Comparative genomics for evolutionary cell biology using AMOEBAE: Understanding the Golgi and beyond

Lael D. Barlow[1,2*], William Maciejowski[1], Kiran More[1], Kara Terry[3], Romana Vargová[4], Kristína Záhonová[5,6] & Joel B. Dacks[1,3,5,7*]

[1] Department of Biological Sciences, University of Alberta, Edmonton, Alberta, Canada
[2] Division of Biological Chemistry and Drug Discovery, School of Life Sciences, University of Dundee, Dundee, UK
[3] Division of Infectious Diseases, Department of Medicine, University of Alberta, Edmonton, Alberta, Canada
[4] Department of Biology and Ecology, Faculty of Science , University of Ostrava, Czechia
[5] Department of Parasitology, Faculty of Science, Charles University, BIOCEV, Czechia
[6] Institute of Parasitology, Biology Centre, Czech Academy of Sciences, Czechia

[7] Centre for Life's Origin and Evolution, Department of Genetics, Evolution and Environment, University College of London, United Kingdom

*Correspondence concerning this article should be addressed to Lael D. Barlow, School of Life Sciences, University of Dundee, Dundee DD1 5EH, UK. E-mail: lbarlow001@dundee.ac.uk

or Joel B. Dacks, Division of Infectious Diseases, University of Alberta, 1-124 Clinical Sciences Building, 11350-83 Avenue, Edmonton, CANADA, T6G 2G3. Email:dacks@ualberta.ca

Abstract

Taking an evolutionary approach to cell biology can yield important new information about how the cell works and how it evolved to do so. This is true of the Golgi apparatus, as it is of all systems within the cell. Comparative genomics is one of the crucial first steps to this line of research, but comes with technical challenges that must be overcome for rigor and robustness. We here introduce AMOEBAE, a workflow for mid-range scale comparative genomic analyses. It allows for customization of parameters, queries, and taxonomic sampling of genomic and transcriptomics data. This protocol article covers the rationale for an evolutionary approach to cell biological study (i.e. when would AMOEBAE be useful), how to use AMOEBAE, and discussion of limitations. It also provides an example dataset, which demonstrates that the Golgi protein AP4 Epsilon is present as the sole retained subunit of the AP4 complex in basidiomycete fungi. AMOEBAE can facilitate comparative genomic studies by balancing reproducibility and speed with user-input and interpretation. It is hoped that AMOEBAE or similar tools will encourage cell biologists to incorporate an evolutionary context into their research.

Keywords: Comparative genomics, Homology searching, BLAST, Evolutionary Cell Biology, Golgi, Basidiomycete, Adaptin, Molecular evolution, Workflow, Computational pipeline

1. Introduction

At its most aspirational, the field of cell biology strives to uncover fundamental mechanisms of cellular function. It is hoped that discoveries apply not only to the cell type in which they are made but more generally, perhaps even universally. This is the basis for the use of model systems including *Drosophila*, yeast, or *Caenorhabditis elegans*, and as such cell biology is inherently comparative. While this assumption of universality of cellular discoveries has largely been implicit, with the advent of large-scale genomic sequencing across the diversity of eukaryotes, the advances in the field of molecular evolution, and the development of model organisms from distantly related lineages, the assumption is now possible to test explicitly.

Combining molecular evolution and molecular cell biology is one facet of the emerging field of Evolutionary Cell Biology (ECB) [1] and allows questions to be addressed from several different perspectives. Comparative genomics can identify proteins of unknown function that are present in organisms across the span of eukaryotic diversity. These proteins can then be characterized by molecular cell biology in model systems to determine the cellular systems within which they act (*e.g.*, the recently described mitochondrial secretion system [2]). Reciprocally, the evolutionary distributions of a newly identified component from a model system can be assessed using comparative genomics to determine how generally or narrowly it can be incorporated into models of cell biology (*e.g.*, the new GBF1 binding protein C10orf76 [3] or the Retriever complex [4]). These questions relate to modern cellular function, but an ECB approach also addresses evolutionary perspectives. The components discovered through molecular cell biology can be assessed by molecular evolutionary techniques to reveal details of our cellular history from its deepest origins through transitions as the various eukaryotic lineages have adapted to ecological niches and survival strategies [5, 6].

1.1 Evolution of MTS and Golgi apparatus

One area that has fruitfully benefitted from this evolutionary approach to cell biology is that of membrane-trafficking. Examples abound of components discovered in model systems that have turned out to be found across eukaryotes, supporting their inclusion in general models of membrane-trafficking (*e.g.*, [7, 8]). The same is true for examples of components found bioinformatically to be more limited in scope and thus applicability (*e.g.*, Caveolin [9]). The Endosomal Sorting Complex Required for Transport involved in multivesicular body formation alone has components with both distributions [10]. Examples also exist where taking a molecular evolutionary approach identified a protein or complex which later turned out to be relevant across the span of eukaryotes [11]. In one such example, the discovery of the Adaptor Protein 5 (AP5) complex [12] led to the designation of a new type of lysosomal storage disorder in humans [13]. All together these ECB studies have also provided a detailed picture of the evolution of the membrane-trafficking machinery, with a sophisticated system being present in the Last Eukaryotic Common Ancestor (LECA). Both expansions and losses of ancestral gene families, sometimes in combination within a lineage, have occured as the descendants of the LECA evolved into the myriad forms of eukaryotes that we see today (reviewed in [14, 15]).

The Golgi is at the center of the eukaryotic membrane-trafficking system, and is one of the more recognizable organelles in the cell, with its hallmark stack of pita bread or pancakes morphology. As this feature can be readily observed in organisms from across the tree of eukaryotes [16, 17], it is clear that the organelle is ancient and well conserved. This is corroborated by the conservation across eukaryotes of many proteins known to function in vesicle trafficking at the Golgi, most of which were initially identified in the metazoan and yeast model systems [18]. And yet there are some lineages in which canonical Golgi stacks are not visually apparent. This initially led to the suggestion that the organelle may have evolved later in eukaryotic evolution, with these 'Golgi-lacking' lineages diverging prior to the organelle's emergence [19]. Both molecular evolutionary and molecular cell biological data refuted this early idea [20–22]. The outcome, however, was something suspected even from the early studies in yeast: the Golgi body can take on a much broader range of morphologies than was previously considered. There are simplified forms such as distributed single cisternae as seen in yeast [23] or Entamoebae [22], reticulated networks as seen in the heterolobosean *Naegleria* [24], vesicles as seen in some ciliates [25], all the way up to massively expanded networks possessing 10-100s of stacks and 1000s of cisternae per stack as seen in some parabasalid flagellates [26].

With the tremendous advances in molecular understanding of cell biological systems, and the massive influx of genomic data, ECB analyses addressing questions of every cellular system have become at the same time more conceptually attainable and more technically challenging. Additional sampling points make for better designed and more theoretically robust analyses, but also increase the scale of the project. Not all proteins researchers wish to investigate are going to be well-conserved or easily identified, nor will all of the genomes of interest be easy to interrogate, due to rapid rates of genomic evolution. This means that for a robust conclusion to be reached, one-step, easy searches are insufficient. Instead rigorous and reliable analysis involves taking several different approaches, with more than one starting point and probing different databases with multiple similarity search algorithms. For example, in our recent investigation of Golgi evolution [27], we asked whether proteins implicated in Golgi-stacking were conserved across eukaryotes and what that meant for the complexity of the Golgi in the LECA. We found that nine such proteins (which in model systems act at cis, medial, and Trans-Golgi Network (TGN)) were identified in diverse eukaryotes, and thus the LECA was deduced to have possessed a differentiated Golgi organelle [27]. This analysis involved application of a custom set of search criteria to identify homologues of 27 proteins with diverse biological function and evolutionary histories across 88 genomes. The challenges inherent to this project indeed prompted the development of the informatic pipeline that we are presenting here.

1.2 Why AMOEBAE?

As with any area of technical proficiency, comparative genomics involves both specific skills and subject-wide expertise to design, implement and interpret analyses. Because not all proteins evolve in the same way or at the same rate, one-size fits all criteria will fail to capture the subtleties of the real-world data in many cases and care needs to be taken to make reliable inferences. As such comparative genomics can be both time-consuming and subject to human error, particularly because of the repetitive nature of some tasks. At the same time, full automation of data interpretation throws away the expertise of the investigator who has carefully designed the study and knows their system in detail.

Simple linux/unix shell or Python scripts are a standard tool in the toolbox of research labs that do evolutionary analyses of genomic data. These are sometimes taken for granted and referred to as “custom in-house scripts”. For over a decade, routine tasks in the Dacks lab have been performed using many renditions of such scripts. However, as genomic data, bioinformatics methods, and scientific knowledge accumulates, simple scripts have become inadequate.

We are introducing AMOEBAE (**A**nalysis of **MO**lecular **E**volution with **BA**tch **E**ntry), a reproducible semi-automated workflow to allow for efficient homology searching with detailed summaries for in-depth verification of quality control and follow-up analysis at key steps. This vastly improves the efficiency of analysis by removing user-driven data input and porting results to different analytical steps, while allowing user quality checking at several intermediate stages of the analysis. The AMOEBAE workflow is the result of gradual development by progressive addition of code to perform homology searching tasks in a similar manner to how they were done “manually” or “visually”, but in a more standardized and reproducible manner. In practice, we have found that AMOEBAE can accomplish in a matter of minutes or hours what would take weeks or months without it. Also, the ability to reproduce complex analyses automatically and to re-run analyses with modified parameters is indispensable.

This protocol paper is meant primarily for cell biologists who wish to investigate their system of interest from an evolutionary perspective and would like tools more sophisticated than are available from simple graphical user interfaces such as BLAST at NCBI. Since evolutionary biologists have also shown interest in AMOEBAE, this paper also serves as the publication of the method and links to the live Github resource that will be maintained for the tool. This paper both walks the reader through the use of this novel tool, and integrates the experience of two types of users, relative novices (i.e. summer undergraduate researchers with no prior molecular evolutionary experience) and more experienced graduate students or postdoctoral fellows. All of these co-authors specifically validated usability of this workflow in the protocol defined below by each running the example dataset and providing feedback to improve the workflow's usability, as well as field-testing the final version on their own unpublished datasets to identify areas where particular attention needs to be paid because of the nature of the question being addressed. This article will talk through the planning stages, protocol, common challenges users faced, and interpretation of the data.

1.3 Applicability and research questions

AMOEBAE is a customizable bioinformatics workflow for identifying homologues (and potential orthologues) of genes of interest among a mid-size sampling of genomes. This workflow is designed to be run on high-performance computing clusters (HPCs) by researchers with some prior experience with comparative genomics methods. Identifying homologous genes is an essential and common goal of many software packages available to biologists, and selecting the right software and protocol for a particular project is key to success.

For most evolutionary analyses, AMOEBAE will not and should not be the first port of call. In many cases, information already available from online databases or obtainable *via* less time-intensive methods may be sufficient or may be important for directing a research project. For example, pre-computed orthology predictions are available from databases such as EggNOG-Mapper [28, 29], and there are a variety of curated databases such as InterPro [30]. Often when novel analysis is required, webservices such as those supplied by the NCBI (https://blast.ncbi.nlm.nih.gov/Blast.cgi) and the European Molecular Biology Laboratory-European Bioinformatics Institute (EMBL-EBI) (https://www.ebi.ac.uk/Tools/hmmer/) provide a means to readily investigate the evolution of one or a few genes via similarity searching. Generic large-scale analysis workflows such as OrthoMCL [31], SonicParanoid [32], OrthoFinder [33] and others attempt to rapidly perform orthology prediction for all genes among several genomes.

AMOEBAE is instead suited for analyses which are too cumbersome to be performed *via* webservices or simple scripts and yet require a level of detail and flexibility not offered by generic large-scale analysis workflows. For example, this might involve analyzing the distribution of homologues of up to approximately 30 genes/proteins among a sampling of no more than approximately 100 eukaryotic genomes. In other words, AMOEBAE is useful for mid-scale comparative genomics studies that might otherwise require time-intensive and *overly repetitive* visual interpretation and manual manipulation of data, which is very difficult to reproduce. AMOEBAE also allows results to be adjusted based on follow-up sequence analyses.

This can be achieved by refining search parameters and re-running the AMOEBAE workflow, or by manually adjusting the final result spreadsheet before plotting.

We have selected a simple example to illustrate the utility of the AMOEBAE workflow: The evolution of the AP-4 complex in Basidiomycota (Fungi). In mammalian cells, the AP-4 complex localizes to the TGN, where it functions in export of cargo from the TGN to other membranes in the cell [34, 35], including roles in autophagy [36]. Previous studies on the evolution of AP complexes in fungi have noted losses of genes encoding subunits of the AP4 complex, but curiously, presence of a lone subunit of the complex, the epsilon subunit, in the basidiomycete *Cryptococcus neoformans* [37, 38]. The sister group to Basidiomycota, the ascomycetes (including *Saccharomyces cerevisiae*), lack the AP4 complex altogether, while the more distantly related glomeromycete *Rhizophagus irregularis* has retained all four subunits of AP4 [37]. This distribution of genes encoding AP4 subunits implies multiple gene loss events in the evolution of Basidiomycota. However, this could be explained by many alternative hypotheses regarding the details of when such gene loss events occured. For example, it could be that many basidiomycetes retain complete AP4 complexes and the absence of some in *C. neoformans* and all from other basidiomycetes such as *Ustilago maydis* are a consequence of recent gene losses. On the other hand, *C. neoformans* may truly be exceptional among basidiomycetes, being the lone representative to retain any AP4 subunit. Identification of a lone AP4 subunit in a single basidiomycete also raises the possibility of false positive or negative results due to gene prediction errors or sequence divergence, or invites more controversial evolutionary explanations such as acquisition *via* horizontal gene transfer. As we describe below, AMOEBAE is applicable for addressing these hypotheses, by searching in a larger sampling of basidiomycete genomes using sensitive search methods.

1.5 Prerequisite skills

In practice we have found that several basic computing skills are required for researchers to make use of AMOEBAE effectively. In particular, these include navigation and manipulation of file systems as well as running software on HPCs *via* the linux/unix command-line. These skills are the bioinformatic equivalent of benchwork skills such as pipetting. However, these skills are generally not included in university biology curricula, and this gap in training is usually addressed by accessing other resources such as those offered by the Software Carpentry Foundation [39]. As with other disciplines such as microscopy, some skills are generally applicable while others will be specific to the resources available at different institutions. Thus it will be important to consult with your local system administrators.

Beyond basic computing skills, users must understand several background concepts of sequence analysis. First, it is critical to understand the completeness and quality of the genomic (or transcriptomic) data selected for input to AMOEBAE. Common issues with eukaryotic genome assemblies include incompleteness, false segmental duplications, and incomplete or otherwise inaccurate coding sequence predictions (gene models). These types of errors cause AMOEBAE to output false-negative or false-positive search results due to missing data, assembly artefacts, or fragmented gene sequences. Quality control for genomic data is beyond the scope of this protocol and has been reviewed elsewhere [40, 41].

Second, the user of AMOEBAE must understand how to infer homology of genes (and predicted amino acid sequences) based on sequence similarity. This is a foundational skill with many applications, and has been discussed extensively elsewhere [42]. This tends to involve working with sequence files in FASTA format as well as generating and scoring sequence alignments with BLASTp, tBLASTn, and HMMER [43, 44]. While the aim of this protocol is to retrieve sequences from a database with considerable similarity to query sequences (i.e., similarity searching), we must emphasize that it is entirely the responsibility of the individual biologist applying the protocol to evaluate the results and to determine whether they support hypotheses regarding evolutionary relationships between each query and the similar sequences. This is because, to our knowledge, there are no universally applicable similarity thresholds that are sufficient to distinguish between homologues and non-homologues (for further discussion, see [42]). Moreover, distinguishing among types of homologues, such as orthologues and paralogues, is also often essential for research projects. This is best done using phylogenetic analysis, and has been discussed in detail elsewhere [18, 45–47].

2 Code and data availability

All code needed for reproducing the analysis presented herein, as well as additional documentation, is available as version 3.0 of the AMOEBAE code repository on GitHub (https://github.com/laelbarlow/amoebae). This code repository is also permanently archived on

Zenodo (DOI: 10.5281/zenodo.5825385) [51]. All sequence data analyzed herein are automatically downloaded when the described code is run.

3 Method

3.1 Overall approach

Like many other workflows, AMOEBAE involves retrieval of similar sequences and comparative information useful for inferring some basic information about the phylogeny of the retrieved sequences from similarity search results (Figure 1). In addition to applying a variety of similarity search algorithms to identify potential homologues, AMOEBAE applies a reciprocal-best-hit (also known as best bi-directional hit) search strategy (Figure 1). This is a simple approach which has been common since the advent of the genomic age (for example, see [48]).
Reciprocal-best-hits are often orthologous among species in a given taxonomic group, and AMOEBAE allows application of E-value thresholds which make such designations more reliable (see below). Reciprocal-best-hit searching involves two main steps: 1) Searching a subject genome (database) with a query sequence from a reference genome. This is usually referred to as a *forward search*. 2) Searching back in the reference genome (the source of the original query) using forward search hit sequences as queries. These are usually referred to as *reverse searches*. If a forward search hit retrieves the original query as the top hit in a reverse search, then this is an indication that it has a close relationship with the original query and may be orthologous (Figure 1).

Three features distinguish AMOEBAE from other workflows. Firstly, the organization of tBLASTn searches (searches with peptide sequence queries against nucleotide sequence databases) in parallel with searches of amino acid sequences. This contrasts with workflows such as OrthoFinder [33] that do not (currently) perform tBLASTn searches. This may be important for some projects, because homologous sequences are often absent from predicted protein sequences while present in nucleotide sequences of genome assemblies [49]. However, extraction of meaningful amino acid sequence predictions and comparison of sequences to those retrieved from predicted proteins can be very difficult to manage without automation. Secondly, AMOEBAE provides ease of running searches with custom sequence profile queries (in parallel with other search methods). This allows more sensitive taxon-specific sequence alignments to be used, and allows for custom trimming of alignments to focus on domains of interest. This is useful for identifying distant homologues. Third and finally, AMOEBAE provides a means to implement custom criteria for filtering out redundant sequence copies to identify the number of paralogues of interest genomes contain. This is particularly important in cases where high-quality amino acid sequence predictions are not available and genome assemblies are prone to artefacts due to confusion of alleles with paralogues [50].

3.2 Installation

3.2.1 Computational resources

The protocol described herein is for installing and running AMOEBAE on a computer with the Ubuntu Linux operating system (https://ubuntu.com). The same procedures should work on MacOS operating systems and Linux distributions run in virtual machines or subsystems on Windows machines. This protocol does not include instructions for running AMOEBAE on Linux HPCs, as this requires system-specific configuration. However, guidelines for this are provided in the full documentation for AMOEBAE which is available *via* GitHub and archived on Zenodo [51]. Most applications of AMOEBAE will benefit from use of an HPC cluster due to requirements of storage (~30GB or more) and computation time. Analysis of the example input files described below will take between 30 and 60 minutes, depending on resource availability on your system.

3.2.2 Dependency installation

This workflow has minimal essential dependencies for installation, which are all widely used in many areas of computational biology. These are all free software and can be installed as described below. Initial setup is done *via* the Conda package and environment manager (https://docs.conda.io/projects/conda/en/latest/user-guide/index.html). AMOEBAE is executed *via* the Snakemake workflow management system, which will install further packages in virtual environments for each analysis step as needed [52].

Follow these steps to install initial dependencies on your computer:

1. If you do not have the Conda package manager installed, install it *via* the latest version of the Miniconda3 installer appropriate to your system (https://docs.conda.io/en/latest/miniconda.html).
2. Use Conda to install the Mamba (https://github.com/mamba-org/mamba) software package for increasing installation speed, using the following command in your command-line terminal:

   ```
   conda install -c conda-forge mamba
   ```

3. Use Mamba to install additional dependencies (snakemake, graphviz, and git) in a new conda environment named “snakemake”:

   ```
   mamba create -n snakemake -c anaconda -c conda-forge -c bioconda \
   ```

```
snakemake \
graphviz \
git
```

4. Activate the newly created conda environment as follows (commands described in subsequent steps of the workflow must always be run with this environment activated):

```
conda activate snakemake
```

3.2.3 Acquiring a copy of the AMOEBAE code

Navigate to an appropriate directory *via* your command-line terminal, and clone the AMOEBAE code repository using Git (https://git-scm.com/):

```
git clone https://github.com/laelbarlow/amoebae.git
```

3.3 Workflow configuration

3.3.1 Genome sequence files

Input FASTA files for searching may be predicted peptide FASTA files (.faa) and/or nucleotide FASTA files (.fna). For any genomic nucleotide FASTA files used, you may also include an associated General Feature Format Version 3 (GFF3) (https://github.com/The-Sequence-Ontology/Specifications/blob/master/gff3.md) annotation file (.gff3), which defines where genes are located in the genomic nucleotide sequences and is usually provided with genomic data. These data files can either be provided locally or automatically downloaded from databases such as NCBI as a part of the workflow.

The sequence files for our example analysis are listed in the config/example_genomes.csv file. These are all available for download from the NCBI genome database (https://www.ncbi.nlm.nih.gov/genome), and this Comma-Separated values (CSV) file contains all the information AMOEBAE needs to download them automatically. To use these example genomes, navigate to the cloned amoebae directory in your command-line terminal, and make a copy of this example file:

```
cd amoebae

cp config/example_genomes.csv config/genomes.csv
```

In addition, one of the peptide sequence files must be specified as the reference genome. This will be interrogated in the reverse searches. For our example analysis, we will use the peptide sequence FASTA file for *Rhizophagus irregularis* proteins, because this species possesses a complete set of AP complex subunits [37]. To define this as the reference genome, copy the example file as follows:

```
cp config/example_reference_db_list.txt config/reference_db_list.txt
```

3.3.2 Query sequence files

Input query files must contain peptide (amino acid) sequences, and may be in either single-FASTA (for BLAST searches) or aligned multi-FASTA format (for profile searches with HMMER).

The query sequence files for our example analysis are listed in the config/example_queries.csv file. These are all available for download from the NCBI protein database (https://www.ncbi.nlm.nih.gov/protein), and this CSV file contains all the information AMOEBAE needs to download them automatically. To use these example queries, navigate to the cloned amoebae directory in your command-line terminal, and make a copy of this example file:

```
cp config/example_queries.csv config/queries.csv
```

AMOEBAE allows searching for the same set of genes with multiple queries. This works by assigning a title to each query, with some titles assigned to multiple queries (*e.g.*, known orthologues of interest). These query titles are specified in the filenames of query sequence files as the text before the first underscore character. For example, in the example queries.csv file, the query title in both the filenames “AP4beta_Rirregularis_XP_025167196.1_query.faa” and “AP4beta_hmm.faa” is “AP4beta”. So, any positive hits retrieved with these queries (*via*

BLASTp and HMMER, respectively) will be reported as positive hits for AP4beta in the output Coulson plot.

3.3.3 Sequence similarity search parameters

AMOEBAE allows numerous search parameters to be configured. To customize parameters, it is necessary to modify the workflow/Snakefile file (Snakemake workflow definition file) in the cloned amoebae directory. For a full description of parameters, refer to the AMOEBAE command line interface documentation on GitHub. For information on Snakemake workflow definition files, refer to the Snakemake documentation (https://snakemake.readthedocs.io/en/stable/). Key sequence similarity search parameters and their default values are as follows:

1. <u>Forward search E-value threshold</u>. This is the maximum E-value for including forward search hits for downstream analysis. Forward search hits that do not meet this maximum threshold will not be used as reverse search queries. The default value is 0.0005.

2. <u>Reverse search E-value threshold</u>. This is the maximum E-value for including reverse search hits in the analysis. Reverse search hits that do not meet this maximum threshold will not be counted. The default value is 0.05.

3. <u>Order of magnitude E-value difference threshold</u>. This is the difference in E-value (as an order of magnitude) between the top reverse search hit and subsequent hits. So, this is a measure of how much more similar a forward search hit is to the original

query than to other sequences in a reference genome. The purpose is to identify cases where reverse search hits have approximately equivalent similarity to a reverse search query (see Figure 1C for examples). The default value is 5.

3.3.4 Output plot organization

To summarize results of multiple similarity searches, AMOEBAE outputs heatmap-style presence/absence plots, as well as Coulson plots [53]. In both cases, genome sequence files or species names are listed on the left side of the plot. The order in which these appear in the plot may be specified for easier interpretation (in accordance with taxonomic relationships). Also, Coulson plots contain subplots with results for different queries grouped together. These groupings can also be specified, and for our example analysis these correspond to the various subunits of different AP complexes. To use the example plot configuration, copy the example database (genome) list file and Coulson plot organization file as follows:

```
cp config/example_output_plot_row_order.txt \
config/output_plot_row_order.txt
cp config/example_coulson_plot_organization.csv \
config/coulson_plot_organization.csv
```

3.4 Running the workflow

After configuration, the workflow is run *via* the Snakemake command-line interface (https://snakemake.readthedocs.io/en/stable/). The workflow is composed of numerous steps, proceeding from sequence download through sequence similarity searching to plotting results (Figure 2), and each of these is defined in the Snakemake workflow definition file (Snakefile). Results of sequential analysis steps are progressively appended to summary files in CSV format, and these are summarized in the plots. An essential break point in the workflow is selection of reference sequences from the reference genome for the purpose of interpreting reverse search results (see below).

3.4.1 Selecting reference sequences

Before running the full workflow, it is necessary to select reference sequences for interpreting reverse searches. This is done by first running the initial steps in the workflow, and stopping after searching the reference genome (peptide sequences) using all the queries. Then an intermediate result file (saved as "config/Ref_seqs_1_manual_selections.csv") is modified to identify all sequences in the reference genome which are expected to be retrieved as top hits by sequences of interest from other genomes. Each row in this file corresponds to a sequence in the reference genome retrieved with one of the queries in a BLASTp sequence similarity search. Each reference sequence is identified as either accepted or unaccepted as a top hit in reverse searches.

To generate this intermediate file, run the following command in your command-line terminal (after navigating again to the amoebae directory and activating the appropriate conda environment):

```
snakemake get_ref_seqs -j 100 --use-conda
```

This will generate a file named “results/Ref_seqs_1_auto_predictions.csv”. For our example analysis, we will use a previously configured version of this file, which you can copy as well:

```
cp config/example_Ref_seqs_1_manual_selections.csv \
config/Ref_seqs_1_manual_selections.csv
```

Otherwise, copy the newly generated automatic predictions file to the config directory:

```
cp results/Ref_seqs_1_auto_predictions.csv \
config/Ref_seqs_1_manual_selections.csv
```

Then modify values in the 5th column of the “config/Ref_seqs_1_manual_selections.csv” file. In this column, “+” indicates inclusion of a sequence as a representative of the query used, and “-” indicates that the sequence is too distantly related to the query to be relevant.

3.4.2 Executing searches in all the genomes

Execute the remainder of the workflow to perform all searches in your genomes of interest:

```
snakemake -j 100 --use-conda
```

This may take several minutes or hours to run, depending on the number of queries and genomes, and will generate several output files and directories within the amoebae/results directory. The most important output files are as follows:

1. The heatmap (results/plot.pdf; See Figure 3) and the Coulson plot (results/plot_coulson_both.pdf; see Figure 4) containing results of searches in both amino acid sequences and nucleotide sequences.
2. The final result summary spreadsheet in CSV format (Supplementary file 1): results/fwd_srchs_1_rev_srch_1_interp_with_ali_col_nonredun.csv
3. The directory containing alignments of identified homologous sequences: results/fwd_srchs_1_rev_srch_1_interp_with_ali_col_nonredun_fasta_ali_files

3.5 Interpreting results

Results require careful interpretation, and in most cases re-analysis with modified parameters will be necessary as well as follow up with additional methods such as phylogenetic analysis.

In the case of our example analysis, searches of protein sequences (BLASTp, HMMER) yielded similar results to searches in nucleotide sequences (tBLASTn) (Supplementary file 1, Figure 3).

In our case *C. neoformans* sequences were retrieved *via* all three methods, and tBLASTn did not retrieve any additional homologues. AP4 epsilon subunits appear to be retained in many basidiomycetes including mushroom-forming fungi such as *Amanita muscaria*, not just *C. neoformans*. However, none of the basidiomycete genomes examined appear to retain any other AP4 subunits. Overall, these results support the hypothesis that the lone AP4 epsilon subunit plays an important role in basidiomycete cell biology, despite the absence of other AP4 subunits.

4 Limitations, pitfalls, and mitigation strategies

Output from any workflow must be carefully considered by the biologists performing the analysis and for any AMOEBAE results, it is important to consider how they might be misleading. Very often comparative genomics data benefits from phylogenetic analysis to identify the timing of important and evolutionarily informative duplications of cellular machinery, and to simply distinguish orthologues from paralogues. Molecular cell biological investigation of newly identified proteins is often enlightening and exciting. Even when stopping at the comparative genomics stage, it is worth considering possible sources of error. False-negative results may occur due to insufficient sensitivity of BLAST, and even HMMER. More sensitive methods are available such as HHblits [54], and may be important for some projects. With any sequence similarity search method, even if sensitivity is not an issue, there are no universally applicable thresholds for measures of sequence similarity such as E-values. As a consequence, stringent thresholds will yield more false-negative results, and inclusive thresholds will yield more false-positive results. AMOEBAE currently does not provide the option to specify per-query or per-genome search criteria. Different thresholds may be appropriate for different analyses. It may be useful to apply inclusive thresholds initially and analyze results to identify false-positives, and adjust criteria accordingly. The detailed results summary tables output by AMOEBAE make this process relatively easy.

In the preparation of this manuscript, in addition to running the example dataset provided along with the AMOEBAE package on Github, each of the co-authors also used the package to reproduce analyses on their own unpublished data. These data had been analyzed by different algorithms or by using those integrated into AMOEBAE, but run outside the workflow. This analysis of a variety of datasets, allowed us to identify dataset types that warrant extra attention and caution. These stem from important limitations which are inherent to any workflow that relies on reciprocal-best-hit similarity searching to predict which homologues are orthologues and are not issues with sensitivity, but with specificity. Chief among these is the case where a set of protein machinery within a given genome or group of organisms has undergone paralogous expansion followed by differential loss of paralogues among lineages. This can easily create situations in which reciprocal-best-hit searches yield false-positive results. In some cases, it may be possible to circumvent this by selecting reference genomes that have a relatively complete set of paralogues of genes of interest. Alternative methods such as OrthoMCL [31] which cluster sequences based on all-against-all pairwise alignments may be less susceptible to this source of error in principle. As well, variation in sequence divergence rates among paralogues can hinder any method based on pairwise sequence comparisons. This will be most noticeable when investigating sequence data from organisms that are distantly related to the other comparison points, i.e. newly discovered organisms, or in rapidly evolving organisms, exemplified by some parasitic lineages. The obvious way to address these issues is with phylogenetic analysis methods that implement appropriate models of amino acid sequence substitution. The filtered sets of homologues output by AMOEBAE may be particularly useful as input to multiple sequence alignments for phylogenetic analysis.

5 Final conclusions

Regardless of whether AMOEBAE gains further popularity, we foresee a continued need for reproducible bioinformatics workflows with the functionality offered by AMOEBAE to address questions in the field of (evolutionary) cell biology as discussed above. Thus, we hope that AMOEBAE will eventually serve as a benchmark for inevitable development of further bioinformatics workflows in this area of research.

6 Author contributions

LDB conceived of and wrote the AMOEBAE workflow. JBD conceived of the manuscript and directed the collaboration. LDB, WM, KM, KT, RV, and KZ tested the workflow on example datasets as well as other datasets (data not shown) and provided feedback for iterative improvement or identification of problematic dataset types. LDB and JBD drafted the manuscript, and all authors contributed to editing the manuscript.

7 Acknowledgments

The authors wish to thank Igor Sinelnikov, Information Services and Technology University of Alberta for his technical expertise in setting up and maintaining the computational cluster on which the Dacks lab AMOEBAE version is housed and on which the beta testing of this version was performed. His technical support over the years for the Dacks Lab computational resources has been invaluable.

AMOEBAE was initially developed at the Dacks Laboratory at the University of Alberta, and was supported by National Sciences and Engineering Council of Canada (NSERC) Discovery grants RES0021028, RES0043758, and RES0046091 awarded to Joel B. Dacks, as well as an NSERC Postgraduate Scholarship-Doctoral awarded to Lael D. Barlow. Kara Terry was supported by an Alberta Innovates Health Solutions Summer Research Studentship. Will Maciejowski was supported by an Office of the Provost and VP (Academic) Summer Studentship Award. Kiran More was supported by an NSERC Alexander Graham Bell Canada Graduate Scholarship - Master's and a Walter H. Johns Graduate Fellowship.

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG).

The authors gratefully acknowledge helpful discussion with many additional past members of the Dacks laboratory, as well as numerous colleagues around the globe particularly in the fields of protistology and evolutionary cell biology.

8 References

1. Lynch M, Field MC, Goodson HV, et al (2014) Evolutionary cell biology: Two origins, one objective. Proc Natl Acad Sci 111:16990–16994. https://doi.org/10.1073/pnas.1415861111
2. Horváthová L, Žárský V, Pánek T, et al (2021) Analysis of diverse eukaryotes suggests the existence of an ancestral mitochondrial apparatus derived from the bacterial type II secretion system. Nat Commun 12:2947. https://doi.org/10.1038/s41467-021-23046-7
3. Chan CJ, Le R, Burns K, et al (2019) BioID Performed on Golgi Enriched Fractions Identify C10orf76 as a GBF1 Binding Protein Essential for Golgi Maintenance and Secretion. Mol Cell Proteomics 18:2285–2297. https://doi.org/10.1074/mcp.RA119.001645
4. McNally KE, Faulkner R, Steinberg F, et al (2017) Retriever is a multiprotein complex for retromer-independent endosomal cargo recycling. Nat Cell Biol 19:1214–1225. https://doi.org/10.1038/ncb3610
5. Stairs CW, Dharamshi JE, Tamarit D, et al (2020) Chlamydial contribution to anaerobic metabolism during eukaryotic evolution. Sci Adv 6:eabb7258. https://doi.org/10.1126/sciadv.abb7258
6. Zaremba-Niedzwiedzka K, Caceres EF, Saw JH, et al (2017) Asgard archaea illuminate the origin of eukaryotic cellular complexity. Nature 541:353–358. https://doi.org/10.1038/nature21031
7. Archuleta TL, Frazier MN, Monken AE, et al (2017) Structure and evolution of ENTH and VHS/ENTH-like domains in tepsin. Traffic 18:590–603. https://doi.org/10.1111/tra.12499

8. Gershlick DC, Schindler C, Chen Y, Bonifacino JS (2016) TSSC1 is novel component of the endosomal retrieval machinery. Mol Biol Cell 27:2867–2878. https://doi.org/10.1091/mbc.e16-04-0209

9. Kirkham M, Nixon SJ, Howes MT, et al (2008) Evolutionary analysis and molecular dissection of caveola biogenesis. J Cell Sci 121:2075–2086. https://doi.org/10.1242/jcs.024588

10. Leung KF, Dacks JB, Field MC (2008) Evolution of the multivesicular body ESCRT machinery; retention across the eukaryotic lineage. Traffic 9:1698–716. https://doi.org/10.1111/j.1600-0854.2008.00797.x

11. Hirst J, Schlacht A, Norcott JP, et al (2014) Characterization of TSET, an ancient and widespread membrane trafficking complex. eLife 3:e02866. https://doi.org/10.7554/eLife.02866

12. Hirst J, D. Barlow L, Francisco GC, et al (2011) The Fifth Adaptor Protein Complex. PLoS Biol 9:e1001170. https://doi.org/10.1371/journal.pbio.1001170

13. Hirst J, Edgar JR, Esteves T, et al (2015) Loss of AP-5 results in accumulation of aberrant endolysosomes: Defining a new type of lysosomal storage disease. Hum Mol Genet 24:4984–4996. https://doi.org/10.1093/hmg/ddv220

14. Dacks JB, Field MC (2018) Evolutionary origins and specialisation of membrane transport. Curr Opin Cell Biol 53:70–76. https://doi.org/10.1016/j.ceb.2018.06.001

15. More K, Klinger CM, Barlow LD, Dacks JB (2020) Evolution and Natural History of Membrane Trafficking in Eukaryotes. Curr Biol 30:R553–R564. https://doi.org/10.1016/j.cub.2020.03.068

16. Archibald JM, Simpson AGB, Slamovits CH (2017) Handbook of the Protists, Second edition. Springer International Publishing

17. Mowbrey K, Dacks JB (2009) Evolution and diversity of the Golgi body. FEBS Lett 583:3738–3745. https://doi.org/10.1016/j.febslet.2009.10.025

18. Klute MJ, Melaçon P, Dacks JB (2011) Evolution and diversity of the Golgi. Cold Spring Harb Perspect Biol 3:1–17. https://doi.org/10.1101/cshperspect.a007849

19. Cavalier-Smith T (1987) The Origin of Eukaryote and Archaebacterial Cells. Ann N Y Acad Sci 503:17–54. https://doi.org/10.1111/j.1749-6632.1987.tb40596.x

20. Dacks JB, Davis LAM, Sjögren AM, et al (2003) Evidence for Golgi bodies in proposed "Golgi-lacking" lineages. Proc Biol Sci 270 Suppl:S168-71. https://doi.org/10.1098/rsbl.2003.0058

21. Marti M, Hehl AB (2003) Encystation-specific vesicles in *Giardia*: a primordial Golgi or just another secretory compartment? Trends Parasitol 19:440–446. https://doi.org/10.1016/S1471-4922(03)00201-0

22. Talamás‐Lara D, Acosta‐Virgen K, Chávez‐Munguía B, et al (2021) Golgi apparatus components in *Entamoeba histolytica* and *Entamoeba dispar* after monensin treatment. Microsc Res Tech 84:1887–1896. https://doi.org/10.1002/jemt.23745

23. Beznoussenko GV, Ragnini-Wilson A, Wilson C, Mironov AA (2016) Three-dimensional and immune electron microscopic analysis of the secretory pathway in *Saccharomyces cerevisiae*. Histochem Cell Biol 146:515–527. https://doi.org/10.1007/s00418-016-1483-y

24. Herman EK, Yiangou L, Cantoni DM, et al (2018) Identification and characterisation of the cryptic Golgi apparatus in *Naegleria gruberi*. J Cell Sci jcs.213306.

https://doi.org/10.1242/jcs.213306

25. Kurz S, Tiedtke A (1993) The Golgi Apparatus of *Tetrahymena Thermophila*. J Eukaryot Microbiol 40:10–13. https://doi.org/10.1111/j.1550-7408.1993.tb04874.x

26. Brugerolle G, Viscogliosi E (1994) Organization and composition of the striated roots supporting the Golgi apparatus, the so-called parabasal apparatus, in parabasalid flagellates. Biol Cell 81:277–285. https://doi.org/10.1016/0248-4900(94)90010-8

27. Barlow LD, Nývltová E, Aguilar M, et al (2018) A sophisticated, differentiated Golgi in the ancestor of eukaryotes. BMC Biol 16:. https://doi.org/10.1186/s12915-018-0492-9

28. Cantalapiedra CP, Hernández-Plaza A, Letunic I, et al (2021) eggNOG-mapper v2: Functional Annotation, Orthology Assignments, and Domain Prediction at the Metagenomic Scale. Bioinformatics

29. 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

30. Blum M, Chang H-Y, Chuguransky S, et al (2021) The InterPro protein families and domains database: 20 years on. Nucleic Acids Res 49:D344–D354. https://doi.org/10.1093/nar/gkaa977

31. Li L (2003) OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes. Genome Res 13:2178–2189. https://doi.org/10.1101/gr.1224503

32. Cosentino S, Iwasaki W (2019) SonicParanoid: fast, accurate and easy orthology inference. Bioinformatics 35:149–151. https://doi.org/10.1093/bioinformatics/bty631

33. 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

34. Burgos PV, Mardones GA, Rojas AL, et al (2010) Sorting of the Alzheimer's disease amyloid precursor protein mediated by the AP-4 complex. Dev Cell 18:425–436. https://doi.org/10.1016/j.devcel.2010.01.015

35. Hirst J, Bright NA, Rous B, Robinson MS (1999) Characterization of a Fourth Adaptor-related Protein Complex. Mol Biol Cell 10:2787–2802. https://doi.org/10.1091/mbc.10.8.2787

36. Davies AK, Itzhak DN, Edgar JR, et al (2018) AP-4 vesicles contribute to spatial control of autophagy via RUSC-dependent peripheral delivery of ATG9A. Nat Commun 9:3958. https://doi.org/10.1038/s41467-018-06172-7

37. Barlow LD, Dacks JB, Wideman JG (2014) From all to (nearly) none: Tracing adaptin evolution in Fungi. Cell Logist 4:e28114. https://doi.org/10.4161/cl.28114

38. Field MC, Gabernet-Castello C, Dacks JB (2007) Reconstructing the evolution of the endocytic system: insights from genomics and molecular cell biology. Adv Exp Med Biol 607:84–96. https://doi.org/10.1007/978-0-387-74021-8_7

39. Wilson G (2016) Software Carpentry: lessons learned. F1000Research 3:62. https://doi.org/10.12688/f1000research.3-62.v2

40. Ekblom R, Wolf JBW (2014) A field guide to whole-genome sequencing, assembly and annotation. Evol Appl 7:1026–1042. https://doi.org/10.1111/eva.12178

41. Yandell M, Ence D (2012) A beginner's guide to eukaryotic genome annotation. Nat Rev Genet 13:329–342. https://doi.org/10.1038/nrg3174

42. Pearson WR (2013) An Introduction to Sequence Similarity ("Homology") Searching. Curr

Protoc Bioinforma 1:1286–1292. https://doi.org/10.1002/0471250953.bi0301s42.An

43. Camacho C, Coulouris G, Avagyan V, et al (2009) BLAST+: architecture and applications. BMC Bioinformatics 10:421. https://doi.org/10.1186/1471-2105-10-421

44. Eddy SR (1998) Profile hidden Markov models. Bioinformatics 14:755–763. https://doi.org/10.1093/bioinformatics/14.9.755

45. Altenhoff AM, Dessimoz C (2012) Inferring Orthology and Paralogy. In: Anisimova M (ed) Evolutionary Genomics. Humana Press, Totowa, NJ, pp 259–279

46. Gabaldón T (2008) Large-scale assignment of orthology: back to phylogenetics? Genome Biol 9:235. https://doi.org/10.1186/gb-2008-9-10-235

47. Hooff JJE, Tromer E, Dam TJP, et al (2019) Inferring the Evolutionary History of Your Favorite Protein: A Guide for Molecular Biologists. BioEssays 41:1900006. https://doi.org/10.1002/bies.201900006

48. Rivera MC, Jain R, Moore JE, Lake JA (1998) Genomic evidence for two functionally distinct gene classes. Proc Natl Acad Sci 95:6239–6244. https://doi.org/10.1073/pnas.95.11.6239

49. Deutekom ES, Vosseberg J, van Dam TJP, Snel B (2019) Measuring the impact of gene prediction on gene loss estimates in Eukaryotes by quantifying falsely inferred absences. PLOS Comput Biol 15:e1007301. https://doi.org/10.1371/journal.pcbi.1007301

50. Larson RT, Dacks JB, Barlow LD (2019) Recent gene duplications dominate evolutionary dynamics of adaptor protein complex subunits in embryophytes. Traffic 20:961–973. https://doi.org/10.1111/tra.12698

51. Barlow LD (2022) AMOEBAE v3.0. Zenodo. https://doi.org/10.5281/zenodo.5825385

52. Molder F, Jablonski KP, Letcher B, et al (2020) Sustainable data analysis with Snakemake. F1000Research 10:33. https://doi.org/10.12688/f1000research.29032.2

53. Field HI, Coulson RM, Field MC (2013) An automated graphics tool for comparative genomics: the Coulson plot generator. BMC Bioinformatics 14:141. https://doi.org/10.1186/1471-2105-14-141

54. Steinegger M, Meier M, Mirdita M, et al (2019) HH-suite3 for fast remote homology detection and deep protein annotation. BMC Bioinformatics 20:473. https://doi.org/10.1186/s12859-019-3019-7

9 Figures

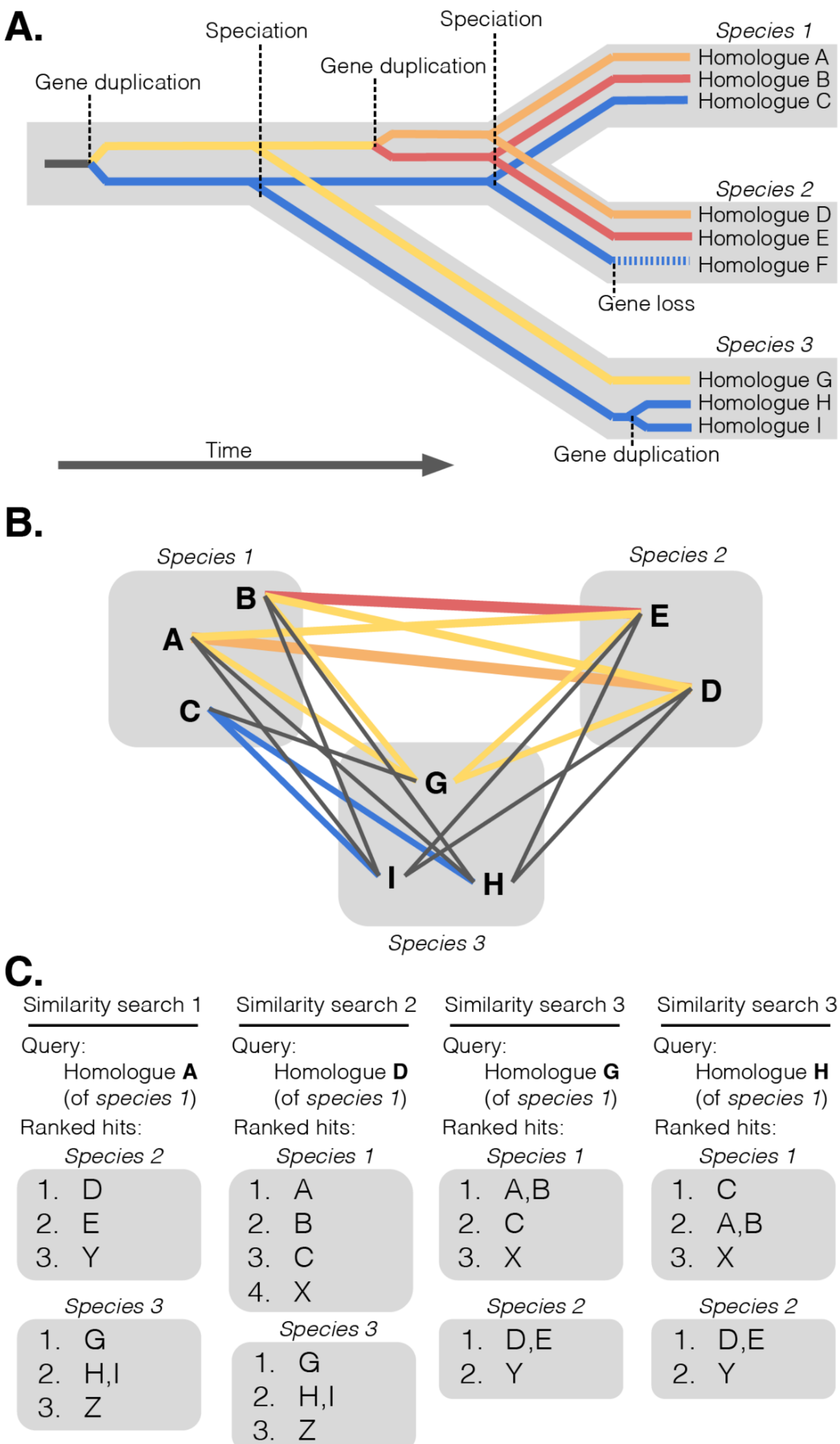
A.
Speciation
Speciation
Gene duplication
Gene duplication
Species 1
Homologue A
Homologue B
Homologue C
Species 2
Homologue D
Homologue E
Homologue F
Gene loss
Species 3
Homologue G
Homologue H
Homologue I
Gene duplication
Time
B.
Species 1
Species 2
Species 3
A
B
C
D
E
G
H
I
C.
Similarity search 1
Query: Homologue A (of species 1)
Ranked hits:
Species 2
1. D
2. E
3. Y
Species 3
1. G
2. H,I
3. Z
Similarity search 2
Query: Homologue D (of species 1)
Ranked hits:
Species 1
1. A
2. B
3. C
4. X
Species 3
1. G
2. H,I
3. Z
Similarity search 3
Query: Homologue G (of species 1)
Ranked hits:
Species 1
1. A,B
2. C
3. X
Species 2
1. D,E
2. Y
Similarity search 3
Query: Homologue H (of species 1)
Ranked hits:
Species 1
1. C
2. A,B
3. X
Species 2
1. D,E
2. Y

Figure 1: A simple example of the connection between gene phylogeny and similarity search results. A) The phylogeny of a set of homologous genes (which is unknown in the context of real analyses). The evolution of genes is punctuated by speciation and gene duplication events, as well as gene loss. B) Assuming constant, moderate rates of stochastic basepair substitution over time, homologous genes produced by more recent speciation or gene duplication events will retain higher degrees of sequence similarity (line thickness is proportional to sequence similarity between pairs of homologous genes). C) Similarity search results are rankings of sequences by similarity to a query sequence, as indicated by E-values or other measures. Here, hits with equivalent similarity to the query are assigned the same rank, and X, Y, and Z are not homologous to the other genes (yet have some similarity). Such results are a consequence of the evolutionary events in (A) and the differing degrees of pairwise sequence similarity in (B), and allow the phylogenetic relationships between homologues to be predicted to some extent. In this case, it is clear from (C) that homologues A and D are more closely related than either of these is to homologues G or H, as homologues A and D are reciprocal-best-hits (a.k.a. best bi-directional hits).

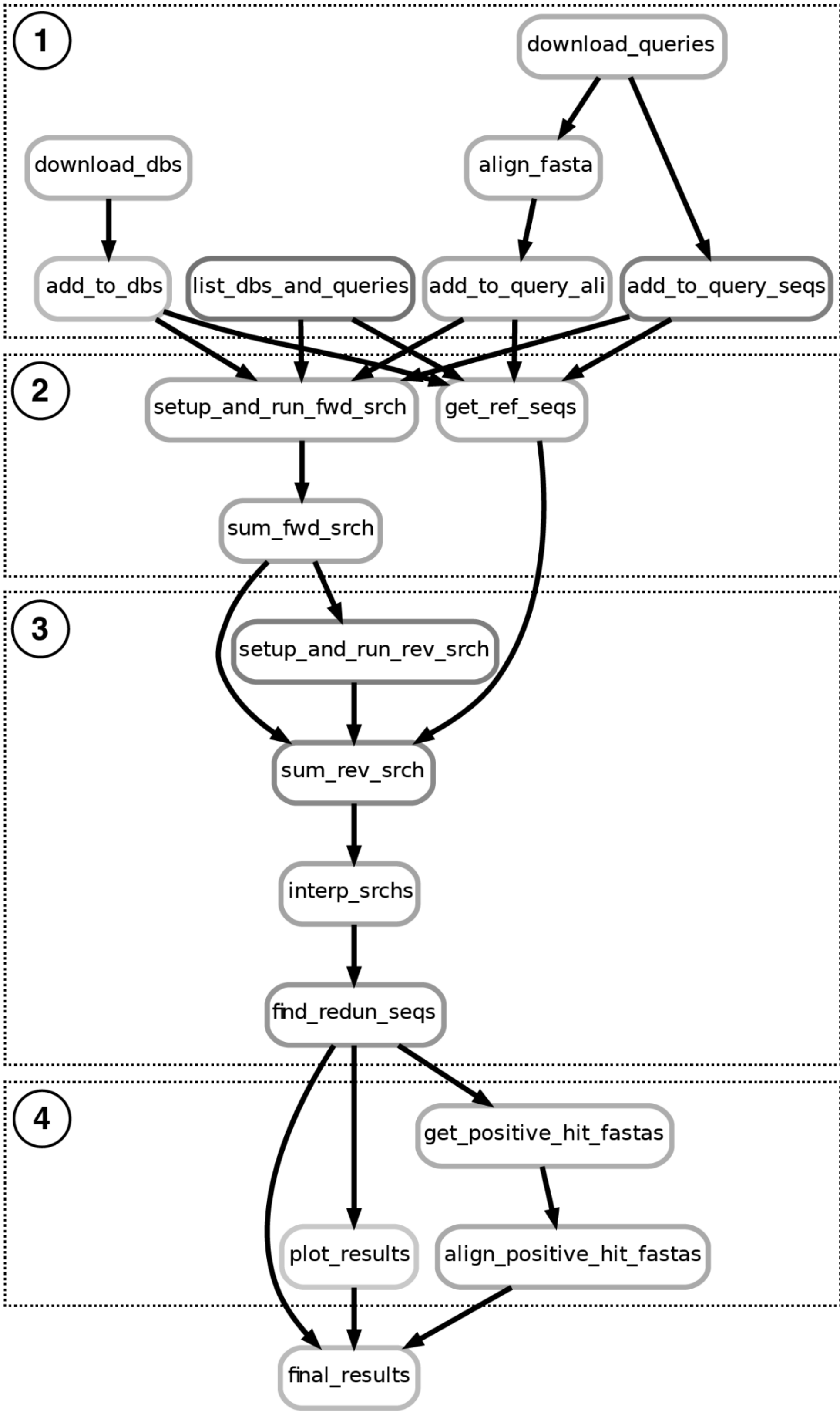
1
download_queries
download_dbs
align_fasta
add_to_dbs
list_dbs_and_queries
add_to_query_ali
add_to_query_seqs
2
setup_and_run_fwd_srch
get_ref_seqs
sum_fwd_srch
3
setup_and_run_rev_srch
sum_rev_srch
interp_srchs
find_redun_seqs
4
get_positive_hit_fastas
plot_results
align_positive_hit_fastas
final_results

Figure 2: Diagram of connections between steps in the AMOEBAE Snakemake workflow. Arrows indicate where output files from an analysis step are used as input for subsequent steps. Conceptually the workflow is divided into four stages (shown as numbered boxes): 1) Analysis set-up; 2) Forward searches; 3) Reverse searches and filtering; 4) Data visualization. Notably, AMOEBAE provides the intermediary results files and summaries for all steps and so users can examine the output at any stage, particularly to assess potential false positives or negatives.

	('Rhizophagus irregularis', 'Rhizophagus_irregularis.faa')	('Saccharomyces cerevisiae', 'Saccharomyces_cerevisiae.faa')	('Rhodotorula graminis', 'Rhodotorula_graminis.faa')	('Wallemia ichthyophaga', 'Wallemia_ichthyophaga.faa')	('Tremella mesenterica', 'Tremella_mesenterica.faa')	('Cryptococcus neoformans', 'Cryptococcus_neoformans.fna')	('Cryptococcus neoformans', 'Cryptococcus_neoformans.faa')	('Botryobasidium botryosum', 'Botryobasidium_botryosum.faa')	('Postia placenta', 'Postia_placenta.faa')	('Pleurotus ostreatus', 'Pleurotus_ostreatus.faa')	('Amanita muscaria', 'Amanita_muscaria.faa')
('AP1beta', 'AP1beta_Rirregularis_XP_025178476.1_query.faa')	2	2	2	2	2	0	2	2	2	2	2
('AP1gamma', 'AP1gamma_Rirregularis_XP_025180406.1_query.faa')	1	1	1	1	1	0	1	1	1	1	1
('AP1mu', 'AP1mu_Rirregularis_XP_025177352.1_query.faa')	1	2	1	1	1	0	1	1	1	1	1
('AP1sigma', 'AP1sigma_Rirregularis_XP_025175476.1_query.faa')	1	1	1	1	1	0	1	1	1	2	2
('AP2alpha', 'AP2alpha_Rirregularis_XP_025185160.1_query.faa')	1	1	1	1	1	0	1	1	1	1	1
('AP2beta', 'AP2beta_Rirregularis_XP_025182356.1_query.faa')	2	2	2	2	2	0	2	2	2	2	2
('AP2mu', 'AP2mu_Rirregularis_XP_025168352.1_query.faa')	1	1	1	1	1	0	1	1	1	1	1
('AP2sigma', 'AP2sigma_Rirregularis_XP_025170073.1_query.faa')	1	1	1	1	1	0	1	1	1	1	1
('AP3beta', 'AP3beta_Rirregularis_XP_025173006.1_query.faa')	1	1	1	1	1	0	1	1	1	1	1
('AP3delta', 'AP3delta_Rirregularis_XP_025171449.1_query.faa')	1	1	1	1	1	0	1	2	1	1	1
('AP3mu', 'AP3mu_Rirregularis_XP_025177884.1_query.faa')	1	1	1	1	1	0	1	1	1	1	1
('AP3sigma', 'AP3sigma_Rirregularis_XP_025179253.1_query.faa')	1	1	1	1	1	0	1	1	1	1	1
('AP4beta', 'AP4beta_Rirregularis_XP_025167196.1_query.faa')	1	0	0	0	0	0	0	0	0	0	0
('AP4beta', 'AP4beta_hmm.afaa')	0	0	0	0	0	-	0	0	0	0	0
('AP4epsilon', 'AP4epsilon_Rirregularis_XP_025176105.1_query.faa')	1	0	0	0	0	0	0	1	0	0	0
('AP4epsilon', 'AP4epsilon_hmm.afaa')	0	0	0	0	1	-	1	0	1	1	1
('AP4mu', 'AP4mu_Rirregularis_XP_025175475.1_query.faa')	1	0	0	0	0	0	0	0	0	0	0
('AP4mu', 'AP4mu_hmm.afaa')	0	0	0	0	0	-	0	0	0	0	0
('AP4sigma', 'AP4sigma_Rirregularis_XP_025183263.1_query.faa')	1	0	0	0	0	0	0	0	0	0	0
('AP4sigma', 'AP4sigma_hmm.afaa')	0	0	0	0	0	-	0	0	0	0	0

Figure 3: Positive hit counts by genome file and query file. The column labels indicate the query title and query file used for a similarity search. The row labels indicate the species/genome name and FASTA file searched. Only unique hits are counted (i.e., positive hits retrieved using multiple methods are only counted once in the case with the strongest E-value).

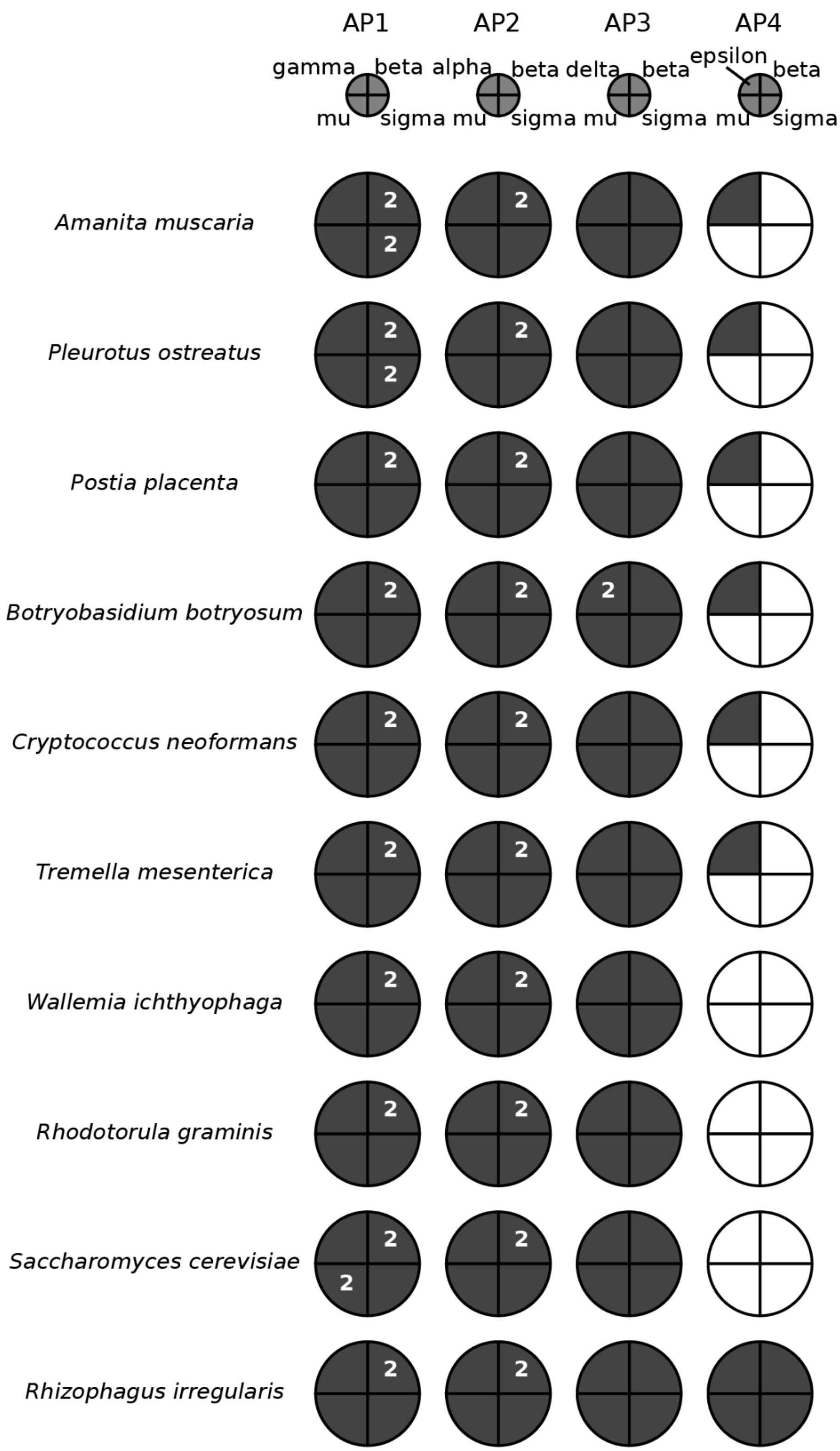
AP1
AP2
AP3
AP4
gamma
beta
alpha
beta
delta
beta
epsilon
beta
mu
sigma
mu
sigma
mu
sigma
mu
sigma
Amanita muscaria
Pleurotus ostreatus
Postia placenta
Botryobasidium botryosum
Cryptococcus neoformans
Tremella mesenterica
Wallemia ichthyophaga
Rhodotorula graminis
Saccharomyces cerevisiae
Rhizophagus irregularis
2

Figure 4: Coulson plot of search results, output by AMOEBAE. Each row contains results for searches in a different genome. Each column contains results of searches for subunits of a different Adaptor Protein (AP) complex (indicated in the legends above). In each subplot, white-filled sectors indicate absence of any orthologues of the relevant AP subunit, grey-filled sectors indicate presence of at least one orthologue, and numbers in white font indicate where more than one orthologue was identified.

10 Supplementary information

Supplementary file 1: Detailed summary of similarity search results output by AMOEBAE.