Biology 545
Phylogenetics

Laboratory 3: Likelihood-based Phylogenetic Approaches

This exercise will cover likelihood-based approaches in phylogenetics. These methods use information from the data and model of evolution in order to estimate a topology. Likelihood-based methods currently are in wide use; they have been shown to be cosistent as long as an adequate model of sequence evolution is selected. The tradeoff is that likelihood-based methods have to calculate a likelihood at each step of the analysis, a non-trivial mathematical hurdle that slows down analysis.

We will be performing two searches, one using maximum likelihood with RAxML and IQ-TREE, and another using Bayesian methods with BEAST. In the next lab, we will build on these analyses by comparing topologies and evaluating nodal support.

Section 1: Maximum-Likelihood Inference

1a. Background

Maximum likelihood is a statistical approach developed by Sir Ronald Fisher in 1922. Maximum-likelihood inferences proceed by attempting to find the parameters with the highest *likelihood*; in phylogenetics, we look to find the tree that confers the highest probability on the data (an alignment), given the model. This is the *maximum-likelihood estimate* of topology. MLEs have many appealing properties, not least of which is the property of statistical consistency; the estimated tree becomes closer to the true tree as the amount of data (in this case, the number of characters) approaches infinity.

Searching for the ML tree is analogous to other hill-climbing approaches in phylogenetics. As discussed in lecture, we can visualize tree space as a landscape with the vertical axis representing the likelihood of the tree. From any particular starting point (i.e., starting tree), the goal is to move up the landscape to the global optimum. If there is a single global optimum, the surface is said to be *well-behaved*; if there are multiple optima or a large number of trees with equal or approximately equal likelihoods (a "ridge"), then the surface is described as *poorly-behaved*.

1b. Inference using RAxML

RAxML represents a rapid, approximate approach for searching for ML trees. It has been developed and maintained by Alexis Stamatakis and represents one of the first approaches to explicitly combine optimality criteria in a single analysis.

As we discussed in lecture, RAxML generates a collection of starting trees via parsimony stepwise addition & many random addition sequences. It swaps on each of these using lazy random SPR rearrangements by regrafting only to branches close to prune point. Branch length optimization is limited to just the three branches closest to regraft point.

If you want, you can download a GUI version from the following site:
https://antonellilab.github.io/raxmlGUI/

For the lab, we'll use the command-line version on the cluster.

We will use the same dataset as last lab, `biol545ParDat.nex`. Before we can infer our ML tree, we first need to infer a model of sequence evolution. We can do this in PAUP*.

Log into the cluster, load the PAUP module, and execute your dataset. Before using the AutoModel function in PAUP, we need to generate a starting tree. We can do this by generating a neighbor joining tree using LogDet distances.

In PAUP:

`exe biol545ParDat.nex;` (Optional: include the -L [output prefix].log option after your exe command to log your PAUP console output to a file for later reference)

```
dset distance=logdet;
nj;
```

Now that we have a starting tree stored in our buffer, we can use the AutoModel function in PAUP. By default, PAUP will evaluate models of sequence evolution under both AICc (corrected Akaike information criterion) and BIC (Bayesian information criterion) and will output the best fit model under both criteria (For a complete list of options you can type `AutoModel ?`).

```
AutoModel;
```

In your automodel output, you can see all 55 models of sequence evolution that were tested listed in descending order based on their likelihood scores. At the top of the lists (and listed at the bottom of the output), you can see which model is selected by each criterion.

Answer question 1 in the assignment.

We are now ready to estimate a ML tree using raxml. First, load the raxml module on the cluster by typing `module load raxml.` We will perform ML estimation with 100 bootstrap replicates. RAxML's options for model of sequence evolution are quite limited, so we will use GTR+I+G.

```
module load raxml

raxmlHPC –s biol545ParDat.fasta –m GTRGAMMAI –p 12345 –n [name of run]
```

RAxML will output a number of few trees, but we are interested in our maximum likelihood tree. This can be found in the "RAxML_bestTree.[name of run] file in NEWICK format.

1c. ML estimation using IQ-TREE

As we discussed in lecture, IQ-TREE is another implementation of approximate ML tree estimation that combines parsimony and likelihood. Sit is seeing increasing usage, largely because it has been packaged with useful model selection and hypothesis testing approaches. In searching tree space, IQ-TREE begins with a set of 100 candidate trees that are generated with a variety of options; this may include a set of (100) starting trees via parsimony stepwise addition with random addition sequences. Trees are ranked under ML and the 20 best are selected; each in then subjected to local NNI swapping, again with only neighboring branch lengths reoptimized. The best 5 trees are retained, and one is selected at random for a random NNI. If the new tree is better than any of the 5 trees in memory, the new tree replaces it. Random perturbations (NNIs) are repeated until 100 successive iterations fail to find a better tree.

When using IQ-TREE, you are able to specify your own model of sequence evolution, or you can use the integrated model selection tool (Model Finder Plus). IQTREE has very similar usage to RAxML, although this time we will generating a consensus tree based on 1000 bootsrap replicates.

```
module load iq-tree

iqtree2 -s biol545ParDat.fasta -m MFP --prefix [output prefix] -B 1000
```

Since we bootstrapped our ML tree in this analysis, the final consensus tree can be found in [output prefix].contree.

Answer question 2 in the assignment.

Make a new tree file that contains both the RAxML best tree, and the IQ-TREE consensus tree, and load these trees into PAUP. Compute Robinson-Foulds distances between your two trees.

```
cat [RAxML_bestTree file] [iqtree.contree] >> new_file.tre

paup

gettrees file=new_file.tre;

treedist;
```

Answer question 3 in the assignment.

Section 2: Bayesian Inference

2a. Background

Bayesian inference has become widely used in phylogenetics. Bayesian methods differ from ML methods philosophically; in Bayesian inference, we include our beliefs, or prior information about how our inference, as part of our analysis. Specifically, we use Bayes' Theorem:

$$p(\theta \mid D) = \frac{p(D \mid \theta)p(\theta)}{p(D)}$$

where

$$p(D) = \int_{\Theta} p(D \mid \theta)p(\theta)d\theta$$

Here, the conditional probability of our model given the data (the posterior probability) is equal to the probability of the data given the model (the likelihood), weighted by the prior probability of the model, and all this is divided by the probability of the data. Defining "probability of the data" is conceptually difficult. Think of it as a distribution composed of the probabilities of the data given the model weighted by the probability of the tree; to get the probability of data, you simply integrate over this distribution. However, it's not that easy. We don't know anything about the shape of this distribution *a priori*, so we take samples from it using Markov chain Monte Carlo (MCMC) techniques.

We'll talk more about MCMC methodology and the use of priors in lecture, but there are some terms you should be familiar with. We will take a *sample* (point estimate of parameter values) every N *generations* (one iteration or step in the chain) once the Markov chain has reached *stationarity* (began sampling from the correct underlying distribution). **Two runs are said to have *converged* if they are sampling from the same underlying distribution** (assessed by looking at the mean and variances of parameter estimates, marginal probability distributions, deviations of split frequencies, or MCMC "traces") You will learn more about these properties later in the course.

Bayesian analyses require prior probabilities to be placed on all parameters of interest. In molecular phylogenetics, we place priors on the tree, the nucleotide frequencies, the categories in the rate matrix, the branch lengths, the molecular clock processes, the relative and absolute dates of nodes, etc. – *every* parameter which impacts our analysis.

Luckily, there are some pretty good default priors available as a result of analyzing thousands of datasets. Priors can also be *uninformative*, assigning equal probability to all parameter values across a range, or *unbounded*, using distributions with one or more tails that decrease asymptotically to infinity.

2b. Bayesian Evolutionary Analysis by Sampling Trees (BEAST)

One of the most popular and well-developed Bayesian phylogenetics programs is BEAST. BEAST can be run from the command line (for long analyses or simulation studies), but we'll do a run using the GUI. BEAST requires an XML file for input, and the team has developed an intuitive way to generate one: BEAUti.

Start BEAUti, and click on the "+" icon in the bottom-left corner. Select the "Biol545ParsDat.nex" file. We are not going to do a partitioned analysis, so make sure that the data type, site model, clock model, and partition trees are all equivalent. You would change some of these if you wanted to partition your dataset. The 'Link' and 'Unlink' tabs along the top of the window can help you do this.

Next, along the very top of the window, select the tab labeled "Site Model". We want to use HKY model. Check the boxes to estimate number of Gamma categories, and proportion of invariant sites. Set base frequencies to "Empirical"

Under the "Clock Model" tab, select "Relaxed Clock Log Normal" from the menu, and check the "Normalize" box.

Under the "Priors" tab, change the prior to a "Yule Model".

Take a look at the "Priors" tab. These settings, and their associated 'operators', are the means through which BEAST will propose changes during its MCMC. Some of what you are seeing is due to selections you made when you selected your model of sequence evolution; the operators are at default values and, in general, are pretty good.

Under the "MCMC" tab, change the chain length to 5000000 (for runtime). Go to file and save the parameters file somewhere you know, preferably a new folder, with a .xml extension.

Now, do the exact same thing, except this time check the box that says "Sample from prior only". Save this XML file under a different name.

Now, open up the BEAST executable. Select one of the two XML files that you generated, uncheck the Beagle library box, then execute the run. Do the same for the other file one once you're done.

We will analyze these runs next week.

Assignment 3

Biology 545: Principles of Systematics

Name: _____

1. What model(s) of sequence evolution was/were selected by AICc and BIC? Are they the same?

2. What model of sequence evolution was selected by Model Finder Plus?

3. How do your ML tree estimates compare between RAxML and IQ-TREE?

4. Let's say you performed a model selection analysis and K80+I+Γ was selected as the best fitting model. How would you implement this in BEAST?