Phylogeny Tutorial

Course: Applied Bioinformatics (BBT045)

Teacher: Vi Varga

Date: 14.02.2023

Introduction

In this exercise you will practice your newly acquired skills in phylogenetic inference and "tree thinking", by analyzing the evolutionary history of a gene family.

You will (roughly) follow the phylogenetic analysis workflow discussed during the lecture, starting with collecting the data necessary for running the analysis (in the form of homologous protein sequences from different species), through the interpretation of results (i.e., comparing species and gene trees).

For this tutorial, you will be working with a protein from *Trichomonas vaginalis*. *T. vaginalis* is a parasitic protist that causes the sexually transmitted infection Trichomoniasis. While infection is generally asymptomatic, complications can include up to infertility or sterility. This particular protist belongs to the eukaryotic supergorup Metamonada, which is comprised of 4 primary phyla (Anaeramoebidae, Parabasalia, Preaxostyla and Fornicata) and contains a wide variety of parasites, commensals, and free-living organisms.

Setting up the environment

conda environment & file structure

Before we begin, we will set up a new conda environment in which we will install software for use on the command line while following this tutorial. If you open multiple terminal windows while working on this exercise, please make sure to activate the conda environment in each one. In general, it is a good idea to get into the habit of using conda environments, and activating the relevant environment directly after opening the terminal (or, directly after logging on to the server). This way, you won't accidentally try to run software that isn't installed in your base conda environment.

It's also good practice to double-check the environment you have activated prior to installing any new software with conda. Un-installing programs that you've accidentally installed takes much, *much*, ***much*** longer than you'd expect! (Think, "need to leave the computer running overnight" kinds of situations. (。_。))

When should you create a new conda environment? Generally, it's a good idea to have a dedicated conda environment for any project you're working on. That way, at the end of the project, you can synthesize all the information related to version numbers of the programs you used quite quickly. However, you may at times need to create environments for specific programs - this is particularly common for older programs that may require older-than-standard versions of some dependency packages, particularly programming language versions (ex.: programs written with Python 2.X will not be compatible with a conda environment running Python 3.X, and visa versa).

Create a conda environment:

```
# working on the server
conda create -n phylo-tutorial-env python=3.9
# feel free to name the environment whatever you like
# just try to make sure your name is descriptive, so you can remember what it was
for
# the `python=3.9` is necessary to ensure a current version of Python is
installed, instead of an outdated one
# You'll see a lot of scrolling text, and then need to confirm creation of the
environment with "y"
# once setup is done, activate the environment:
conda activate phylo-tutorial-env
# you can deactivate the environment later with:
conda deactivate
# remember not to use that above command in your base conda environment!
```

Please also make a directory on the server in which you will store your files for this tutorial, as well as the exercise to follow. You would be surprised how swiftly the number of files you're using gets out of hand, so try to develop good habits from the beginning! For example, it's good practice to have a `bin/` directory in your home directory, where you store executable files and the like for programs that you cannot simply install via `conda`.

```
# in the directory where you have your files for Applied Bioinformatics
# for ex.: a directory named AppliedBioinfo/
mkdir PhyloWorkflow/
cd PhyloWorkflow/
mkdir Exercise/ Tutorial/
# note that `mkdir` can take multiple arguments
# so we've just created both directories with one line of code
```

Above, I used my personal preferred naming convention, but feel free to use whatever file names you wish, *as long as they're descriptive*.

Software installations

Now that we have our directories and `conda` environment set up, let's install the relevant software we will all be using during this tutorial.

```
# if you have not already done so, activate the environment:
conda activate phylo-tutorial-env
# install MAFFT MSA software
conda install -c bioconda mafft
# install IQ-TREE phylogenetic tree generating software
conda install -c bioconda iqtree
```

Obtaining & Exploring Data

Preliminary data exploration

For this tutorial, we will be using the XP_001322682.1__Tvag.fasta file. Take a look!

1. What kind of FASTA file is it?

> *Fill in your answer!*

2. What is the protein ID?

> *Fill in your answer!*

This is a RefSeq protein, which means it is considered good quality. To quote the NCBI, the RefSeq database is a "comprehensive, integrated, non-redundant, well-annotated set of reference sequences including genomic, transcript, and protein." (Source: https://www.ncbi.nlm.nih.gov/refseq/)

In *T. vaginalis*, XP_001322682.1 is predicted to function in the cytosol, though paralogs of the protein in *T. vaginalis* are known to function in the mitochondrion (Smutná et al. 2022)[^1].

[^1]: *Technically speaking*, *T. vaginalis* doens't actually have a mitochondrion, *per se*. *T. vaginalis* and all other organisms within supergroup Metamonada have "Mitochondrion-Related Organelles" (MROs), which are extremely functionally reduced mitochondria. *Monocercomonoides exilis*, a member of supergroup Metamonada, is actually the only known eukaryote to completely lack a mitochondrion!

Finding homologous sequences

In order to find homologs of this gene to use in a gene tree, we're going to use NCBI BLAST. While command-line BLAST technically exists, overall it is much simpler and faster to use the web browser version. This is particularly the case as the conda installation of NCBI BLAST doesn't work very well.

3. What *type* of BLAST do we need to run? Explain your reasoning.

> *Fill in your answer!*

From the NCBI BLAST homepage (https://blast.ncbi.nlm.nih.gov/Blast.cgi) select the appropriate BLAST algorithm.

You can BLAST the sequence in one of two ways: either you can copy the sequence into the search box, or you can use the gene name (XP_001322682.1). Since this is the name of the protein in the NCBI database, it is possible to search for BLAST hits using only the protein ID - if this was a protein from an organism you sequenced, with no official name in the NCBI database, you would only have the option of performing a sequence-based BLAST.

Go ahead and BLAST the gene, and maybe take a short coffee break. BLAST can sometimes take a minute.

ヾ(@⌒ー⌒@)ノ

The first hit *should* be our protein - go ahead and check! Do you notice anything about the quality of the BLAST hits?

> *Fill in your answer!*

In the light green bar above the search results labeled "Sequences producing significant alignments" you'll find a "Download" drop-down menu. Select "FASTA (complete sequence)" and you'll download a file named seqdump.txt. Rename this file to something meaningful (ex.: XP_001322682.1__MSAprep.fasta) and move it to your Tutorial/ working directory. You can do this using copy/paste into `nano`, an SSH file transfer program like FileZilla, or `scp` (`scp XP_001322682.1__MSAprep.fasta USERNAME@phoebe.math.chalmers.se:/home/PATH/TO/WORKING/DIRECTORY` where you can fill in your username and the path to your working directory).

4. How many sequences are in the file?

> *Fill in your answer!*

Cleaning the data

Generally, when you compile protein sequences to use in an analysis, you want to *clean* the data in some way. A few common data transformations include: capitalizing all letters in sequence lines, editing header lines, removing non-standard characters from sequence lines, and conversion between multi-line and single-line FASTA format.

5. Why might it be important to clean the data in this way?

> *Fill in your answer!*

For sake of time, I've prepared a cleaning script using Python (clean_MSA_seqs.py). You can use it from the command line like so:

```
# model usage:
python clean_MSA_seqs.py input_fasta
# Adapt the command above to your file names!
```

I tend to comment my code pretty thoroughly, so it should be quite readable, but let me know if you have any questions. If you're up for a challenge, see if you can write something like this in R! (But finish the tutorial first - come back to this later if you still have the time.)

6. Compare the original and cleaned files. Do they contain the same number of sequences? The same number of characters? What changed, if anything? Why do you think this is?

> *Fill in your answer!*

Since our sequences are now cleaned, we can move on to generating the alignment.

Multiple Sequence Alignment

Generating the MSA

In order to generate the MSA, we're going to be using the MAFFT software. You can find more information on this program at the links below:

- Homepage: https://mafft.cbrc.jp/alignment/software/
- Manual page: https://mafft.cbrc.jp/alignment/software/manual/manual.html

7. Take a look at the MAFFT manual page. Which algorithm do you think best suits our purposes? Why?

> *Fill in your answer!*

Now, go ahead and create the MSA:

```
mafft --localpair --maxiterate 1000 --amino XP_001322682.1__MSAprep_CLEAN.fasta > XP_001322682.1__MSA.fasta
# Adapt the command above to your file names!
# `--localpair --maxiterate 1000` tells MAFFT to use the L-INS-i algorithm
# `--amino` tells MAFFT that the input is a protein FASTA
```

Viewing the MSA

There are many different tools that you can use to view an MSA. I've provided a few examples below:

- Web-based tools:
  - As with most aspects of bioinformatics, there are tools available on the web in order to view MSAs. As is often the case with web-based programs, though, their scope is rather limited (especially for the tree programs).
  - EMBL-EBI MView: EMBL-EBI provides a web-based tool where you can upload an MSA, and see the results. Access it from here: https://www.ebi.ac.uk/Tools/msa/mview/
  - NCBI Multiple Sequence Alignment Viewer: The NCBI provides a web-based MSA viewer, which you can access from here: https://www.ncbi.nlm.nih.gov/projects/msaviewer/
- Stand-alone software:
  - Software designed for phylogenomics analysis provides far more flexibility than web-based tools, though this of course comes with the trade-off of requiring installation, and taking up space on your hard drive.
  - AliView: (My personal favorite) This program from Uppsala Univeristy provides smooth viewing and editing of MSAs. Find more information on it here: https://ormbunkar.se/aliview/
  - MEGA-11: The MEGA software suite allows a huge range of phylogenomics analysis tools. You can create MSAs, edit alignments, visualize phylogenetic trees, perform bootstrap testing... All from within a GUI window! Find it here: https://www.megasoftware.net/

For now, we will use the web-based tool provided by EMBL-EBI (). Upload your MSA file, give it a minute to process, and then take a look at the results!

8. Do you notice any patterns?

> *Fill in your answer!*

Creating the tree

We will be using the IQ-TREE software to generate the gene tree. You can find more information about this program at the links below:

- Homepage: http://www.iqtree.org/
- Manual: http://www.iqtree.org/doc/iqtree-doc.pdf

Go ahead and run the command you see below - there will be a lot of text printed to the screen, but don't worry about redirecting it to a file to look at later, because all of it will also be printed to the log file generated automatically by IQ-TREE. *This will take a few minutes*. So feel free to grab a coffee, take a short break!

♪(^▽^*)

Once you're ready, feel free to read the little chunk of text below the code block here - it'll provide a little more information this type of analysis.

```
iqtree -s XP_001322682.1__MSA.fasta --prefix XP_001322682.1__MSA_IQ -m LG+R5 -seed
12345 -wbtl -T AUTO -ntmax 8
# Adapt the command above to your file names!
# -s is the option to specify the name of the alignment file that is always
required by IQ-TREE to work.
# -m is the option to specify the model name to use during the analysis.
# The special MFP key word stands for ModelFinder Plus, which tells IQ-TREE to
perform ModelFinder
# and the remaining analysis using the selected model.
# Here, the model to use has been pre-selected: LG+R5
# To make this reproducible, need to use -seed option to provide a random number
generator seed.
# -wbtl Like -wbt but bootstrap trees written with branch lengths. DEFAULT: OFF
# -T AUTO: allows IQ-TREE to auto-select the ideal number of threads
# -ntmax: set the maximum number of threads that IQ-TREE can use
```

Note that a typical tree-finding process is quite a bit longer than what you did here. IQ-TREE has a specific argument `-m MFP` that calls a process called Model Finder Plus which tests many, *many* different tree models, and finds the one that best fits the data. (Don't worry about what these models are - that's beyond the scope of this class. Suffice to say, it's complicated statistics.) I ran this analysis with `-m MFP` while preparing this exercise, and even for such a small dataset (only 100 sequences), the process took 42 minutes! Clearly, not something we could all do together in class. Tree finding is a complex, computationally demanding process, but is a crucial part of phylogenetic reconstruction, and not the step where you should try to spare CPU hours.

Visualizing Trees

There are many different tools that you can use to visualize a phylogenetic tree. I've provided a few examples below:

- Web-based tools:
    - As with most aspects of bioinformatics, there are tools available on the web in order to visualize phylogenetic trees. As is often the case with web-based programs, though, their scope is rather limited (especially for the tree programs).
    - ETE Toolkit: The ETE Toolkit is available as a Python package, but they also have a web server where you can visualize your trees, here: http://etetoolkit.org/treeview/
    - NCBI Tree Viewer: The NCBI provides a web-based phylogenetic tree viewer, which you can access from here: https://www.ncbi.nlm.nih.gov/tools/treeviewer/
- Stand-alone software:

    - Software designed for phylogenomics analysis provides far more flexibility than web-based tools, though this of course comes with the trade-off of requiring installation, and taking up space on your hard drive.
    - FigTree: (My personal favorite) This program allows you to open trees and edit components of its visualization, before exporting in a variety of different file types (PNG, JPEG, SVG, etc.). It's a JAVA-based application, so if you have Java installed on your computer, no further installation processes will be necessary to open FigTree. Find it here: http://tree.bio.ed.ac.uk/software/figtree/
    - MEGA-11: The MEGA software suite allows a huge range of phylogenomics analysis tools. You can create MSAs, edit alignments, visualize phylogenetic trees, perform bootstrap testing... All from within a GUI window! Find it here: https://www.megasoftware.net/
- Packages built for bioinformaticians:
    - There are plenty of packages/libraries available for the visualization of phylogenetic trees, built to work with the programming languages most used by bioinformaticians: Python and R. These editing tools have a higher learning curve, since you need to code to change aspects of the tree, but they also allow far more flexibility than either web-based tools or stand-alone software.
    - Python:
        - The ETE Toolkit (mentioned above) is actually primarily a Python package. Find it here: http://etetoolkit.org/
        - Biopython is a whole suite of Python packages for bioinformattics analysis, so of course, they have their own package for working with phylogenetic trees, `Phylo`. FInd it here: https://biopython.org/wiki/Phylo
    - R:
        - The `ape` library in R can be used to visualize and edit phylogenetic trees. It can be installed the usual way (`install.packages(ape)`). The creators of the package have provided a great tutorial, which you can find here: http://ape-package.ird.fr/misc/DrawingPhylogenies.pdf
        - The `ggtree` library was created by Bioconductor, which provides a suite of R tools for bioinformatics analysis. The program is built to work like `ggplo2`, except for trees. You can find more information (including installation instructions, which are a bit different for Bioconductor packages) here: https://bioconductor.org/packages/release/bioc/html/ggtree.html

Feel free to explore these programs and packages at your leisure, and find what works best for you. For now, for the sake of time, I have written an R script using the `ape` library that you can use to visualize your results, named `visualize_PhyloTree_base.R`. Fill in the file name and path to your files, and you should be good to go! (You may have to install some R packages - let me know if you get stuck!)

For the input file for this script, use the FILENAME.treefile file output by IQ-TREE. This file contains the phylogenetic tree generated from the MSA in NEWICK format. A Newick tree is a 1-line simple text representation of a phylogenetic tree, that should be recognized by any phylogenetic tree visualization software.

9. Take a look at the tree that you have generated. What do you notice? Are there any interesting patterns?

> *Fill in your answer!*

Comparing trees

At this stage of the tutorial, two paths are available: familiarizing yourself with some web tools, or performing alignment editing. Either way, the goal of this final section is to compare trees generated in different ways. Please read through the descriptions of these options, and select the one that is right for you.

- Web Tools:
    - Pick this option if you have <15 minutes left until we look over the answers together, and/or if you are struggling with this material a bit.
    - This option is less technically demanding, which means you can spend more of your time working with the concepts.
    - You will have the opportunity to test some web-based tools for phylogenetic analysis,
- Alignment editing:
    - Pick this option if you feel confident on the conceptual parts of tree evaluation, and you have >15 minutes left until we go through the exercise.
    - You will try your hand at editing alignments, in order to improve the gene tree.

Once you have completed your selected option and generated a new tree, answer the following question:

10. How does the cleaned tree compare to the original version?

> *Fill in your answer!*

Web tools for phylogeny

Every passing year, bioinformatics becomes a larger and more significant part of biology. This of course presents some problems for those biologists that were trained as *biologists*, and not bioinformaticians. Dry and wet lab skills are not the same, and not everyone has the time or means to learn a vastly different set of skills.

Fortunately, as is often the case with bioinformatics, tools are freely-available on the web that allow you work with phylogenomic data. These tools are generally more limited in scope than their command-line counterparts - they have more limitations regarding dataset size and the degree to which you can fine-tune your search/request. However, they are still a great tool even for bioinformaticians, if the query is straightforward and involves a small dataset. Sometimes it's simply simpler and easier to press 3 buttons than to write an entire script.

Sometimes these tools are available as stand-alone programs from the same organization that made the software. MAFFT, for example, allows you to create an MSA on their website, here: https://mafft.cbrc.jp/alignment/server/

For our purposes, however, we will be using the collection of MSA software made available by EMBL-EBI, here: https://www.ebi.ac.uk/Tools/msa/

As you can see, a variety of aligners are available on this website, including (but certainly not limited to): Clustal Omega, MAFFT and MUSCLE, three of the most commonly-used MSA softwares. Note that, quite usefully, these programs all generate not just an alignment, but a visualized tree for you, too. Feel free to download a Newick tree file and visualize it if you wish, but it's also fine to just look at the ones the website generated for you.

Try out at least two aligners on the website (whichever ones strike your fancy), and compare the results to the trees we made with MAFFT on the server.

Editing the MSA

If you finish all of the above with time left, try your hand at editing an MSA!

As we discussed during the lecture, cleaning up an MSA is an important part of a phylogenetic analysis workflow. Test out some of the strategies we discussed on the MSA you made, and see how if anything changes!

In order to edit an MSA, you have two options:

1. Install AliView on your local computer (*not* the server). This program will allow you to examine and edit alignments manually. Find it here: https://ormbunkar.se/aliview/
2. Create a conda environment (as shown below), and play around with the settings of a MSA editing software, for ex.: TrimAl (http://trimal.cgenomics.org/introduction) or CIAlign (https://cialign.readthedocs.io/en/latest/pages/introduction.html). If you choose this option, please make sure to visualize the MSAs you create in a web browser, so that you can see for yourself the differences in the alignment.

Note that while Option 1 does require you to install software, it is likely the simpler option, especially for those with less experience coding. It is also more interactive.

AliView

Install AliView by following the instructions for your operating system, at: https://ormbunkar.se/aliview/

Then do the following:

- Open the program
- Navigate: File → Open File → Nagivate to and select your MSA to open it in the program
- Turn on Edit Mode: Edit → Edit mode (should have a check mark if edit mode is turned on)
- Select portions of the alignment to remove: Select & drag your cursor along the position numbers at the top → Edit → Delete selected
    - You can also try a variety of different editing options within the Edit menu (ex.: Delete gap-only columns)
- Save the new MSA to a new file with: File → Save as Fasta
- Visualize the gene tree again with the new MSA, and compare it to the species tree and other gene tree(s). What has changed (if anything)?

Editing using conda

Install alignment editing software within that environment:

- TrimAl installation with conda: https://anaconda.org/bioconda/trimal
- CIAlign installation instructions: https://cialign.readthedocs.io/en/latest/pages/installation.html

```
# if you have not already done so, activate the environment:
conda activate phylo-tutorial-env
# install TrimAl
conda install -c bioconda trimal
# or install CIAlign
# note that this program has dependencies we need to install first!
conda install matplotlib
conda install numpy # numpy may install with matplotlib, so this may be
unnecessary
conda install scipy
conda install -c bioconda cialign
```

And use the options found in the program manuals to play around with editing the alignments.

- TrimAl command line usage manual: http://trimal.cgenomics.org/use_of_the_command_line_trimal_v1.2
- CIAlign command line usage manual: https://cialign.readthedocs.io/en/latest/pages/usage.html

```
#TrimAl example usage:
trimal -in XP_001322682.1__MSA.fasta -out XP_001322682.1__MSA_trim1.fasta -
gappyout
# CIAlign example usage:
CIAlign --infile XP_001322682.1__MSA.fasta --outfile_stem
XP_001322682.1__MSA_trim2 --clean
```

Citations

> Smutná, T., Dohnálková, A., Sutak, R., Narayanasamy, R. K., Tachezy, J., & Hrdý, I. (2022). A cytosolic ferredoxin-independent hydrogenase possibly mediates hydrogen uptake in Trichomonas vaginalis. *Current Biology*, *32*(1), 124-135.e5. https://doi.org/10.1016/j.cub.2021.10.050