org/10.1093/nar/gkac998
19. Shen W, Le S, Li Y, Hu F (2016) SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation. PLOS ONE 11:e0163962. https://doi.org/10.1371/journal.pone.0163962
20. Capella-Gutiérrez S, Silla-Martínez JM, Gabaldón T (2009) trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. Bioinformatics 25:1972–1973. https://doi.org/10.1093/bioinformatics/btp348
21. Li L, Stoeckert CJ, Roos DS (2003) OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes. Genome Res 13:2178–2189. https://doi.org/10.1101/gr.1224503
22. Setubal JC, Stadler PF (2018) Gene Phylogenies and Orthologous Groups. Methods Mol Biol Clifton NJ 1704:1–28. https://doi.org/10.1007/978-1-4939-7463-4_1
23. Tatusov RL, Koonin EV, Lipman DJ (1997) A Genomic Perspective on Protein Families. Science 278:631–637. https://doi.org/10.1126/science.278.5338.631
24. Galperin MY, Kristensen DM, Makarova KS, et al (2019) Microbial genome analysis: the COG approach. Brief Bioinform 20:1063–1070. https://doi.org/10.1093/bib/bbx117
25. Emms DM, Kelly S (2015) OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. Genome Biol 16:157. https://doi.org/10.1186/s13059-015-0721-2
26. Nevers Y, Jones TEM, Jyothi D, et al (2022) The Quest for Orthologs orthology benchmark service in 2022. Nucleic Acids Res 50:W623–W632. https://doi.org/10.1093/nar/gkac330
27. Zahn-Zabal M, Dessimoz C, Glover NM (2020) Identifying orthologs with OMA: A primer. F1000Research 9:27. https://doi.org/10.12688/f1000research.21508.1
28. Altenhoff AM, Gil M, Gonnet GH, Dessimoz C (2013) Inferring Hierarchical Orthologous Groups from Orthologous Gene Pairs. PLOS ONE 8:e53786. https://doi.org/10.1371/journal.pone.0053786
29. Linard B, Ebersberger I, McGlynn SE, et al (2021) Ten Years of Collaborative Progress in the Quest for Orthologs. Mol Biol Evol 38:3033–3045. https://doi.org/10.1093/molbev/msab098
30. Persson E, Kaduk M, Forslund SK, Sonnhammer ELL (2019) Domainoid: domain-oriented

orthology inference. BMC Bioinformatics 20:523. https://doi.org/10.1186/s12859-019-3137-2
31. Nevers Y, Kress A, Defosset A, et al (2019) OrthoInspector 3.0: open portal for comparative genomics. Nucleic Acids Res 47:D411–D418. https://doi.org/10.1093/nar/gky1068
32. Galperin MY, Wolf YI, Makarova KS, et al (2021) COG database update: focus on microbial diversity, model organisms, and widespread pathogens. Nucleic Acids Res 49:D274–D281. https://doi.org/10.1093/nar/gkaa1018
33. The Gene Ontology Consortium (2019) The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Res 47:D330–D338. https://doi.org/10.1093/nar/gky1055
34. Kanehisa M, Goto S, Sato Y, et al (2014) Data, information, knowledge and principle: back to metabolism in KEGG. Nucleic Acids Res 42:D199–D205. https://doi.org/10.1093/nar/gkt1076
35. Drula E, Garron M-L, Dogan S, et al (2022) The carbohydrate-active enzyme database: functions and literature. Nucleic Acids Res 50:D571–D577. https://doi.org/10.1093/nar/gkab1045
36. Buchfink B, Xie C, Huson DH (2015) Fast and sensitive protein alignment using DIAMOND. Nat Methods 12:59–60. https://doi.org/10.1038/nmeth.3176
37. Steinegger M, Söding J (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol 35:1026–1028. https://doi.org/10.1038/nbt.3988
38. Mistry J, Finn RD, Eddy SR, et al (2013) Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions. Nucleic Acids Res 41:e121. https://doi.org/10.1093/nar/gkt263
39. Koonin EV (2003) Comparative genomics, minimal gene-sets and the last universal common ancestor. Nat Rev Microbiol 1:127–136. https://doi.org/10.1038/nrmicro751
40. Wu M, Eisen JA (2008) A simple, fast, and accurate method of phylogenomic inference. Genome Biol 9:R151. https://doi.org/10.1186/gb-2008-9-10-r151
41. Wu D, Jospin G, Eisen JA (2013) Systematic Identification of Gene Families for Use as “Markers” for Phylogenetic and Phylogeny-Driven Ecological Studies of Bacteria and Archaea and Their Major Subgroups. PLOS ONE 8:e77033. https://doi.org/10.1371/journal.pone.0077033
42. Lan Y, Rosen G, Hershberg R (2016) Marker genes that are less conserved in their sequences are useful for predicting genome-wide similarity levels between closely related prokaryotic strains. Microbiome 4:18. https://doi.org/10.1186/s40168-016-0162-5
43. Wang S, Ventolero M, Hu H, Li X (2022) A revisit to universal single-copy genes in bacterial genomes. Sci Rep 12:14550. https://doi.org/10.1038/s41598-022-18762-z
44. Huerta-Cepas J, Szklarczyk D, Heller D, et al (2019) eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. Nucleic Acids Res 47:D309–D314. https://doi.org/10.1093/nar/gky1085
45. Zdobnov EM, Kuznetsov D, Tegenfeldt F, et al (2021) OrthoDB in 2020: evolutionary and functional annotations of orthologs. Nucleic Acids Res 49:D389–D393. https://doi.org/10.1093/nar/gkaa1009
46. Schreiber F, Patricio M, Muffato M, et al (2014) TreeFam v9: a new website, more species and orthology-on-the-fly. Nucleic Acids Res 42:D922–D925. https://doi.org/10.1093/nar/gkt1055
47. Arnold R, Goldenberg F, Mewes H-W, Rattei T (2014) SIMAP—the database of all-against-all protein sequence similarities and annotations with new interfaces and increased coverage. Nucleic Acids Res 42:D279–D284. https://doi.org/10.1093/nar/gkt970