ARCHE: A HIERARCHICAL, EASY TO DOWNLOAD, FUNCTIONAL-OPTIMIZED ANNOTATOR FOR MICROBIAL META(GENOMES)

Daniel Gonzalo Alonso-Reyes[1] and Virginia Helena Albarracín[1, 2]*

[1]Laboratorio de Microbiología Ultraestructural Y Molecular, Centro Integral de Microscopía Electrónica (CIME,), CONICET-Universidad Nacional de Tucumán, Camino de Sirga s/n, Finca El Manantial, Yerba Buena (4107), San Miguel de Tucumán, Tucumán, Argentina

[2]Facultad de Ciencias Naturales e Instituto Miguel Lillo, Universidad Nacional de Tucumán, Tucumán, Argentina

SUMMARY

High-throughput sequencing advancements, high amounts of genomic data and huge protein databases for homology comparisons have yielded a need for accessible, accurate, and standardized analysis. Genomic annotation tools have revolutionized gene prediction and metabolic reconstruction, and have achieved state-of-the-art accuracy. Despite these advancements, there is no hierarchical pipeline to prioritize the annotation of certain valuable ortholog identifiers and enzyme commission numbers. Such data can be introduced in many tools for metabolic analysis, but currently their annotation is hindered by inaccurate workflows and meaningless information retrieval. This is true especially for those annotators based on very small or too-large protein databases. In this study, we provide a solution based on a hierarchical methodology and optimized databases which additionally overcome data download challenges. The herein presented software framework demonstrates highly optimized and accurate results, and provides an analysis pipeline that can accommodate different search algorithms across prokaryotic genomes, and metagenomes.

1. INTRODUCTION

A crucial step in assessing the metabolic role of a microorganism is the functional gene prediction and metabolic reconstruction [1]. The functional gene data “annotated” by means of bioinformatic tools can be used to infer the metabolic differences with close relatives, characterize novel potential metabolisms, or detect the presence of antibiotic resistance genes or toxins, among other genes of interest [2,3].

Genome annotation is the process of identifying and labeling all the relevant features on a genome sequence [4]. At a minimum, this should include coordinates of predicted coding regions and their putative products, but it is desirable to go beyond by identifying non-coding RNAs, signal peptides and so on. The genome annotation starts with gene identification or gene calling (structural annotation), which can be automatically performed using several tools, including Prodigal and GeneMark among others [5–7]. The next annotation step relies on the use of reference protein databases (DBs) to assign functions to query (unknown) protein sequences based on homology or orthologs searches (functional annotation) [8,9]. Some of the DBs used are comprehensive, including protein sequences derived from complete and partial genomes and are usually updated periodically; widely used examples include the Uniprot [10] and RefSeq [11], InterPro [12], Pfam [13] and SEED [14] DBs. On the other hand, specialized DBs aim at curating the entries to include only protein sequences belonging to specific functions or protein families of interest, e.g., CARD for antibiotic resistance proteins [15].

The methods used to search query protein sequences against the DBs vary in complexity, comprehensiveness, speed, scalability, requirements and outputs. For example, the NCBI Prokaryotic Genome Annotation Pipeline (PGAP) [16], Prokka [17], RAST [18], and DRAM [19] start from genomes and predict genes and proteins, tRNAs, rRNAs, and perform functional annotation of the predicted proteins. Others such as InterProScan [20], EggNOG-Mapper [8] and MicrobeAnnotator [21] start from already predicted protein sequences and perform functional annotations using mostly Blast [22], DIAMOND [23], SWORD [24] or HMMER [25] as search tools. PGAP service requires an account and email, with a turn-around time measured in days. The same happens with RAST, a web server that provides annotation results in under a day [18]. These classes of tools are valuable, but they are not useful where throughput or privacy is critical. On the other hand Prokka, MicrobeAnnotator, DRAM, and EggNOG-Mapper can be run in a desktop computer but they also have disadvantages. Prokka relies on a very short database (DB) and generates a limited number of annotated proteins. At the opposite, MicrobeAnnotator in standard mode demands the download of a huge DB (250-300 GB) which requires a lot of disk space and fast internet connection. EggNOG-Mapper is circumscribed to the eggNOG DB and fails to combine different search algorithms (heuristic, Smith-Waterman [S-W] based and hidden Markov model [HMM] based) in the same run. Finally, DRAM is designed to be run on high-performance computers.

Given the pro and cons of the abovementioned tools, there is a need for an annotation tool that can simultaneously accounts for the following points: i) resolve trait prediction from vast amounts of genomic content; ii) combine diverse DBs and algorithms in a hierarchical way; iii) be launchable from a desktop/laptop; iv) be easy to download. Likewise, the software should produce high informative annotation tables for further metabolic pathway construction, ortholog comparison or singular trait screening. Here we present Arche, a bash-based command-line tool that coordinates a suite of existing software tools to achieve a rich, optimized and reliable annotation of prokaryotic contigs. Arche can take advantage of multiple processing cores to annotate several genomes simultaneously or speed up individual genome annotation.

2. IMPLEMENTATION

2.1. Input

Arche expects preassembled genomic DNA sequences in FASTA format. This sequence file is the only mandatory parameter to the software.

2.2. Annotation process

Arche relies on external feature prediction tools to identify the (meta)genomic features within contigs (Fig. 1). These tools are:

(1) barrnap (https://vicbioinformatics.com/software.barrnap.shtml) for ribosomal RNA genes (rRNA).
(2) tRNAscan-SE 2.0 [26] for transfer RNA genes (tRNA).
(3) GeneMarkS-2 [27] for coding sequences (CDS) in genomes.
(4) MetaProdigal [28] for coding sequences (CDS) in metagenomes.

Once the coordinates of candidate protein genes have been identified, a description is required for the putative gene product. The traditional way to predict what a gene codes for is to compare it with a large DB of known sequences, usually at a protein sequence level, and transfer the annotation of the best significant match.

Arche uses this method, but in a hierarchical manner, starting with a short or medium size set of annotated proteins (depending of the mode), moving to curated HMM profiles of protein families, and finally to a domain-specific HMM database. By default, an e-value threshold of 10^{-8} is used with the following series of included DBs:

(1) A set of informative sequences picked from the UniProtKB DB and clustered in new redesigned DBs (NRDBs). Four options are provided by Arche, including three NRDBs each one biased to the annotation of a certain functional code —Kegg Orthologs (KO [Kanehisa et al., 2016]), eggNOG orthologs (eggNOG [Huerta-Cepas et al., 2019]), or Enzime Comission numbers (E.C. [Bairoch, 2000])— or a trimmed Swiss-Prot DB that annotates the three codes with no bias.
(2) A series of HMM profile DBs (HMMDBs) in this order: a prokaryotic version of KOfam [31], NCBIFAMs [32], TIGRFAMs [33] and Pfam [13]. HMMER3 [25] is used for the search.
(3) If no matches can be found, label as 'hypothetical protein'.

To provide greater flexibility to the user in the querying of the NRDBs clusters, Arche supports the use of three different popular search tools, i.e., BLASTp [22], DIAMOND [23] and SSEARCH [34]. The user can also modify the filtering thresholds used to select the best match found in the DBs (i.e., e-value, or percent alignment across the matching reference protein sequence). Finally, multiprocessing is also supported with the -t (threads) option.

There are two modes of running Arche, “standard” and “heavy”. The “standard” mode uses a reduced Swiss-Prot DB that annotates E.C., KO or eggNOG codes, before starts using HMMDBs. The “heavy” mode has three options; it performs almost the same pattern of the “standard” scanning but replaces Swiss-Prot by a NRDB which prioritizes the annotation of E.C., KO and eggNOG codes, based on the option provided by the user. This NRDB is subdivided into seven sequence clusters. At the top of the scanning are those clusters of sequences which have associated the functional code prioritized (e.g. KO), and at the bottom the remaining clusters. The scanning will proceed through all the seven clusters before starts with HMMDBs. For example, if the code prioritized is E.C., the clusters will be searched in this order: KO + eggNOG + E.C. (cluster 1), KO + EC (cluster 2), eggNOG + EC (cluster 3), EC (cluster 4), KO + eggNOG (cluster 5), KO (cluster 6), and eggNOG (cluster 7), after that HMMDBs will start to run. In case the code prioritized is KO, KOfam scanning will start after the last cluster containing KO. Once KOfam searching is over, the truncated cluster-scanning will be resumed, and then HMMDBs will run (except KOfam). This procedure is intended to maximize the annotation of KOs.

During annotation, query sequences are labeled with the database-specified code of the matching sequence. This code will be used by the pipeline to retrieve the metadata from each of mapping tables implemented in Arche (see Fig. 2). This built-in table summarizes all the functional information (names, KO identifiers, E.C. numbers, etc.) provided by different DBs. However, some data is not coming directly from a database. For example, the Uniprot-associated KO identifiers were obtained through the conversion of the KEGG identifiers (provided by the Uniprot database) to KO identifiers via script. This procedure involves the downloading of the KEGG entry and the finding of the associated KO (Fig. 2). Moreover, the E.C. numbers associated to the Pfam DB were obtained from the ECDomainMiner approach [35].

At the end, all annotations will be compiled in a single table per genome, which includes all metadata associated with each best match.

2.3. Output

Arche produces a single main output folder per run, which contains several files described in Table 1.

3. RESULTS

3.1. Computing Requirements of Arche compared to other tools.

We compared Arche to other popular genome annotation pipelines, including Prokka v1.14.6 [17], RAST [18], EggNOG-mapper v2.1.3 [29], and MicrobeAnnotator light v.2.0.5 (which can recover similar metabolic reconstructions to those obtained using the standard mode [[21]]). The first comparison was in terms of download size of the required DBs to perform an annotation. These DBs include FASTA protein entries, HMM profiles, and mapping files depending on the tool. Table 2 shows that Arche (13 Gb) and eggNOG-mapper (6,4 Gb) required downloads are bigger than the other tools. This difference is mostly driven by the inclusion of the NRDBs, four different HMMDBs used by Arche, and their respective mapping tables. Those DBs are formatted for all Arche’s modes and searching tools, and suitable for the annotation of both bacterial and archaeal genomes, and metagenomes. The Arche's NRDBs include only those sequences associated with E.C./KO/eggNOG data while most of the sequences and map files used by the other programs have not such data associated.

More important than the download size, we compared each tool’s annotation performance in terms of speed, proteins annotated, and level of consistency between tools. For this, we annotated 15 *Exiguobacterium* genomes selected from the NCBI assembly DB (https://www.ncbi.nlm.nih.gov/assembly [Table S1]) with all the aforementioned tools using an E-value threshold of 1e-8. We included in the testing all modes of Arche (“standard”, “kegg”, “ec” and “eggnog”), and we used the blast search option for both Arche and MicrobeAnnotator.

Each genome was annotated independently using 20 threads per run in a Intel(R) Core(TM) i9-10850K computer.

Of the tools tested, Prokka and MicrobeAnnotator (light mode) were the fastest, taking on average 13 and 44 min per set of genomes, respectively (Table 2). These were followed by the standard mode of Arche and eggNOG-mapper which took on average 2.9 h and 3.9 h to annotate the 15 *Exiguobacterium* genomes. In our tests and depending on the server load, the web version of RASTtk can take up to 5.4 h (this includes queuing and processing time). Finally, the "eggnog", "keeg", and "ec" modes of Arche required significantly more time to fully annotate the full set (10.8 h, 12.9 h, and 25,9 h respectively). If a faster annotation is desired, Arche in "kegg" mode using DIAMOND as the search tool can annotate the set in approximately half of the time (6.5 h), without significative loss of advantage related to the number of KO annotations (35734 vs 34685). Accordingly, the use of the standard mode in Arche with DIAMOND takes even less time (2.2 h).

3.2. Annotation quality of Arche compared to other tools

In general, most tools could annotate ~ 70% or more proteins of the set of *Exiguobacterium* genomes, except for Prokka, which annotated ~ 57%, on average. Overall, Arche and eggNOG-mapper annotated the most proteins compared to all other tools (~77-91% and ~84%, respectively). Arche's highest percentage of annotations indicates that the use of several DBs allows for the annotation of hypothetical proteins (or other functions) that could have potential biological importance. Therefore, even if Arche run times are longer than those for the other tools, the resulting annotations have more annotated proteins (Table 2). Moreover, if the DIAMOND option in Arche is used, the runtimes can be reduced even more, without a significant loss in annotation performance (data not shown).

To test how comprehensive and consistent the annotations of each tool are, we quantified the number of matches (labelled protein names) and the presence/absence of functional identifiers in the output, such as E.C. numbers. The Arche's method classifies a protein as annotated if it had at least one match in any of its DBs, which would come with a high chance of getting an associated functional ID (E.C., eggNOG and/or KO). All non-matched sequences (identified as "hypothetical protein" in Arche's output) are the unannotated ones because of lack of protein name and any kind of associated information. In Prokka, RAST and MicrobeAnnotator tests, we used a quite similar logic, considering as annotated all the sequences except those labeled as "hypothetical protein" or "no match found", which in also lacks of any associated information. In the case of eggNOG-mapper, the classification was different: a protein is considered annotated when a match was found, and had at least a name/description or a KO ID.

Out of all the tools tested, RAST had the lowest number of identifiers, including E.C. numbers in ~ 23% of the CDS. Prokka annotations, on the other hand, included E.C. numbers in ~ 32% of the CDS. The remaining tools (MicrobeAnnotator light, EggNOG-Mapper and Arche) consistently included several additional identifiers from multiple DBs to complement the text-based annotation descriptions. The most prevalent of these identifiers were KEGG KO identifiers, which can be used to summarize metabolic potential into pathways and modules. Arche had the highest harvests of KO annotations in all its modes. Particularly, the “kegg” mode run depicted an amazing result of ~76% of CDS annotated with KO identifiers, far away of MicrobeAnnotator light and eggNOG-mapper results (~55% and ~56% respectively). More surprisingly, when run in “eggNOG” and “kegg” modes, Arche collected more eggNOG identifiers (~90% and ~88% respectively) than the eggNOG-mapper tool.

This success of Arche against the other tools in regards to KO annotation have noticeable effects on pathway reconstruction. Those effects are seen when we load the different retrieved sets of KO identifiers from the different tools on the 'Reconstruct tool' of the KEGG mapper online service (https://www.genome.jp/kegg/mapper/reconstruct.html). When filtered by “complete only” modules (Fig. 3), Arche “kegg” mode reconstructed 58 pathway modules, and it was followed by Arche “standard” mode (56 modules), eggNOG-mapper (39 modules) and

MicrobeAnnotator light (11 modules). Surprisingly, Arche “kegg” mode executed with the DIAMOND search aligner showed the same results of the executed with the default (BLASTp).

3.3. Quantitative and qualitative advantages of Arche’s annotations over the other tools.

Arche presents a set of exclusive tools and procedures, while some of them are shared with the other annotators (Fig. 4). Theoretically, Arche should be ahead of the rest in regards to the number of annotation steps, procedure options, or quality of some steps. For example, Arche is a complete genomic annotation pipeline since it can accept as input a genome (complete or fragmented) or even a metagenome. These contigs are going to be undergo structural annotation through the appropriate tools for the prediction of RNAs and proteins. MicrobeAnnotator in its two versions (light and standard) lacks this option and depends on the user directly providing the already predicted protein fasta files. EggNOG-mapper, on the other hand, does not map RNAs and performs protein gene calling only optionally. Regarding to the programs used for structural annotation, both Arche and Prokka use Barrnap (https://github.com/tseemann/barrnap) for the prediction of rRNAs. However, in the case of tRNAs, Arche exploits tRNAscan-SE 2.0 [26], a more updated and improved program than ARAGORN [36] and tRNAscan-SE 1.0 [18,37] both implemented in Prokka and RAST respectively. With regards to the protein gene calling in genomes, Arche implements the amazing GeneMarkS-2 [27] tool which has been demonstrated to give better results than both Prodigal [6] (used by Prokka and optionally by eggNOG-mapper) and Glimmer [38] (used by RAST). Furthermore GeneMarkS-2's output classifies coding regions in 'native' for species-specific genes and 'atypical' for harder-to-detect genes likely (horizontally transferred) [27]. In the case of metagenome annotation, both Arche and Prokka implement the MetaProdigal tool [28] which has been proven to be one of the best algorithms for metagenomic gene prediction. On the other hand, RAST has no option for metagenomic analysis at all.

When it comes to DBs used and search procedures, Arche is complete and accurate. The pipeline has four optional NRDBs made of picked sequences from the UniProtKB DB, and four different HMMDBs which are mandatory. All these DBs are run in a hierarchical strategy, prioritizing the search through sequence alignment, and then applying HMM searches on the remaining sequences. This is based on the concept that HMM profile comparisons are more appropriate for detecting remote homologs [39].

Prokka uses the same logic although with less and shorter DBs. MicrobeAnnotator light mode reverses the procedure, launching an HMM search against the KOfam DB during the first step, and then applying a sequence alignment search against Swiss-Prot over the remaining sequences. EggNOG-mapper fails to combine alignment search methods with HMM searches (they exist as alternative procedures), and RAST doesn't have any HMM search at all. When we consider the search tools used during the sequence alignment step(s), Arche is also the frontrunner. It’s possible to choose between both BLASTp or DIAMOND as the main heuristic algorithms, or the SSEARCH implementation of the S-W algorithm. The options provided by MicrobeAnnotator are similar, although the SWORD tool replaces SSEARCH. However, in contrast to Arche MicrobeAnnotator formats the downloaded DBs for the use of only the chosen one of the three algorithms, which would require a DB redownload in case the user wants to run the annotation with another search tool. EggNOG-mapper has DIAMOND and MMseqs2 as alternatives, having MMseqs2 almost a similar sensitivity than Blast, but still less at the end. Finally, both Prokka and RAST uses only Blast in this step which makes them the least flexible of all the tools tested.

Another Arche’s gift to the user is the retrieval of additional text-based information from the NRDBs and KOfam, like the locus name(s), the alternative name(s) of a protein and text about the feature function whenever available (Fig. 2, 4). This gives the user a great benefit in case of looking for particular features of interest using text search tools.

4. DISCUSSION

We provide a new software tool ‘‘Arche’’ a bash-based command-line tool for annotation of prokaryotic genomes that provides a hierarchical workflow and incorporates state-of-the-art programs for both tRNA and protein prediction. The software demonstrates an innovative segmentation of steps towards a functional-oriented annotation of genes, and a well-organized output of data within one tool. Overall, Arche provides an automated end-to-end solution for the entire annotation of prokaryotic contigs that is both flexible and suitable for extracting the kind of information that the user needs for a chosen downstream analysis. At present, only a handful of groups have developed software that produces sufficient quantity and quality of functional codes to allow adequate metabolic reconstruction. Some of these tools like DRAM demands high-performance computers; other require the download of huge datasets. For example, MicrobeAnnotator standard mode requires the download 200-300 Gb of DB only for a simple blast analysis. Some of the download blocks are very heavy, like the TrEMBL dat files (>150 Gb), thus requiring a fast and uninterrupted connection to be downloaded successfully. These abovementioned tools also rely on DBs which are full of redundant and poorly characterized sequences, a fact that causes penalization on data retrieval because of the lack of data associated to a given match. The major advantages of excluding such useless sequences from a DB is that it can provide a platform to analyze the type of data that the user is more interested in. Thus, by combining the use of functional-oriented DBs and an accurate methodology for genomic annotation, we created a universal tool that can potentially be utilized across different genomic sources (bacteria, archaea and metagenomes) and can annotate functional data in variety of flavors (E.C., KO, and eggNOG numbers/identifiers). We validated this last point by training our program with fifteen *Exiguobacterium* genomes where Arche showed remarkable results among all tools tested (Fig. 3).

Very recently, different annotation approaches were adopted for genomic and metagenomic analysis by our group, including the annotation of metagenomic reads through specific gene-based DBs [40], Prokka (with modified DBs) annotation of genomic contigs [41], and HMMER screening using specific gene profiles [42]. All our previous analysis lacked of a fast, standardized pipeline capable of giving rich outputs of useful metabolic data. That would require the right DBs combined with a carefully picking of all available functional codes (E.C., KO, eggNOG) that such DBs could provide. Arche overcomes the previous limitations by the implementation of NRBDs along with their associated mapping tables (Figs. 1, 2) to provide a more robust and high-throughput analysis of genomes and metagenomes.

Our genomic pipeline incorporates multiple and optional search algorithms, and comprises an update of the novel hierarchical procedures used by Prokka and MicrobeAnnotator for the searching steps. In addition, it employs GeneMarkS-2 to fundamentally learn of the horizontally transferred features across the genome [27]. Arche also has the singularity that it powers the text-based annotations. Because of these features and the other aforementioned, the Arche pipeline can distinguish from all the tested annotators with high precision and robustness for analysis (Figs. 3, 4).

In conclusion, Arche bash-based command-line tool for genomic and metagenomic annotation provides an automated, accurate, and flexible workflow that can achieve high-quality results of use for further metabolic studies. The software versions, extended documentation and examples of usage are provided at https://github.com/gundizalv/Arche.

5. LIMITATIONS OF THE STUDY

This study provides a novel solution for standalone annotation of prokaryotic meta(genomes) which overcomes the challenges of internet connection, is accurate, and produce enriched outputs. However, the proposed pipeline requires many of their programs to be downloaded/installed manually (due to licenses, bad Ubuntu repositories, or specific versions required) which could make the initial setup patience-demanding. Another potential limitation could be that Arche has been tried only on modern versions of Ubuntu (19.04 and 20.04); its behavior in other Linux distributions is unknown to us, and it doesn’t work for Windows. Also, Bash is mandatory to be the default shell, which would require the user to change it in case of using another (e.g., zsh) as default.

6. ACKNOWLEDGMENTS

The authors acknowledge the generous financial support by PICT RAICES 2019–3216 projects. VHA as a Georg Forster Experienced Researcher acknowledges the Alexander von Humboldt Foundation, which has also financially supported this project. VHA is a staff researcher from the National Research Council (CONICET) in Argentina. DA is a postdoctoral researcher at the Center for Electron Microscopy (CIME). This manuscript has been released as a Pre-Print at bioRxiv.

7. AUTHOR'S CONTRIBUTION

DGAR designed the tool, wrote the code, performed the analyses, and wrote the manuscript, VHA provided the hardware for the analysis, and wrote the manuscript. All authors read and approved the final manuscript.

8. REFERENCES

1. Christoffels, A., and van Heusden, P. (2019). Genome annotation: perspective from bacterial genomes. In Encyclopedia of bioinformatics and computational biology, S. Ranganathan, M. Gribskov, K. Nakai, and C. Schönbach, eds. (Oxford: Academic Press), p. 152.

2. Tsementzi, D., Wu, J., Deutsch, S., Nath, S., Rodriguez-R, L.M., Burns, A.S., Ranjan, P., Sarode, N., Malmstrom, R.R., Padilla, C.C., et al. (2016). SAR11 bacteria linked to ocean anoxia and nitrogen loss. Nature *536*, 179–183. 10.1038/nature19068.

3. D., Y.N., Jacobo, de la C.-Z., H., R.G., Silke, D., Nathalie, S., Chris, W., Gabrielle, S., H., F.A., and E., L.R. (2020). Large-Scale Metagenome Assembly Reveals Novel Animal-Associated Microbial Genomes, Biosynthetic Gene Clusters, and Other Genetic Diversity. mSystems *5*, e01045-20. 10.1128/mSystems.01045-20.

4. Richardson, E.J., and Watson, M. (2013). The automatic annotation of bacterial

genomes. Brief. Bioinform. *14*, 1–12. 10.1093/bib/bbs007.

5. Zhu, W., Lomsadze, A., and Borodovsky, M. (2010). Ab initio gene identification in metagenomic sequences . Nucleic Acids Res. *38*, e132–e132. 10.1093/nar/gkq275.

6. Hyatt, D., Chen, G.-L., Locascio, P.F., Land, M.L., Larimer, F.W., and Hauser, L.J. (2010). Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics *11*, 119. 10.1186/1471-2105-11-119.

7. Noguchi, H., Taniguchi, T., and Itoh, T. (2008). MetaGeneAnnotator: Detecting Species-Specific Patterns of Ribosomal Binding Site for Precise Gene Prediction in Anonymous Prokaryotic and Phage Genomes. DNA Res. *15*, 387–396. 10.1093/dnares/dsn027.

8. Huerta-Cepas, J., Forslund, K., Coelho, L.P., Szklarczyk, D., Jensen, L.J., von Mering, C., and Bork, P. (2017). Fast Genome-Wide Functional Annotation through Orthology Assignment by eggNOG-Mapper. Mol. Biol. Evol. *34*, 2115–2122. 10.1093/molbev/msx148.

9. Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., and Tanabe, M. (2016). KEGG as a reference resource for gene and protein annotation. Nucleic Acids Res. *44*, D457–D462. 10.1093/nar/gkv1070.

10. UniProt Consortium, T. (2018). UniProt: the universal protein knowledgebase. Nucleic Acids Res. *46*, 2699. 10.1093/nar/gky092.

11. Haft, D.H., DiCuccio, M., Badretdin, A., Brover, V., Chetvernin, V., O'Neill, K., Li, W., Chitsaz, F., Derbyshire, M.K., Gonzales, N.R., et al. (2018). RefSeq: an update on prokaryotic genome annotation and curation. Nucleic Acids Res. *46*, D851–D860. 10.1093/nar/gkx1068.

12. Mitchell, A.L., Attwood, T.K., Babbitt, P.C., Blum, M., Bork, P., Bridge, A., Brown, S.D., Chang, H.-Y., El-Gebali, S., Fraser, M.I., et al. (2019). InterPro in 2019: improving coverage, classification and access to protein sequence annotations. Nucleic Acids Res.

47, D351–D360. 10.1093/nar/gky1100.

13. El-Gebali, S., Mistry, J., Bateman, A., Eddy, S.R., Luciani, A., Potter, S.C., Qureshi, M., Richardson, L.J., Salazar, G.A., Smart, A., et al. (2019). The Pfam protein families database in 2019. Nucleic Acids Res. *47*, D427–D432. 10.1093/nar/gky995.

14. Overbeek, R., Begley, T., Butler, R.M., Choudhuri, J. V, Chuang, H.-Y., Cohoon, M., de Crécy-Lagard, V., Diaz, N., Disz, T., Edwards, R., et al. (2005). The Subsystems Approach to Genome Annotation and its Use in the Project to Annotate 1000 Genomes. Nucleic Acids Res. *33*, 5691–5702. 10.1093/nar/gki866.

15. G., M.A., Nicholas, W., Fazmin, N., Austin, Y., A., A.M., J., B.A., Kirandeep, B., J., C.M., Gianfranco, D.P., Linda, E., et al. (2013). The Comprehensive Antibiotic Resistance Database. Antimicrob. Agents Chemother. *57*, 3348–3357. 10.1128/AAC.00419-13.

16. Tatusova, T., DiCuccio, M., Badretdin, A., Chetvernin, V., Nawrocki, E.P., Zaslavsky, L., Lomsadze, A., Pruitt, K.D., Borodovsky, M., and Ostell, J. (2016). NCBI prokaryotic genome annotation pipeline. Nucleic Acids Res. *44*, 6614–6624. 10.1093/nar/gkw569.

17. Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. Bioinformatics *30*, 2068–2069. 10.1093/bioinformatics/btu153.

18. Aziz, R.K., Bartels, D., Best, A.A., DeJongh, M., Disz, T., Edwards, R.A., Formsma, K., Gerdes, S., Glass, E.M., Kubal, M., et al. (2008). The RAST Server: Rapid Annotations using Subsystems Technology. BMC Genomics *9*, 75. 10.1186/1471-2164-9-75.

19. Shaffer, M., Borton, M.A., McGivern, B.B., Zayed, A.A., La Rosa, S.L. 0003 3527 8101, Solden, L.M., Liu, P., Narrowe, A.B., Rodríguez-Ramos, J., Bolduc, B., et al. (2020). DRAM for distilling microbial metabolism to automate the curation of microbiome function. Nucleic Acids Res. *48*, 8883–8900. 10.1093/nar/gkaa621.

20. Quevillon, E., Silventoinen, V., Pillai, S., Harte, N., Mulder, N., Apweiler, R., and

Lopez, R. (2005). InterProScan: protein domains identifier. Nucleic Acids Res. *33*, W116–W120. 10.1093/nar/gki442.

21. Ruiz-Perez, C.A., Conrad, R.E., and Konstantinidis, K.T. (2021). MicrobeAnnotator: a user-friendly, comprehensive functional annotation pipeline for microbial genomes. BMC Bioinformatics *22*, 1–16. 10.1186/s12859-020-03940-5.

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

23. Buchfink, B., Xie, C., and Huson, D.H. (2014). Fast and sensitive protein alignment using DIAMOND. Nat. Methods *12*, 59.

24. Vaser, R., Pavlović, D., and Šikić, M. (2016). SWORD—a highly efficient protein database search. Bioinformatics *32*, i680–i684. 10.1093/bioinformatics/btw445.

25. Eddy, S.R. (2011). Accelerated Profile HMM Searches. PLOS Comput. Biol. *7*, e1002195.

26. Chan, P.P., Lin, B.Y., Mak, A.J., and Lowe, T.M. (2021). TRNAscan-SE 2.0: Improved detection and functional classification of transfer RNA genes. Nucleic Acids Res. *49*, 9077–9096. 10.1093/nar/gkab688.

27. Lomsadze, A., Gemayel, K., Tang, S., and Borodovsky, M. (2018). Modeling leaderless transcription and atypical genes results in more accurate gene prediction in prokaryotes. Genome Res. *28*, 1079–1089. 10.1101/gr.230615.117.

28. Hyatt, D., LoCascio, P.F., Hauser, L.J., and Uberbacher, E.C. (2012). Gene and translation initiation site prediction in metagenomic sequences. Bioinformatics *28*, 2223–2230. 10.1093/bioinformatics/bts429.

29. Huerta-Cepas, J., Szklarczyk, D., Heller, D., Hernández-Plaza, A., Forslund, S.K., Cook, H., Mende, D.R., Letunic, I., Rattei, T., Jensen, L.J., 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. 10.1093/nar/gky1085.

30. Bairoch, A. (2000). The ENZYME database in 2000. Nucleic Acids Res. *28*, 304–305. 10.1093/nar/28.1.304.

31. Aramaki, T., Blanc-Mathieu, R., Endo, H., Ohkubo, K., Kanehisa, M., Goto, S., and Ogata, H. (2020). KofamKOALA: KEGG Ortholog assignment based on profile HMM and adaptive score threshold. Bioinformatics *36*, 2251–2252. 10.1093/bioinformatics/btz859.

32. Li, W., O'Neill, K.R., Haft, D.H., DiCuccio, M., Chetvernin, V., Badretdin, A., Coulouris, G., Chitsaz, F., Derbyshire, M.K., Durkin, A.S., et al. (2021). RefSeq: expanding the Prokaryotic Genome Annotation Pipeline reach with protein family model curation. Nucleic Acids Res. *49*, D1020–D1028. 10.1093/nar/gkaa1105.

33. Haft, D.H., Selengut, J.D., Richter, R.A., Harkins, D., Basu, M.K., and Beck, E. (2013). TIGRFAMs and Genome Properties in 2013. Nucleic Acids Res. *41*, D387–D395. 10.1093/nar/gks1234.

34. Lipman, D.J., and Pearson, W.R. (1985). Rapid and Sensitive Protein Similarity Searches. Science (80-.). *227*, 1435–1441. 10.1126/science.2983426.

35. Alborzi, S.Z., Devignes, M.-D., and Ritchie, D.W. (2017). ECDomainMiner: discovering hidden associations between enzyme commission numbers and Pfam domains. BMC Bioinformatics *18*, 107. 10.1186/s12859-017-1519-x.

36. Laslett, D., and Canback, B. (2004). ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. Nucleic Acids Res. *32*, 11–16. 10.1093/nar/gkh152.

37. Lowe, T.M., and Eddy, S.R. (1997). tRNAscan-SE: A Program for Improved Detection

of Transfer RNA Genes in Genomic Sequence. Nucleic Acids Res. *25*, 955–964. 10.1093/nar/25.5.955.

38. Delcher, A.L., Harmon, D., Kasif, S., White, O., and Salzberg, S.L. (1999). Improved microbial gene identification with GLIMMER. Nucleic Acids Res. *27*, 4636–4641. 10.1093/nar/27.23.4636.

39. Madera, M., and Gough, J. (2002). A comparison of profile hidden Markov model procedures for remote homology detection. Nucleic Acids Res. *30*, 4321–4328. 10.1093/nar/gkf544.

40. Alonso-Reyes, D.G., Farias, M.E., and Albarracín, V.H. (2020). Uncovering cryptochrome/photolyase gene diversity in aquatic microbiomes exposed to diverse UV-B regimes. Aquat. Microb. Ecol. *85*, 141–154. 10.3354/AME01947.

41. Alonso-Reyes, D.G., Galván, F.S., Portero, L.R., Alvarado, N.N., Farías, M.E., Vazquez, M.P., and Albarracín, V.H. (2021). Genomic insights into an andean multiresistant soil actinobacterium of biotechnological interest. World J. Microbiol. Biotechnol. *37*, 166. 10.1007/s11274-021-03129-9.

42. Alonso-Reyes, D.G., Galván, S., Irazoqui, J.M., Amadio, A., Tschoeke, D., Thompson, F., Albarracín, V.H., and Farias, M.E. (2022). Dissecting Light Sensing and Metabolic Pathways on the Millimeter Scale in High-Altitude Modern Stromatolites. Microb. Ecol. https://doi.org/10.1007/s00248-022-02112-7.

Figure and Table legends.

Figure 1. Arche workflow. The RNA prediction stage (orange boxes) is optional and only applied to genomic inputs. GeneMarkS-2 and MetaProdigal are used to perform protein prediction (blue boxes) from assembled contigs in genomes and metagenomes respectively. During the first search stage (green boxes), BLASTp, DIAMOND, or SSEARCH can be used to

align the input proteins to four alternative NRBDs (purple containers). Each of those NRBDs are split in several clusters of sequences which have functional information associated (KO identifiers, eggNOG identifiers or Enzime Comission numbers). Except the “standard” NRBD, each one prioritizes the annotation of a particular functional identifier. During the second search stage (grey box) the remanent unannotated proteins serve as input for HMMER which scan over several HMMBDs (cyan containers). “KO” NRBD is interrupted by KOfam DB (cyan container) after the four cluster, which means that all the input sequences unnanotated by the previous clusters would be scanned on KOfam DB using HMMER. After this, the workflow will resume the alignment with the remaining clusters of the “KO” NRBD. The final output of the workflow is a summary table with biological information associated to the matches.

Figure 2. Picture showing the annotation process. Once a group of matches from the search step(s) is defined, they are screened over their corresponding table (colored containers) depending of the DB used in the search. Each table gathers all the useful information provided by the original source (Uniprot, KOfam, etc.). Finally, all the query identifiers, the query matches, and the corresponding related data will be resumed in one single table.

Figure 3. Complete KEGG pathway modules reconstructed through the 'Reconstruct tool' of the KEGG mapper online service (https://www.genome.jp/kegg/mapper/reconstruct.html) using the KO identifiers retrieved from different tools, including two Arche's modes ("kegg" and "standard"). Pinkish blocks indicate the presence of the pathway.

Figure 4. Set of shared and unshared tools and procedures of five different annotators, including Arche.

Table 1. Description of Arche output files.

Table 2. Comparison of annotation of 15 *Exiguobacterium* genomes using five annotators and the four modes of Arche.

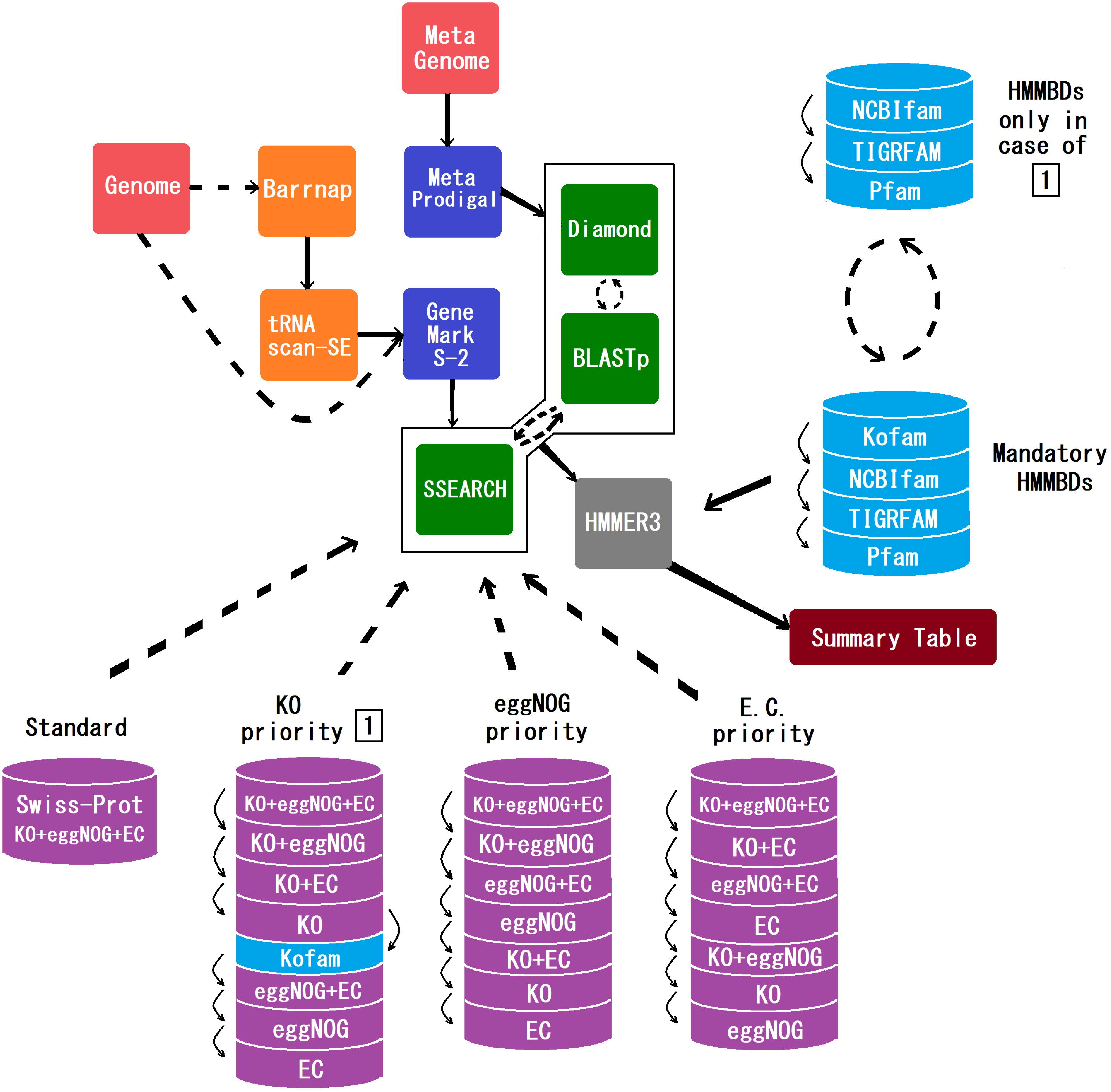
File(s)	Description of file contents
rRNA.tsv	GFF v3 file containing rRNA annotations
rRNA.fna	FASTA file of all rRNA features

tRNA.tsv	Table with tRNA details (coordinates, isotype, anticodon, scores, etc)
tRNA.fna	FASTA file of all tRNA features
[]_struc_annot.fna	FASTA file of all genomic features (nucleotide)
[]_struc_annot.faa	FASTA file of translated coding genes (protein)
heuristic[]_out	Output matches of the search instance(s) performed with BLASTp, DIAMOND or SSEARCH36
heuristic[]_non_match.faa	FASTA file with the remaining non-matched sequences after the search instance(s) performed with BLASTp, DIAMOND or SSEARCH36
hmmscan_[]_out	HMMER3 output table of the search instance(s) performed against a specific hmmdb
[hmmdb]_non_match.faa	FASTA file with the remaining non-matched sequences after the search instance performed against a specific hmmdb
[]_omic_table.tbl	Feature table with fields separated by vertical bars
[]_omic_table.tsv	Feature table with tab-separated fields

Table 1

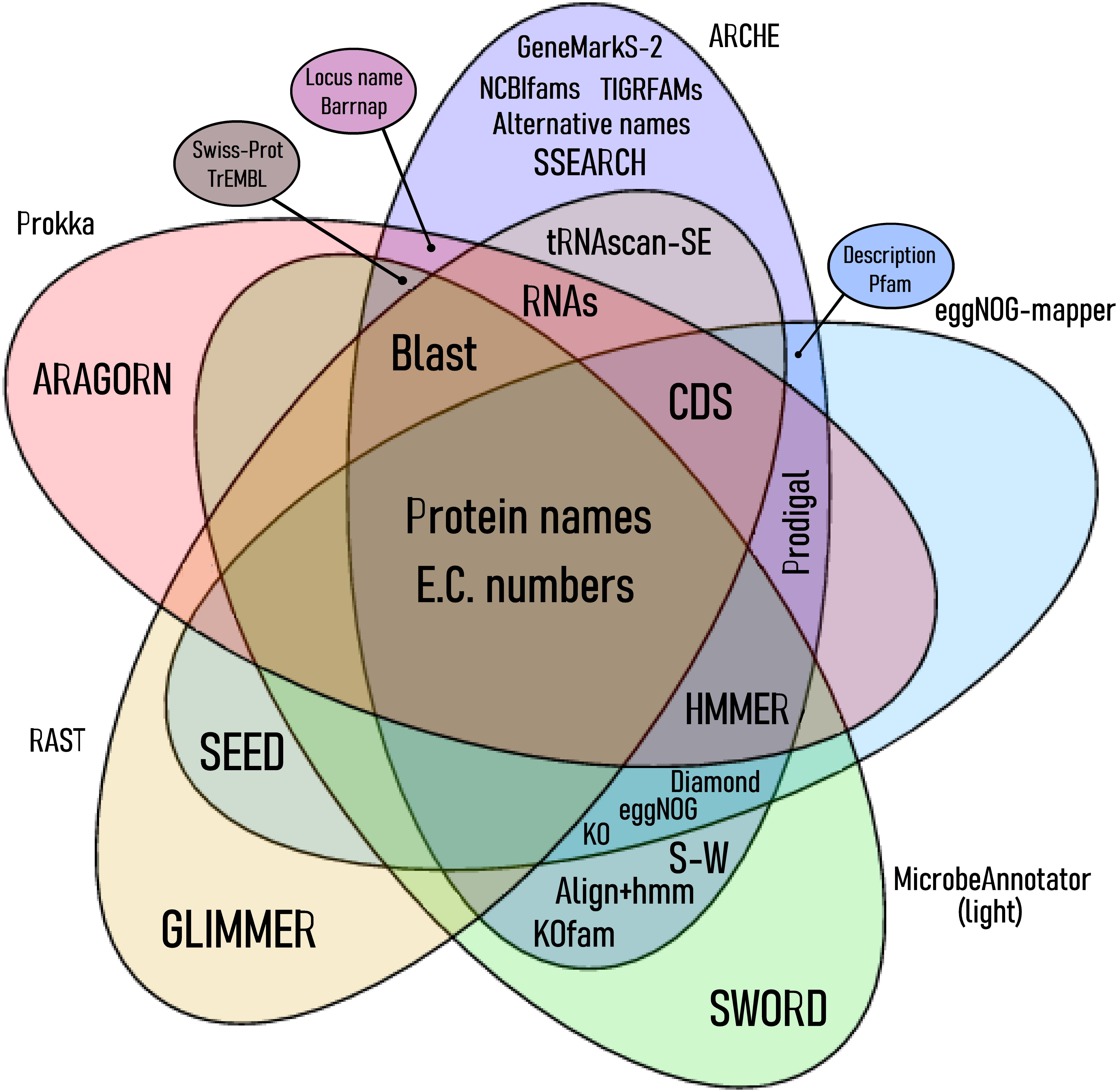
Tool (mode)	CDS	Annotated proteins	Annotated proteins (%)	E.C. annotated	E.C. (%)	KOs annotated	KOs (%)	eggNOGs annotated	eggNOGs (%)	Locus name	Time (s)	Db download size (Gb)
Arche (standard)	46714	36314	77,74	18431	39,45	29170	62,44	30703	65,73	**31586**	10453	13
Arche (kegg)	46714	42496	90,97	27241	58,31	**35734**	**76,50**	41255	88,31	15870	46707	13
Arche (eggNOG)	46714	42472	90,92	27341	58,53	34653	74,18	**42327**	**90,61**	15936	38959	13
Arche (ec)	46714	**42527**	**91,04**	**33456**	**71,62**	31295	66,99	37365	79,99	17601	93595	13
eggNOG-mapper	46903	39710	84,66	12697	27,07	26681	56,89	39710	84,66	0	14047	6,4
Rast	47776	35216	73,71	11288	23,63	0	0,00	0	0,00	0	19560	**0**
MicrobeAnnotator (light)	0*	31985	68,47	16460	35,24	26063	55,79	0	0,00	0	2674	2,11
Prokka	47551	27293	57,40	15350	32,28	0	0,00	0	0,00	25183	**812**	**0**

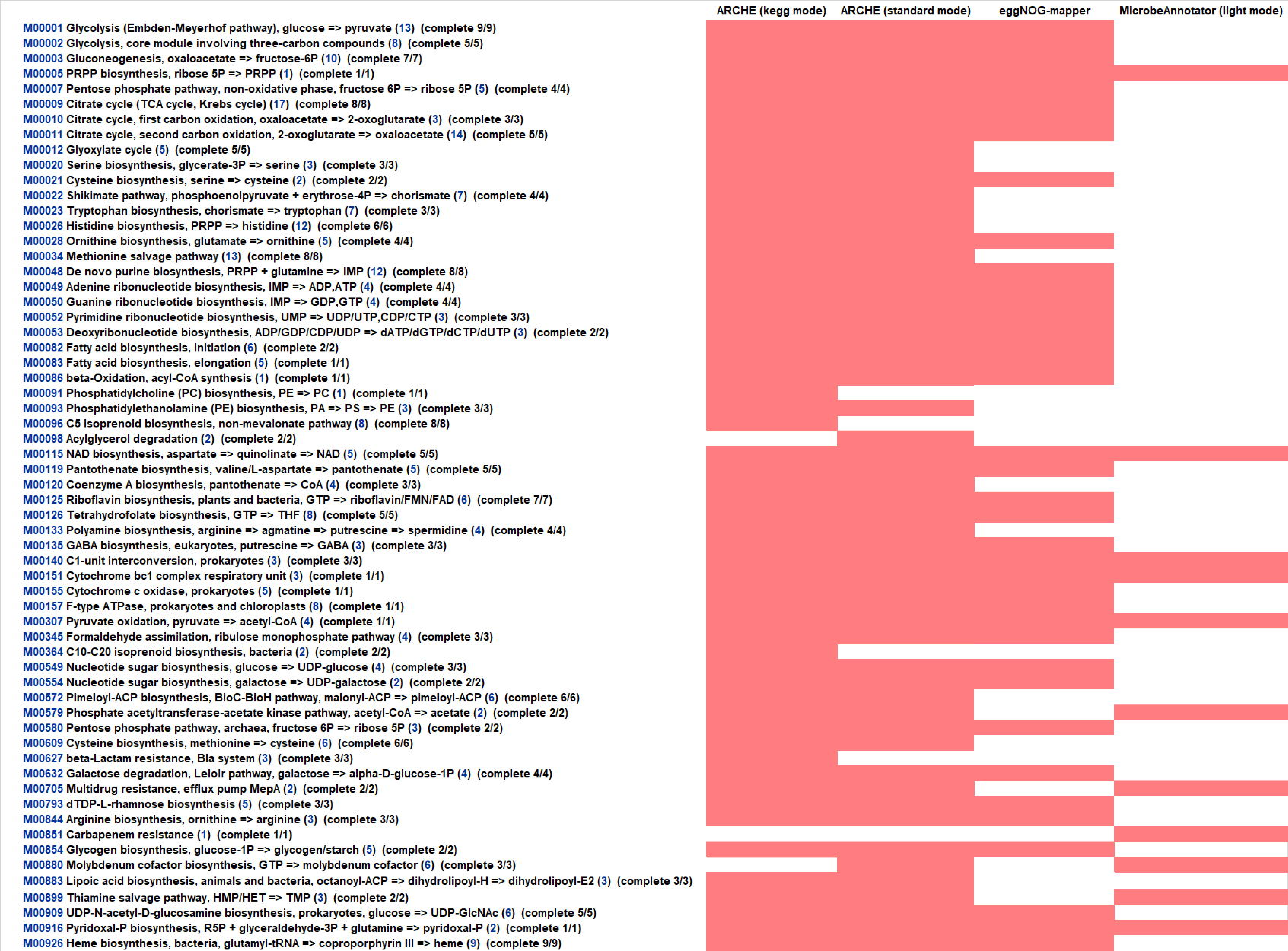
Table 2


Meta Genome
Genome
Barrnap
Meta Prodigal
tRNA scan-SE
Gene Mark S-2
Diamond
BLASTp
SSEARCH
HMMER3
Summary Table
NCBIfam
TIGRFAM
Pfam
HMMBDs only in case of 1
Kofam
NCBIfam
TIGRFAM
Pfam
Mandatory HMMBDs
Standard
Swiss-Prot
KO+eggNOG+EC
KO priority 1
KO+eggNOG+EC
KO+eggNOG
KO+EC
KO
Kofam
eggNOG+EC
eggNOG
EC
eggNOG priority
KO+eggNOG+EC
KO+eggNOG
eggNOG+EC
eggNOG
KO+EC
KO
EC
E.C. priority
KO+eggNOG+EC
KO+EC
eggNOG+EC
EC
KO+eggNOG
KO
eggNOG

Optional NRDBs during search by sequence alignment step(s)

Uniprot entry
Name
AltNam
LocNam
E.C.
eggNOG
Dscrp
KEGG
abcd:A1234_56789
www.genome.jp/entry/abcd:A1234_56789
KO
Uniprot table
KOfam database
Name
AltNam
E.C.
KO
KOfam table
NFAM database
Name
LocNam
E.C.
NFAM table
TIGRFAM database
Name
E.C.
TIGRFAM table
Pfam table
Name
E.C.
Pfam database
ECDomainMiner
List of matches
Table with annotations


ARCHE
GeneMarkS-2
NCBIfams
TIGRFAMs
Alternative names
SSEARCH
Locus name
Barrnap
Swiss-Prot
TrEMBL
Prokka
tRNAscan-SE
Description
Pfam
eggNOG-mapper
RNAs
Blast
ARAGORN
CDS
Prodigal
Protein names
E.C. numbers
HMMER
RAST
SEED
Diamond
eggNOG
KO
S-W
Align+hmm
KOfam
GLIMMER
MicrobeAnnotator
(light)
SWORD



	ARCHE (kegg mode)	ARCHE (standard mode)	eggNOG-mapper	MicrobeAnnotator (light mode)
M00001 Glycolysis (Embden-Meyerhof pathway), glucose => pyruvate (13) (complete 9/9)	🟥	🟥	🟥	
M00002 Glycolysis, core module involving three-carbon compounds (8) (complete 5/5)	🟥	🟥	🟥	
M00003 Gluconeogenesis, oxaloacetate => fructose-6P (10) (complete 7/7)	🟥	🟥	🟥	
M00005 PRPP biosynthesis, ribose 5P => PRPP (1) (complete 1/1)	🟥	🟥	🟥	🟥
M00007 Pentose phosphate pathway, non-oxidative phase, fructose 6P => ribose 5P (5) (complete 4/4)	🟥	🟥	🟥	
M00009 Citrate cycle (TCA cycle, Krebs cycle) (17) (complete 8/8)	🟥	🟥	🟥	
M00010 Citrate cycle, first carbon oxidation, oxaloacetate => 2-oxoglutarate (3) (complete 3/3)	🟥	🟥	🟥	
M00011 Citrate cycle, second carbon oxidation, 2-oxoglutarate => oxaloacetate (14) (complete 5/5)	🟥	🟥	🟥	
M00012 Glyoxylate cycle (5) (complete 5/5)	🟥	🟥		
M00020 Serine biosynthesis, glycerate-3P => serine (3) (complete 3/3)	🟥	🟥		
M00021 Cysteine biosynthesis, serine => cysteine (2) (complete 2/2)	🟥	🟥	🟥	
M00022 Shikimate pathway, phosphoenolpyruvate + erythrose-4P => chorismate (7) (complete 4/4)	🟥	🟥		
M00023 Tryptophan biosynthesis, chorismate => tryptophan (7) (complete 3/3)	🟥	🟥		
M00026 Histidine biosynthesis, PRPP => histidine (12) (complete 6/6)	🟥	🟥		
M00028 Ornithine biosynthesis, glutamate => ornithine (5) (complete 4/4)	🟥	🟥	🟥	
M00034 Methionine salvage pathway (13) (complete 8/8)	🟥	🟥		
M00048 De novo purine biosynthesis, PRPP + glutamine => IMP (12) (complete 8/8)	🟥	🟥	🟥	
M00049 Adenine ribonucleotide biosynthesis, IMP => ADP,ATP (4) (complete 4/4)	🟥	🟥	🟥	
M00050 Guanine ribonucleotide biosynthesis, IMP => GDP,GTP (4) (complete 4/4)	🟥	🟥	🟥	
M00052 Pyrimidine ribonucleotide biosynthesis, UMP => UDP/UTP,CDP/CTP (3) (complete 3/3)	🟥	🟥	🟥	
M00053 Deoxyribonucleotide biosynthesis, ADP/GDP/CDP/UDP => dATP/dGTP/dCTP/dUTP (3) (complete 2/2)	🟥	🟥	🟥	
M00082 Fatty acid biosynthesis, initiation (6) (complete 2/2)	🟥	🟥	🟥	
M00083 Fatty acid biosynthesis, elongation (5) (complete 1/1)	🟥	🟥	🟥	
M00086 beta-Oxidation, acyl-CoA synthesis (1) (complete 1/1)	🟥	🟥	🟥	
M00091 Phosphatidylcholine (PC) biosynthesis, PE => PC (1) (complete 1/1)	🟥			
M00093 Phosphatidylethanolamine (PE) biosynthesis, PA => PS => PE (3) (complete 3/3)	🟥	🟥		
M00096 C5 isoprenoid biosynthesis, non-mevalonate pathway (8) (complete 8/8)	🟥			
M00098 Acylglycerol degradation (2) (complete 2/2)		🟥		
M00115 NAD biosynthesis, aspartate => quinolinate => NAD (5) (complete 5/5)	🟥	🟥	🟥	🟥
M00119 Pantothenate biosynthesis, valine/L-aspartate => pantothenate (5) (complete 5/5)	🟥	🟥	🟥	
M00120 Coenzyme A biosynthesis, pantothenate => CoA (4) (complete 3/3)	🟥	🟥		
M00125 Riboflavin biosynthesis, plants and bacteria, GTP => riboflavin/FMN/FAD (6) (complete 7/7)	🟥	🟥	🟥	
M00126 Tetrahydrofolate biosynthesis, GTP => THF (8) (complete 5/5)	🟥	🟥	🟥	
M00133 Polyamine biosynthesis, arginine => agmatine => putrescine => spermidine (4) (complete 4/4)	🟥	🟥		
M00135 GABA biosynthesis, eukaryotes, putrescine => GABA (3) (complete 3/3)	🟥	🟥	🟥	
M00140 C1-unit interconversion, prokaryotes (3) (complete 3/3)	🟥	🟥	🟥	🟥
M00151 Cytochrome bc1 complex respiratory unit (3) (complete 1/1)	🟥	🟥	🟥	🟥
M00155 Cytochrome c oxidase, prokaryotes (5) (complete 1/1)	🟥	🟥	🟥	
M00157 F-type ATPase, prokaryotes and chloroplasts (8) (complete 1/1)	🟥	🟥	🟥	
M00307 Pyruvate oxidation, pyruvate => acetyl-CoA (4) (complete 1/1)	🟥	🟥	🟥	🟥
M00345 Formaldehyde assimilation, ribulose monophosphate pathway (4) (complete 3/3)	🟥	🟥	🟥	
M00364 C10-C20 isoprenoid biosynthesis, bacteria (2) (complete 2/2)	🟥			
M00549 Nucleotide sugar biosynthesis, glucose => UDP-glucose (4) (complete 3/3)	🟥	🟥	🟥	
M00554 Nucleotide sugar biosynthesis, galactose => UDP-galactose (2) (complete 2/2)	🟥	🟥	🟥	
M00572 Pimeloyl-ACP biosynthesis, BioC-BioH pathway, malonyl-ACP => pimeloyl-ACP (6) (complete 6/6)	🟥	🟥		
M00579 Phosphate acetyltransferase-acetate kinase pathway, acetyl-CoA => acetate (2) (complete 2/2)	🟥	🟥		🟥
M00580 Pentose phosphate pathway, archaea, fructose 6P => ribose 5P (3) (complete 2/2)	🟥	🟥	🟥	
M00609 Cysteine biosynthesis, methionine => cysteine (6) (complete 6/6)	🟥	🟥		
M00627 beta-Lactam resistance, Bla system (3) (complete 3/3)	🟥			
M00632 Galactose degradation, Leloir pathway, galactose => alpha-D-glucose-1P (4) (complete 4/4)	🟥	🟥	🟥	
M00705 Multidrug resistance, efflux pump MepA (2) (complete 2/2)	🟥	🟥		🟥
M00793 dTDP-L-rhamnose biosynthesis (5) (complete 3/3)	🟥	🟥	🟥	
M00844 Arginine biosynthesis, ornithine => arginine (3) (complete 3/3)	🟥	🟥	🟥	
M00851 Carbapenem resistance (1) (complete 1/1)				🟥
M00854 Glycogen biosynthesis, glucose-1P => glycogen/starch (5) (complete 2/2)	🟥	🟥	🟥	
M00880 Molybdenum cofactor biosynthesis, GTP => molybdenum cofactor (6) (complete 3/3)		🟥		🟥
M00883 Lipoic acid biosynthesis, animals and bacteria, octanoyl-ACP => dihydrolipoyl-H => dihydrolipoyl-E2 (3) (complete 3/3)	🟥	🟥		
M00899 Thiamine salvage pathway, HMP/HET => TMP (3) (complete 2/2)	🟥	🟥		🟥
M00909 UDP-N-acetyl-D-glucosamine biosynthesis, prokaryotes, glucose => UDP-GlcNAc (6) (complete 5/5)	🟥	🟥	🟥	
M00916 Pyridoxal-P biosynthesis, R5P + glyceraldehyde-3P + glutamine => pyridoxal-P (2) (complete 1/1)	🟥	🟥	🟥	🟥
M00926 Heme biosynthesis, bacteria, glutamyl-tRNA => coproporphyrin III => heme (9) (complete 9/9)	🟥	🟥	🟥	