How to build phylogenetic species trees with OMA [version 1; peer review: 2 approved with reservations]

David Dylus\textsuperscript{1,3*}, Yannis Nevers\textsuperscript{1,3*}, Adrian M. Altenhoff\textsuperscript{1,4}, Antoine Gürtler\textsuperscript{2,3}, Christophe Dessimoz\textsuperscript{1,3,5,6}, Natasha M. Glover\textsuperscript{1,3}

\textsuperscript{1}Swiss Institute of Bioinformatics, Lausanne, 1015, Switzerland
\textsuperscript{2}Department of Computational Biology, University of Lausanne, Lausanne, 1015, Switzerland
\textsuperscript{3}Center for Integrative Genomics, University of Lausanne, Lausanne, 1015, Switzerland
\textsuperscript{4}Department of Computer Science, ETH Zurich, Zurich, 8092, Switzerland
\textsuperscript{5}Department of Genetics, Evolution and Environment, University College London, London, WC1E 6BT, UK
\textsuperscript{6}Department of Computer Science, University College London, London, WC1E 6BT, UK

* Equal contributors

Abstract
Knowledge of species phylogeny is critical to many fields of biology. In an era of genome data availability, the most common way to make a phylogenetic species tree is by using multiple protein-coding genes, conserved in multiple species. This methodology is composed of several steps: orthology inference, multiple sequence alignment and inference of the phylogeny with dedicated tools. This can be a difficult task, and orthology inference, in particular, is usually computationally intensive and error prone if done \textit{ad hoc}. This tutorial provides protocols to make use of OMA Orthologous Groups, a set of genes all orthologous to each other, to infer a phylogenetic species tree. It is designed to be user-friendly and computationally inexpensive, by providing two options: (1) Using only precomputed groups with species available on the OMA Browser, or (2) Computing orthologs using OMA Standalone for additional species, with the option of using precomputed orthology relations for those present in OMA. A protocol for downstream analyses is provided as well, including creating a supermatrix, tree inference, and visualization. All protocols use publicly available software, and we provide scripts and code snippets to facilitate data handling. The protocols are accompanied with practical examples.

Keywords
phylogenetics, phylogenomics, species tree, OMA, Orthologous Matrix
This article is included in the The OMA collection collection.

**Corresponding author:** Natasha M. Glover (natasha.glover@unil.ch)

**Author roles:**
- **Dylus D:** Conceptualization, Formal Analysis, Investigation, Methodology, Project Administration, Software, Visualization, Writing – Original Draft Preparation; **Nevers Y:** Data Curation, Formal Analysis, Investigation, Methodology, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Altenhoff AM:** Conceptualization, Data Curation, Methodology, Resources, Software, Writing – Review & Editing; **Gürtler A:** Formal Analysis, Investigation, Methodology; **Dessimoz C:** Funding Acquisition, Project Administration, Supervision, Writing – Review & Editing; **Glover NM:** Conceptualization, Project Administration, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing

**Competing interests:** No competing interests were disclosed.

**Grant information:** This work was funded by a Service and Infrastructure grant from the Swiss Institute of Bioinformatics and a Swiss National Science Foundation Professorship grant (Grant 183723). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Copyright:** © 2020 Dylus D et al. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**How to cite this article:** Dylus D, Nevers Y, Altenhoff AM et al. How to build phylogenetic species trees with OMA [version 1; peer review: 2 approved with reservations] F1000Research 2020, 9:511 https://doi.org/10.12688/f1000research.23790.1

**First published:** 04 Jun 2020, 9:511 https://doi.org/10.12688/f1000research.23790.1
Introduction

Inferring accurate and complete species phylogenies is a fundamental problem in biology\(^1\). Traditionally, species trees have been inferred using ubiquitous marker genes such as 16S rRNA ribosomal genes\(^2\). However, as there are fewer sites to sample, using only one gene per species limits the resolution of the inference and the phylogeny of the gene does not necessarily reflect the evolutionary history of the entire species. For this reason, species phylogenies are now overwhelmingly inferred from multiple genes\(^3\). As long as one takes the necessary precautions, notably selecting true orthologs for their comparisons, (see \(\text{3}\) for common pitfalls in phylogenomics), multilocus phylogenies are better resolved and more robust\(^4\). Recently, multiple protein-coding genes were used to infer a tree comprising ~3000 species, but this was still limited to a small number of concatenated ribosomal genes\(^5\). With the rise of next-generation sequencing, many hundreds of genes can now be considered when building species trees, which tremendously increases the available information for the inference.

To make use of all available genes, one needs to identify groups of genes that emerged from a common ancestral gene solely through speciation. These sets of genes, in which all pairs of genes are orthologs\(^6\), are commonly referred to as Orthologous Groups (OGs). The inference of OGs is non-trivial due to additional evolutionary events such as gene duplications, gene losses or horizontal gene transfers\(^7\). Furthermore, there are numerous algorithms for inferring orthology which can result in different OG composition, further complicating matters. In the Orthologous MATrix (OMA), OGs can have at most one representative gene per species. This type of OG is provided by a few orthology databases such as BUSCO\(^8\) and OMA\(^9,10\).

In this tutorial, we focus on OGs obtained from OMA. Alternatively called “OMA Groups,” they are specifically designed for species tree inference. OMA Groups are defined as gene families that contain genes which are all orthologous to each other, with a maximum of one gene per species\(^11\). These are stringently computed orthologs that make use of all the available species in OMA. Another term used for this type of groups are marker genes, or phylogenetic marker genes. Moreover, OMA Groups have repeatedly been shown to produce reliable trees\(^8\) and its underlying algorithm was shown to accurately infer OGs in a large-scale benchmark\(^12\).

The OMA algorithm is freely available as an open source software tool (OMA Standalone\(^13\)) and integrates well with the public OMA Browser (https://omabrowser.org), a database that provides orthology information among more than 2300 genomes across the tree of life. Here, we show how to leverage the publicly available orthology data to infer phylogenetic species trees.

First, we set up the prerequisites and explain how to search for pre-computed OGs for species of interest in the OMA database. Then, we show how to infer a species tree under two scenarios: (1) using only species that are present in the public OMA database, or (2) using species in OMA in addition to other genomes not available in the database, e.g. data obtained from sequencing the genome of a new species. Finally, we show how to do downstream processing and tree inference. Each of these steps is illustrated by an example on real data.

Materials

The tools needed for this tutorial can be found in Table 1. All commands can be run from the command line and/or with python scripts. We reference four tree inference software tools, but there are many other alternatives (see List of phylogenetics software).

Two examples will be used in this tutorial, one to illustrate Protocol 1, and another to illustrate Protocol 2. Both examples can be downloaded from Figshare\(^14\). Instructions on how to obtain the data from the OMA Browser for Protocol 1 is described in the next section, so it is not required to download to complete this tutorial. However, Protocol 2 is used to add external genomes. For this example we chose two genomes available in the OMA Browser, but we set them aside after downloading the data. We then re-add them as external data (FOMPI.fa and YEAST.fa). For reproducibility, these proteomes can be found in the data/AddedGenomes subdirectory of Protocol_2. The rest of the data included in the example tarball are OGs and alignment files needed to compute the trees, which are also included as results.

The tree computations on these data have been performed using both RaXML 8.2.12 and IQTree 1.7.beta17, as specified in the PDF accompanying the examples.

Protocols

Phylogenetic tree inference using OMA is done in three steps: getting OG data, aligning all sequences of every OG and combining them into a supermatrix, and finally, using tree inference tools on the supermatrix. Depending
Table 1. Computational tools needed for making a phylogenetic species tree using OMA. Note that four phylogenetic tree inference softwares are given, but only one (user preference) is needed to complete this tutorial.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Use case</th>
<th>How to get it</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command line</td>
<td>Mandatory to run the commands written in this tutorial</td>
<td>Installed by default on Unix and Mac</td>
</tr>
<tr>
<td>Python 3</td>
<td>Python language interpreter. Mandatory for running the scripts used in this paper</td>
<td><a href="https://www.python.org/downloads/">https://www.python.org/downloads/</a></td>
</tr>
<tr>
<td>OMA Browser</td>
<td>Needed to import orthology relations used for tree inference</td>
<td><a href="https://omabrowser.org">https://omabrowser.org</a></td>
</tr>
<tr>
<td>OMA standalone</td>
<td>Needed to infer orthology for data not available in the OMA Browser</td>
<td><a href="https://omabrowser.org/standalone/#downloads">https://omabrowser.org/standalone/#downloads</a></td>
</tr>
<tr>
<td>MAFFT</td>
<td>Multiple sequence alignment software</td>
<td><a href="https://mafft.cbrc.jp/alignment/software/">https://mafft.cbrc.jp/alignment/software/</a></td>
</tr>
<tr>
<td>High Performance Computing (HPC)</td>
<td>Needed if a high amount of computation is involved</td>
<td>Institutional infrastructure</td>
</tr>
<tr>
<td>IQTree</td>
<td>Phylogenetic tree software</td>
<td><a href="http://www.iqtree.org/#download">http://www.iqtree.org/#download</a></td>
</tr>
<tr>
<td>Phylobayes</td>
<td>Phylogenetic tree software</td>
<td><a href="http://megasun.bch.umontreal.ca/People/lartillot/www/download.html">http://megasun.bch.umontreal.ca/People/lartillot/www/download.html</a></td>
</tr>
<tr>
<td>PhyML</td>
<td>Phylogenetic tree software</td>
<td><a href="http://www.atgc-montpellier.fr/phyml/">http://www.atgc-montpellier.fr/phyml/</a></td>
</tr>
<tr>
<td>Phylo.io</td>
<td>Phylogenetic tree visualization website</td>
<td><a href="http://phylo.io">http://phylo.io</a></td>
</tr>
</tbody>
</table>

on the species requirement, two options are available to obtain OG data, they are detailed in subsection Protocol 1 and 2. Protocol 1 is the fastest and can be used if all species of interest are available on the OMA Browser. Alternatively, Protocol 2 is for the cases when new genomes must be added, or when solely using data computed by OMA Standalone. Later steps are the same for both cases and are addressed in Protocol 3.

Protocol 1: Export marker genes to make a phylogeny of species found in OMA
This method is the quickest way to obtain data to build a phylogenetic species tree, but is only useful if one is interested in making a tree from species already in the OMA database. To do this, the Export marker genes function in the browser takes advantage of the precomputed OMA Groups. As mentioned in the Introduction, OMA Groups are a specific type of OG which contain sets of genes that are all orthologous to one another. This implies that there is at most one gene from each species in a group.

Finding species of interest in OMA. The OMA public database and all related information are accessed through the OMA browser (https://omabrowser.org/). One can search for species of interest in the OMA database by browsing through the available data in OMA using the release info page (in the upper left of the home screen: Explore -> Release information). Two browsing options are available, the default one is through an interactive tree, with colours indicating domains of life: bacteria are blue, archaea are green and eukaryotes are red. The other option is a table viewer featuring a search bar and accessed through the View as Species List icon in the top right of the table.

Export the relevant data from OMA. The way to obtain OGs with only species present in the OMA database is by using the Compute -> Export marker genes option (Figure 1A). This will open a page which allows the
user to select species. Species can be searched by name or clade. A whole clade can be selected by clicking on
the node (select all species). A single species can be selected by clicking on the leaf (select species). All selected
species will be displayed in the right box with additional species information (release info, taxon id, etc.)
(Figure 1B).

Specifying the Minimum Species Coverage and Maximum Number of Markers parameters. After species
selection, exported OGs will depend on the minimum fraction of covered species and the maximum number of
markers parameters:

• Minimum species coverage: the lowest acceptable proportion of selected species that are present in
any given OG in order to be exported.

A more permissive (lower) minimum species coverage will result in a higher number of exported
groups. Choosing this parameter depends on the number of and how closely related are the selected
species. For instance, consider the Drosophila clade versus chordates clade (20 and 116 species in current
release, respectively). If one selects the 20 Drosophila genomes and sets the minimum species coverage
to 0.5, only OGs with at least 10 Drosophila species will be exported. In the current release, this
results in 11,855 OGs which meet this criteria. If using the same 0.5 minimum species coverage
for the chordates, it results in 14,357 OGs exported. On the other hand, for a 0.8 minimum species
coverage, 7,886 and 6,329 OGs are exported for Drosophila and chordates clades, respectively.

• Maximum number of markers: the maximum number of OGs/marker genes to return. To consider as
much information as possible in the tree inference, remove any limit by setting this parameter to -1, in
which case all OGs fulfilling the minimum species coverage parameter will be returned. To speed up
the tree inference, set this value to below 1000 genes.

After filling in the parameters and submitting the request, the browser will return a compressed archive
(“tarball”) that contains a fasta file with unaligned sequences for each OG. Depending on the size of the request,
it may take a few minutes for this operation to complete.

As an example for Protocol 1, we performed an analysis on 20 yeast species, using only OGs shared by
all species (Minimum species coverage: 1) and no limit to the number of OGs retrieved (Maximum number
of markers: -1). We obtained 169 OGs with this query. The corresponding data can be found on Figshare14, in the
Protocol_1 folder.
Upon exporting the marker genes, i.e. OGs, from OMA, the data can be used to make a phylogenetic species tree (skip to Protocol 3: Downstream processing and tree inference).

Protocol 2: Export precomputed OMA all-against-all data as a backbone to add your own genomes and use with OMA standalone

Orthology computation first starts with an all-against-all alignment phase—comparing all proteins in every species of interest to each other. If genomes to be included in the species phylogeny are not present in OMA (hereafter referred to as “added genomes”), it is necessary to first compute orthology predictions for the combined set of species (those in OMA plus the added genomes). This approach is computationally more expensive and requires that computations are performed on a local machine or high performance computing cluster (HPC). However, by using the OMA Browser’s Export -> All/All option, one can take advantage of the precomputed all-against-all data for those species in OMA, saving time. The following protocol describes how to make use of this data and run OMA Standalone, the software for running the OMA algorithm on added genomes. In the case where the user wants to only use genomes unavailable in OMA, skip to the “Running OMA Standalone” section.

**Export the all-against-all from OMA.** Choose species which you want to combine with your own genomes by choosing Compute -> Export All/All from the home page (Figure 1A). This will lead to an interactive species tree of all the species in OMA, for which you can choose your species of interest to export (Figure 1B).

After selecting species and clicking submit, the OMA Browser will export a tarball (described in Figure 2) which contains:

- The all-against-all alignments of the selected species, found in the folder “Cache.”
- All exported genomes, in the format of protein fasta files, found in the folder “DB.”
- The full OMA standalone software tool. No need to download it separately.

**Figure 2.** Tree organization of the tarball downloaded through the OMA Browser after exporting an all-against-all of selected species. The important files and folders are coloured. In green are the executable files mentioned in the course of the tutorial. In blue are the files and folder that will need to be modified. Other files and folders (in black) will not be used in the course of the tutorial. Files and folders not shown are represented by three dots.
Combining the added genomes with exported OMA data. Next, the added genomes data must be combined to the OMA data. For this procedure, the added genomes data must fulfil certain conditions:

- Each genome is in the form of a fasta file, containing protein sequences.
- The name of the fasta file is either the species name or the 5-letter UniProt identifier of the added species (e.g. ARATH for Arabidopsis thaliana).
- Each sequence in the fasta file has a clear and unique identifier. There must be no square brackets or other special characters.

If these conditions are fulfilled, these fasta files must be put into the DB folder with the other exported OMA genomes (Figure 2), where they will be considered as a unique dataset for the following steps.

Setting the parameters for OMA standalone. Before starting the computation, it is wise to adjust the parameters file, called “parameters.drw” (Figure 2), which can be edited with any text editor. If the goal is to only generate a dataset for species phylogeny inference (and not keeping other unrelated orthology inferences), one can avoid doing computations and generating output files that are not needed by the following:

- Uncomment (remove the # from) all the lines starting with WriteOutput EXCEPT #WriteOutput_OrthologousGroupsFasta := false. By keeping that one commented, OMA standalone will produce one fasta file for each inferred OG.
- Deactivate the Hierarchical Orthologous Group inference, which is not needed here, by setting DoHierarchicalGroups := false;
- Likewise, deactivate the gene function prediction by setting DoGroupFunctionPrediction := false;
- Tip: do not omit a semicolon at the end of each uncommented statement.

Running OMA standalone. To run OMA standalone, one needs to be aware that the OMA pipeline can be split into two parts: all-against-all homology inference and orthology calling. Because OMA can compute Smith-Waterman alignments in parallel for all species which were not exported from OMA (see Export the all-against-all from OMA), it is beneficial to perform the computations on a computer cluster. However, if the dataset is small (e.g. 2–3 additional genomes), the computations can be run locally on a standard computer.

To run OMA standalone on a small dataset locally:

1. Within the extracted tarball folder you can start the computation with the command line:

   ```
   $: bin/oma -n NR_PROCESSES
   ```

   NR_PROCESSES should not be higher than the number of CPUs you have available on your machine.

For a larger dataset, we recommend the use of an HPC cluster. We recommend breaking up the computations into two parts: first the all-against-all part, then the orthology inference part:

1. Create a submission script for your cluster. Examples of submission scripts are provided at [https://omabrowser.org/standalone/#schedulers](https://omabrowser.org/standalone/#schedulers) and [https://lab.dessimoz.org/blog/media/2020/04/omastandalone_cheat_sheet.pdf](https://lab.dessimoz.org/blog/media/2020/04/omastandalone_cheat_sheet.pdf).

2. Make sure that the submission script enters the folder into which the tarball was extracted, by either running the script from inside that directory or using the `cd` command appropriately.

3. The line to start the OMA all-against-all computation in the submission scripts is:

   ```
   $: bin/oma -s
   ```

   The -s option means stop after the all-against-all phase. Since this part can be parallelized, we recommend using job-arrays. For this you need to set the number of parallel jobs as an environment variable (export NR_PROCESSES=100) and use the job-array syntax in the submission script (e.g. in LSF: `bsub -J oma[1-$NR_PROCESSES] bin/oma -s`). For environments with limited runtimes/walltimes see [https://omabrowser.org/standalone/#advanced%20optimisations](https://omabrowser.org/standalone/#advanced%20optimisations).

4. Check whether the all-against-all computation is finished using:
This command will output a file formatted as:

```
Summary of OMA standalone All-vs-All computations:
--------------------------------------------------
Nr chunks started: A (D.D%)
Nr chunks finished: B (E.E%)
Nr chunks finished w/o exported genomes: C (F.F%)
```

Where the letters A, B, C, D, E and F represent numbers. Once the computations are completed, D should be equal to 0.0%, and both E and F to 100.0%

5. In the case where the jobs are finished but the all-against-all computation is still not complete, use the oma-cleanup and oma-compact commands before re-submitting.

```
$: bin/oma-cleanup
$: bin/oma-compact
```

These commands remove partially finished output files in the Cache/AllAll folder and zip all partial computations that are finished to one file, respectively.

6. Once the all-against-all computation has finished, the final step is the orthology calling. This step is more memory intensive, requires a single process, and can be called with:

```
$: bin/oma
```

Once the computation finishes, all results will be stored in the newly-created “Output” folder. In this folder there will be an EstimatedSpeciesTree.nwk file that contains a phylogenetic tree that can be visualized using a tree visualisation tool such as Phylo.io\(^5\). This is a distance tree based on the weighted average of the pairwise distances between sequences within the most complete OMA groups. This species tree is a rough estimate that is computed on the fly, and is not the final tree. It can be used as control to identify problems in the dataset but will not be as reliable as the tree inferred using the generated OGs later in this protocol. Therefore it is recommended to use the OGs to compute your own tree with external software. The OGs (OMA Groups)\(^9\) can be found in the OrthologousGroupsFasta folder, with each OG containing at least two species.

Usually for the construction of phylogenetic trees, one would select only OGs that contain at least X% of species, as described above with the parameter **Minimum Species Coverage**. The python script filter_groups.py from the git repository associated to this publication (http://doi.org/10.5281/zenodo.3786201\(^{16}\)) can be used to filter the OMA groups that contain at least X MIN_NR_SPECIES (replace `<MIN_NR_SPECIES>` and `<destination/directory>` with your own values):

```
$: python filter_groups.py --min-nr-species <MIN_NR_SPECIES> --input Output/OrthologousGroupFasta/ --output <destination/directory>
```

For example, we performed an analysis adding two yeast proteomes hypothetically not available in OMA and 18 available yeast proteomes. As a first step, we downloaded the precomputed data for the 18 proteomes from the OMA Browser and launched the computation after adding two separate proteomes. Once the computation finished, we selected 880 OGs with at least 18 species as a dataset to construct a tree. The data used in this example is available from Figshare\(^{14}\) in the Protocol_2 folder.

**Protocol 3: Downstream processing and tree inference**

Once all selected OGs are obtained from either of the first two protocols, the next step is to align all sequences within each OG. This can be done with any Multiple Sequence Alignment (MSA) tools, in this example we use MAFFT\(^{17}\). To run it, navigate to the folder containing the selected OGs and execute the following command, which runs mafft on each fasta file:

```
$: python mafft --input <destination/directory>/OrthologousGroupFasta/ --output <destination/directory>
```
This command generates a MSA file (.aln) for each OG. In order to infer the phylogeny of the species from these alignments, they have to be concatenated in a single alignment commonly referred to as supermatrix. We provide a python script to automate this, concat_alignment.py, available on http://doi.org/10.5281/zenodo.37862016. The --format-output option allows for choosing the output format of this concatenation, either fasta or phylip format (some phylogenetic software requires a specific format as input). Once the python script is downloaded or cloned, ensure that all alignments are in the same folder, and launch using the following command:

```bash
$: python concat_alignments.py <path>/<to>/<alignments>/*aln --format-output [fasta/phylip] > output
```

After computing the supermatrix, the phylogenetic tree can be inferred using any number of available software. We recommend choosing from the tools in Table 2, sorted by computing time and increasing precision.

**Tree visualization.** Most of the current phylogenetic inference tools provide trees in Newick format as output. In order to visualize such a tree, one can use the web-based viewer phylo.io (http://phylo.io) or other tree visualization tools (e.g. Figtree, phylogeny.io, etc).

In our examples, we inferred trees by aligning the sequences with MAFFT, concatenated the alignments using the aforementioned concat_alignments.py, and ran both IQTree and RaxML (Figure 3 and Figure 4). The data used for and the results from the computations can be found on Figshare14 (alignments in the data/Alignments folder, and trees in the tree folder). The exact code used for these examples is on http://doi.org/10.5281/zenodo.37862016.

**Discussion and conclusion**

With the wealth of genomic data available in an era of high-throughput sequencing, there is much to gain by making phylogenies from multiple gene families, rather than from one single gene. This can better represent the evolutionary history of a clade, because the evolutionary history of a single gene can be misrepresentative of a species evolutionary history. The principle of a supermatrix approach is that by combining multiple genes in one single phylogeny, one can combine the phylogenetic signal of multiple genes. One has to be careful however, to not combine “phylogenetic noise”. Orthologs selection is particularly important in this regard14, because errors in orthology inference could add genes that are not true orthologs, but rather paralogs descending from the ancestral genes by a duplication event. Thus, they would have a different evolutionary history than the sought species phylogeny.

OMA Groups (or Orthologous Groups) are a well suited set of orthologs for this kind of analysis, as the criteria used to compute these orthologs are stringent. They require that all genes are reciprocally closest genes in their respective species to all the other genes of the group and do not allow more than one gene in a species, thus excluding paralogs. In the Quest for Orthologs Benchmark12, the community benchmark for orthology inference, OMA Groups are consistently the most specific inference, although lacking in recall. As potentially missing genes are less detrimental to phylogenetic determination than false predictions are13, this is an appropriate choice of

---

**Table 2. Recommended software and example commands for computing a phylogenetic tree.**

Parameters, such as memory or threads, may vary based on size of dataset.

<table>
<thead>
<tr>
<th>Software for making phylogenetic trees</th>
<th>Example command</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQTree</td>
<td>iqtree -s alignment.phy -m LG -T 20 --mem 20G -seed 12345 -bb 1000</td>
</tr>
<tr>
<td>RaXML</td>
<td>raxml-ng --threads 20 --model LG+G8+F --seed 15826 --msa alignment.phy --all --bs-trees 100</td>
</tr>
<tr>
<td>PhyloBayes</td>
<td>pb_mpi -dc -gt -cat -dparm 4 -x 10 1000 -d alignment.phy alignment.chain1</td>
</tr>
</tbody>
</table>
Figure 3. Comparison of phylogenetic trees computed by IQTree (left) and RaXML (right). Trees were computed with 20 Yeast species present in OMA. The leaves of the trees are the UniProt 5-letter species codes. The following export options were used: Minimum species coverage: 1, Maximum nr of markers: -1 (uncapped). 168 marker genes were exported. Visualization done with phylo.io; different shades of blue show variations in topology.

Figure 4. Comparison of phylogenetic trees computed by IQTree (left) and RaXML (right), using additional species. Trees were computed with 18 Yeast species present in OMA, plus two additional proteomes (YEAST and FOMPI). The leaves of the trees are the UniProt 5-letter species codes. Genes used to compute the tree had to be shared by at least 90% of the species (minimum species coverage: 0.9, maximum number markers: -1). This represents 880 OGs. Visualization done with phylo.io; different shades of blue show variations in topology (in this case both trees have identical topology).
phylogenetics have already been published using OMA standalone or data from the OMA Browser, including those for archaea, sharks, spiders, worms, and insects, among others. 

This tutorial demonstrated how to carry out these different steps to infer a phylogenetic tree: orthology determination, sequence alignments, supermatrix construction, and phylogeny inference. It is designed to allow users to leverage the state of the art orthology inference provided by OMA Groups while reducing the necessary computation from their side, namely by relying on precomputed all-against-all alignments provided by the OMA Browser. We include code snippets and scripts that automate the whole process and ensure reproducibility of all phylogenetic analyses following this protocol. The tutorial is accompanied by practical examples with all data available on GitHub and Figshare.

For more information about the theory behind phylogenomics and the different methods, we refer the reader to recent reviews. In the context of this tutorial, we used well-established MSA and phylogenetic tree inference tools. For the more difficult cases however, it advised to carefully choose which tool to use, including some tools which are not mentioned here. For more information about existing tools the readers are invited to turn to the relevant literature. The protocols described here can be adapted to suit any other software compatible with standard data formats.

Data availability
The imported OG data and the OMA standalone software can be obtained from the OMA Browser (https://omabrowser.org), following instructions in this tutorial.

Figshare: Phylogenetic Tree Tutorial Example Data, https://doi.org/10.6084/m9.figshare.10780820.v4

Data are available under the terms of the Creative Commons Zero “No rights reserved” data waiver (CC0 1.0 Public domain dedication).

Additional python scripts (filter_groups.py and concat_alignments.py) are publicly available: https://github.com/DessimozLab/f1000_PhylogeneticTree

 Archived scripts as at time of publication: http://doi.org/10.5281/zenodo.3786201

License for scripts: MIT license

Software availability
OMA Browser available at: https://omabrowser.org/

Source code for OMA Standalone available from: https://github.com/DessimozLab/OmaStandalone/tree/v2.4.0

Archived source code of OMA StandAlone at time of publication: https://doi.org/10.5281/zenodo.3555595

OMA Browser license: Mozilla Public License version 2.

Acknowledgments
We would like to thank Natalia Zajac, Marion Brechet, and Katharina Pfaller for their useful feedback on the tutorial.

References


Denis Baurain

InBioS - PhytoSYSTEMS, Eukaryotic Phylogenomics, University of Liège, Liège, Belgium

This tutorial illustrates how to use the online OMA Browser and OMA Standalone package to assemble datasets of single-copy orthologous genes for phylogenomic inference. It covers two use cases: 1) all species of interest are present in the OMA Browser and 2) some species are not available and must thus be added by the user using OMA Standalone.

Generally speaking, the structure and writing are clear, the figures and tables are appropriate, and the data and code underlying the examples are well described and made available in two public repositories (Figshare and Zenodo). Therefore, I do not have major comments. However, I think that it is possible to easily improve a number of minor points in the manuscript. I provide a list below, simply sorted by order of occurrence. There are no line numbers, so I try to provide some context when needed.

- "16 rRNA ribosomal genes" should rather be "small subunit ribosomal RNA (SSU rRNA 16S/18S) gene" (no plural) to encompass the eukaryotic gene as well.
- "see 3 for common pitfalls in phylogenomics": Don’t you mean Philippe et al. (2017) “Pitfalls in supermatrix phylogenomics” instead?
- "OGs can have at most one representative gene per species": You should probably say a word about recently diverged in-paralogs (which do not harm species phylogenies) because exactly single-copy genes barely exist.
- "among more than 2300 genomes across the tree of life": You might provide a few numbers here: bacteria, archaea, non-animals, non-plants - or at least recall the rationale for genome inclusion in OMA Browser because it is quite incomplete in terms of species diversity.
- "using species in OMA in addition to other genomes not available in the database": It is not clear if OMA Standalone requires conceptual translations of the sequences (i.e., proteins) or if it can translate DNA sequences on the fly; if it cannot, please specify that one needs protein sequences and not genomes to make use of OMA.
"phylogenetic tree inference softwares": I am not a native English speaker, but "software" is uncountable; maybe you mean "software packages"?

IQtree, RaxML, Phylobayes, Figtree etc: Please try to use the correct mixture of uppercase and lowercase letters in each name across the manuscript: IQ-TREE, RAxML, PhyloBayes, FigTree.

A better download address for PhyloBayes would be: http://www.atgc-montpellier.fr/phylobayes/.

The interface of OMA Browser has slightly changed since the manuscript has been written. Two details that would need updating: "in the upper left of the home screen" is incorrect [for "Explore"] and "Compute" has now become "Download".

"To speed up the tree inference, set this value to below 1000 genes": One wonders how these <1000 genes would be prioritized and selected; this should be mentioned here.

"The name of the fasta file is either the species name or the 5-letter UniProt identifier...": This is not specific enough; what about whitespace, dots in strain names, etc? Are they allowed? Similarly, you should better define "other special characters" just below.

"...and not keeping other unrelated orthology inferences": This is a bit mysterious. I am aware that OMA can do more than what you demonstrate here, but this sentence should probably be slightly expanded with some examples.

"...use the job-array syntax in the submission script": This is the part where reproducibility is not straightforward. In particular, is there a master OMA instance controlling the parallelization or is this left to the user? From the code snippet provided, it is not clear, albeit the output of the oma-status command suggests that there is some orchestrator freeing the user from this chore. Please explain a bit more.

"we selected 880 OGs with at least 18 species as a dataset": Considering Figure 4, I understand your point: 90% of 20 species is 18. However, since you start with 18 genomes and add two "new" ones, the re-use of the number "18" is confusing to the user. As the analyses are done, I would not suggest to change this number, but rather to explain directly the 90% rationale in the text.

"This command generates a MSA file (.aln) for each OG.": This "for" loop is a very fine candidate for parallelization. Please remind the user that they could use a job array for this. Otherwise, they may wait a lot of time staring at the shell.

"there is much to gain by making phylogenies from multiple gene families": Here, I am strongly opposed to the words "gene families". The whole point of OMA is to identify single-copy (orthologous) genes, not gene families (i.e., including paralogues). Please use instead "from concatenations of multiple genes, rather than from one single gene" (or something similar).
"because the evolutionary history of a single gene can be misrepresentative of a species evolutionary history": This is said twice in the manuscript (and this is true) but not backed up by specific reasons. Please enumerate a few of them between parentheses.

In the legend of Figure 3 and 4, please provide the evolutionary model used. I know it is in Table 2, but somewhat hidden in the command line arguments of the software packages.

"in this case both trees have identical topology" [Figure 4]: This is correct, but this topology is widely different from the two topologies of Figure 3. This shows that phylogenetic artefacts are at play (which is well known for yeasts...) and this might worth mentioning in the sentences about the caveats of phylogenomics. In other words, it is a pity that this use case does not provide a clear phylogenetic solution for the selected genomes.

References

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Yes

Are sufficient details provided to allow replication of the method development and its use by others?
Partly

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Phylogenomics, comparative genomics, software engineering

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.
Jianbo Xie
National Engineering Laboratory for Tree Breeding, College of Biological Sciences and Technology, Beijing Forestry University, Beijing, China

The manuscript provides useful tools to make use of OMA Orthologous Groups to infer a phylogenetic species tree. This manuscript is an interesting study, but I found some shortcomings that the authors should be improved before the manuscript can be considered for indexing. Some marker gene databases, such as Qiyun Zhu et al. (2019), have been published - could you integrate the database into your tool? Or describe the advantage of your tool?

Minor:

1. The introduction section should include some similar tools such as PGAP, Orthomcl.
2. Please discuss the advantage of OMA compared with other tools.
3. Could you integrate the steps of analyses to one steps with multiple parameters?

References

Is the rationale for developing the new method (or application) clearly explained?
Yes

Is the description of the method technically sound?
Yes

Are sufficient details provided to allow replication of the method development and its use by others?
Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.
Reviewer Expertise: Genetics; evolution

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com