Practical: Phylogenetic analysis

In this practical, you will learn to create a phylogenetic tree from an alignment and visualise it in different tools.

Overview

1. Phylogeny on alignment from morning session using different methods:
   - Simple Neighbor-joining (MEGA)
   - Maximum parsimony with bootstrap (MEGA)
   - Maximum likelihood (IQTREE)
   - Approximate maximum likelihood (fasttree)
2. Visualisation of tree with Microreact, iTOL and ETE3

Before you start

Stand alone software

For this exercise, you will need the following stand alone software:

- Figtree (http://tree.bio.ed.ac.uk/software/figtree/)
- MEGA ≥v10 (https://www.megasoftware.net)
- A Web browser (in order to use Microreact and iTOL and to visualize a .png image)

Exercise data

> **Note:** If you have done these steps for morning session, there is no need to redo them

All required files for the practicals should be available on your course laptops.

Important: Create a subfolder within the repo folder for each tool you are running on the command line, so the output of each tool is in its own folder.

Part 1 - Create a phylogenetic tree

In Part 1 of this practical, we will create phylogenetic trees using different methods.

Exercise 1: Create a Neighbour-joining tree

Neighbor joining (NJ) is a bottom-up (agglomerative) clustering method for the creation of phylogenetic trees, created by Naruya Saitou and Masatoshi Nei in 1987. Neighbour joining takes a distance matrix, which specifies the distance between each pair of taxa, as input. The algorithm starts with a completely unresolved tree, whose topology corresponds to that of a star network, and iterates over several deterministic steps, until the tree is completely resolved, and all branch lengths are known.

Here we use the *MEGA* software to create a NJ tree.

> This requires file `16s_alignment_mafft.fasta` from the `16s_data` folder from the morning session or from the `Precomputed_Data/day7/16s_data` data folder.

1. Open MEGA on your Computer
2. Drag-n-drop the `16s_alignment_mafft.fasta` file on the window.
3. Choose `Analize` because the file is already aligned
4. Choose `Nucleotide Sequences`
5. Choose `Yes` when asked if these are protein-coding sequences because we are using the full 16S sequence.
6. Choose `Standard`
7. Click on Phylogeny and choose a `NJ phyologeny`
8. Choose `yes` for the current file
9. Check the parameters and press `OK`

A tree showing the phylogenetic relationship appears. Read the caption and decide if you agree.

Questions:

1. Which species cluster together?
2. Which genomes are most closely related to the *S. aureus* genomes?
3. Is this a reliable tree?

Exercise 2: Create a Maximum parsimony tree with 100 bootstrap replicates

We again use the *MEGA* software to create a Maximum parsimony tree.

1. Click on Phylogeny and choose a maximum parsimony phyologeny
2. Choose `yes` for the current file
3. Check the parameters and press `OK`. Make sure you enter `100` bootstrap replicates in the “Test phylogeny” field.
4. When the tree is done, it is visualized in `Topology only` mode. Click the `Topology only` button in the menu bar to switch to the tree with actual branch lengths.

A tree showing the phylogenetic relationship appears. Read the caption and decide if you agree.

Questions:

1. What do the numbers mean?
2. Which groupings are most reliable based on this data?

Exercise 3: Remove recombinant sites from SNP matrix

Now we switch to the command line and to the core genome SNPs from the *E. coli* dataset from the morning session.

> This requires file `core.aln` from the `snippy` output folder from the morning session or from the `precomputed/snippy_on_reads` data folder

Load the `phylo` environment using the following command:

```
conda activate phylo
```

And cd into the directory with the course data:

```
cd <your_course_path>/day7
```

To obtain a good alignment of SNPs, we need to take care of regions of putative recombination. We use gubbins to remove these regions from the SNP matrix.
To do so, we first need to strip odd characters from the matix using a sed command because gubbins doesn't like these:

```
mkdir gubbins
cd gubbins
sed -r 's/::.*//' ../snippy/core.aln > core_stripped.fasta
```

This creates a copy of the matrix with the odd characters stripped.
We can then run the gubbins command:

```
run_gubbins.py core_stripped.fasta -c 8
```

This will output a purged SNP matrix with fewer sites but the same number of taxa called `core_stripped.filtered_polymorphic_sites.fasta`

Exercise 4: Create a maximum likelihood tree with IQTREE

We use the very versatile software *IQTREE* to produce a high-quality maximum likelihood tree from the purged SNP matrix.

> This requires file `core_stripped.filtered_polymorphic_sites.fasta` from the `gubbins` output folder.

Run iqtree:

```
cd ..; mkdir iqtree; cd iqtree
iqtree -s ../gubbins/core_stripped.filtered_polymorphic_sites.fasta -m
TEST+ASC -T AUTO --threads-max 8 -pre ML_iqtree -mem 8GB
```

This creates the output file `ML_iqtree.treefile`, which is a *NEWICK* format tree file. To use it further, we need to make a copy with extension .nwk:

```
cp ML_iqtree.treefile ML_iqtree.treefile.nwk
```

Exercise 5 (optional): Create a fast approximate maximum likelihood tree with fasttree

We use the very fast software fasttree to produce a fast approximate maximum likelihood tree from the purged SNP matrix.

```
cd ..; mkdir fasttree; cd fasttree
fasttree -nt -gtr
../gubbins/core_stripped.filtered_polymorphic_sites.fasta >
core_fasttree.nwk
```

This creates the output file `core_fasttree.nwk`, which is a *NEWICK* format tree file.

Part 2 - Visualize the phylogeny

In this part, we will visualize the obtained tree using different methods.

Exercise 1: Visualize a tree using Microreact

Microreact is a tool for open data visualization and sharing for genomic epidemiology. It is freely available and is widely used in public health data analysis.

To get your tree visualized and annotated in Microreact, do the following:

1. Go to https://microreact.org and watch the video if you want to
2. Click on `upload`
3. Choose or drop your tree file
4. Click continue
5. Find out how to display the labels.
6. Re-root your tree with the `Reference`, and samples `Ec007` and `Ec001` as outgroup (use the right-click menu on their common ancestor branch)
7. Add the metadata file `metadata.tsv` (available in the `metadata` folder) to the tree by linking the tree tip labels (`id`) to the `key` column
8. Add color columns for `Region` and `Travel` using the `Metadata blocks` button
9. Add a map by clicking on the pencil (Add or edit panels) on the top right. Use the `lat` and `long` columns from the metadata for the coordinates.
10. Download the map as a `.png` and the tree as a `.svg` file.

Questions:

1. Which isolates are closely related and are therefore probably part of the same outbreak?
2. Do they all come from the same region?

Exercise 2: Visualize a tree using iTOL

iTOL is an online tool for visualizing phylogenies and related metadata. The tool is free to use, but for saving your annotations, paied subscription has been introduced a few years ago.
The tool is frequently used for close to publication ready phylogenetic trees.

To get your tree visualized and annotated in iTOL do the following:

1. Open iTOL on https://itol.embl.de

2. Click on `Upload`
3. Upload or drop your newick tree file by clicking `choose file`
4. Re-root your tree with the `Reference` and samples `Ec007` and `Ec001` as outgroup (use the submenu `Tree structure`)
5. Use the provided templates `dataset_color_strip_template.txt` and `dataset_color_gradient_template.txt` from the `metadata` folder to add annotations:
   a. Go to `Datasets` in the Control panel
   b. Click on `Upload annotation files`
   c. Choose the two files and click `upload`

   > *Note*: More templates can be downloaded from https://itol.embl.de/help.cgi#annoTemplate

6. Export and save your tree with annotations as a `.pdf`

Exercise 3: Visualize a tree using the python library ETE3 (for command line users only)

ETE3 is a python toolkit to do phylogenetic analysis and visualize phylogenetic trees.

Here we have prepared a basic script to plot our tree, called `ete3_phylo.py` (available in the `scripts` folder).

> This script is dependent on the correct folder structure and file names, namely: `iqtree/ML_iqtree.treefile.nwk` and `metadata/metadata.tsv` as well as the script in the `scripts` folder.

To run the script, open a console and type the following commands:

```
python scripts/ete3_phylo.py
open mytree.png
```

Open the file mytree.png and compare it to the figures obtained in other tools.

Inspect the script and try to answer the following questions:

1. What does the `my_layout()` function do?
2. Where are the colors for the regions defined? Can you change one of them?
3. What are the rectangles colored by?

If you have some extra time, try to change some of the settings in the script.