

## Biology 545 Principles of Systematics

### Laboratory 4: Assessing Nodal Support on Trees

In this exercise, we will assess support on trees that we generated last lab as part of our BEAST analyses. We will also demonstrate how to generate bootstrap replicate ML trees using RAxML and explore generating consensus trees in order to generate a point-estimate of topology from a distribution. Finally, you will gain familiarity with FigTree, a widely used program for displaying and editing trees.

#### Section 1: Bootstrap Support

Login to the server using the **ssh** command in a terminal or a SSH client (e.g. PuTTY)

```
ssh username@jayne.ibest.uidaho.edu
```

#### Section 1a – RAxML:

Previously, we have used RAxML to estimate the maximum likelihood tree given a multisample fasta alignment. Now we will use RAxML again, to generate 100 bootstrap replicate trees, and use PAUP\* and DendoPy to generate majority-rule consensus tree from our bootstrap replicates.

Load the RAxML module on the cluster.

```
module load raxml
```

Copy the `biol545ParDat.fasta` file that we used last lab to a new directory (`cp old_fasta_file new_file_location`).

Our new RAxML command is very similar to the one that we used to estimate our ML tree, but now we include the `-b` option for random bootstrap seed, and the `-#` option to specify our number of bootstrap replicates.

```
raxmlHPC -s biol545ParDat.fasta -n raxml_bootstrap -p 12345 -b  
12345 -# 100 -m GTRGAMMAI
```

This will perform 100 bootstrap replicates, and save all 100 trees in newick format to a single file named `RAxML_bootstrap.raxml_bootstrap`. This should take around 5 minutes to run.

## Section 1b - PAUP:

While still connected to the server, load the PAUP\* module.

```
module load paup
```

We will read in the trees from the bootstrap replicates, summarize them via a consensus tree, and apply nodal support values to the tree.

To load in a treefile without data, you first need to load in the dataset. Execute the original datafile. Load the trees into memory by typing:

```
gettrees file= RAxML_bootstrap.raxml_bootstrap storeBrLens=yes;
```

Generate a majority-rule consensus of your trees, and save it to a file by typing:

```
contree/ majrule=y strict=n le50=y grpfreq=n file=majrul.contree.boot.tre;
```

Copy your majority-rule consensus tree from PAUP to your local machine using the scp command in a terminal or an SFTP client to copy the file from the server to your local machine. Note that the below script is all one line.

```
scp username@jayne.ibest.uidaho.edu:/mnt/ceph/username/pathto  
file/majrule.contree.boot.tre .
```

**Answer questions 1-2 in the assignment.**

## Section 2: Summarizing BEAST Results

### 2a. Investigating MCMC results using Tracer

The BEAST runs you performed last lab produced two output files, one '.log' file and one '.trees' file. These are the posterior distributions of parameter estimates and topologies, respectively. It is possible to have another distribution of trees with branch lengths in units of substitutions/site, but by default BEAST produces a chronogram with branches in units of substitutions/site/unit time. This tree is *ultrametric* – the distance from the root to every tip is a fixed value.

Download tracer. <https://github.com/beast-dev/tracer/releases/latest>

Start Tracer. Import the trace (.log) files into memory (one with data and one sampling only from the prior). You can select parameters of interest and view their marginal distributions, trace, and compare parameters among runs. You will learn more about these results in your assignment.

**Answer questions 3-5 in the assignment.**

## **2b. Constructing a Maximum Clade Credibility tree using TreeAnnotator**

BEAST includes another program, TreeAnnotator, that will summarize a posterior distribution of trees. Start the program. Select your '.trees' file from your beast run (the one run with data) as input. For 'posterior probability limit,' type 0.5. For 'node heights,' select "Median heights." The chain length of the analysis was 5,000,000 generations, and we want to remove the first 10% of this as burnin. Enter 10 in the burnin box. Enter 'MaxCladeTree.tre' as the output file name.

Click "Run" to get a point-source of topology with nodal support. Load the results into FigTree, check the "Node Labels" box, and select "posterior."

**Answer question 6 in the assignment.**



5. Look at the parameter 'uclStdev' from your normal run and your prior-only run. This parameter describes the amount of variation in the substitution rates across branches (To do this, highlight both logs – do not include the line titled 'combined' – and then click on the parameter). Compare the parameter estimates and 95% HPDs; also, look at the marginal probability distributions (click on colour by 'tracefile' in the lower right hand corner and put a legend in the viewing frame to easier distinguish each trace). What does this tell you about your analysis?

6. Compare the posterior probabilities vs. the bootstrap values between your two analyses. Are they different? Why? (Comment on the differences between these two approaches. Focus on topological dissimilarity, differences in bootstrap support vs. posterior probability support, and any other differences. Hopefully, each analysis will result in the same tree; if this is not the case, comment on why you think they produced different results.)