Integrative Biology 200, Spring 2014 *University of California, Berkeley*
Principles of Phylogenetics: Systematics *Updated by Traci L. Grzymala*

Lab 05: *PAUP** Continued
UPGMA, NJ, Parsimony, and Maximum Likelihood

Last week you were introduced to the program PAUP* and today we will continue to use this program and explore different optimality criteria for reconstructing phylogenies. We will compare distance methods (UPGMA and Neighbor-Joining), parsimony, and maximum likelihood trees.

Open PAUP*.

Create a new folder (eg. IB200_Lab5) and then set this as your working directory for the day.

```
paup> cd /Users/yourusername/Desktop/IB200_Lab5
```

Load your data into PAUP. Either use some of your own data (in Nexus format) or use the sample file cephalopod_COI_Clustalw.nex. Remember PAUP needs the path of where this file is, so dragging and dropping the file generally works.

```
paup> execute filename
```

Exercise 1: Distance Methods

Distance Methods

We discussed distance methods in class and you have learned that they are not the most theoretically justified of methods for inferring phylogenies, although clustering methods do have some uses in other areas of statistics. They are by far the fastest way to find a tree. Whereas parsimony and likelihood methods have to search through tree space and compare the optimization of the character matrix on many trees, most distance methods use an algorithm to directly generate a tree from the distance matrix. This speed makes it very useful for genomics, where it is often necessary to generate tens of thousands of trees, but getting the exact tree each time is not as important as getting the right tree the vast majority of the time.

Change the optimality criteria to distance.

```
paup> set criterion = distance;
```

UPGMA

UPGMA is a clustering algorithm for generating trees from a distance matrix. It assumes that the trees are ultrametric, meaning that the branch lengths obey the molecular clock. It approximates the least squares tree and is well behaved if the molecular clock is followed. Generate a UPGMA tree with branch lengths and save it:

```
paup> upgma brlens = yes treefile=upgmatree.tre;
```

Neighbor Joining (NJ)

Neighbor joining is another clustering algorithm, but it does not assume the molecular clock. NJ is currently the distance method with the best reputation and is thus the one most commonly used, although UPGMA is still used in a lot of genomics studies. The clustering algorithm of NJ is similar to that of UPGMA in that both replace pairs of OTUs with composite OTUs one after another. However, it makes much more complicated calculations than UPGMA that we won't go over here.

Generate a UPGMA tree with branch lengths and save it:

```
paup> nj brlens = yes treefile=njtree.tre;
```

***Question #1:* What is the total branch length sum for your UPGMA tree? For your neighbor-joining tree? [HINT: Scroll back up through your PAUP window to find this information]**

***Question #2:* Examine these two trees in FigTree. What is the main difference between these two trees in terms of branch lengths?**

Exercise 2: Parsimony

Parsimony

Parsimony is the optimality criterion that minimizes the number of changes on the tree. Unlike distance methods, it is a phylogenetic method that distinguishes between symplesiomorphy and synapomorphy. Change the optimality criteria to Parsimony.

```
paup> set criterion=parsimony
```

Then, run a heuristic search:

```
paup> hs
```

Then save your trees:

```
paup> savetrees file=parstree.tre
```

Open this tree file in FigTree along with your UPGMA and NJ trees to examine them.

***Question #3:* Which of your distance trees is most similar to your parsimony tree? Do you notice any differences?**

***Question #4:* Figure out a way to highlight a clade that is different between the parsimony tree and the neighbor-joining tree in FigTree. Take a screen shot and send it to me.**

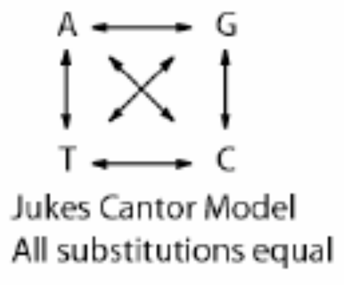
Exercise 3: Maximum Likelihood (ML)

Maximum Likelihood

Maximum likelihood (ML) is a statistical method for reconstructing trees. We'll discuss ML in lecture later this week. Basically, ML operates by calculating the following conditional equation: What is the likelihood of observing a data set given a phylogeny and a model of DNA sequence evolution? The tree with the highest likelihood score is considered the best tree. When using maximum likelihood to build trees, **we have to first select a model of DNA sequence evolution**.

Normally you will do this using jModelTest. But for today I'll describe some of the most commonly used models and then run these in PAUP to examine if the model affects our analyses. We will go into much more detail regarding Maximum Likelihood and model selection later this week, but I don't want to overwhelm you today.

Models of Nucleotide Change



JC69: (Jukes and Cantor 1969) – all nucleotide substitutions are equal and all base frequencies are equal.

F81: (Felsenstein 1981) – all nucleotide substitutions are equal, base frequencies are allowed to vary.

K2P: (Kimura two-parameter model; Kimura 1980) – two nucleotide substitutions types are allowed, those between transitions and transversions. Base frequencies are assumed equal.

HKY85: (Hasegawa-Kishino-Yano 1985) – two nucleotide substitution types are allowed, those between transitions and transversions. Base frequencies are allowed to vary.

GTR: (General time reversible; Lanave et al 1984; Tavare 1986) – Assumes a symmetric substitution matrix, in other words A changes into T with the same rate that T changes into A and each pair has a different rate. Base frequencies are allowed to vary.

Set the optimality criteria to likelihood

```
paup> set criterion=likelihood
```

Specify the Jukes-Cantor model (**JC69**):

```
paup> lset nst=1 basefreq=equal;
```

ML analyses are notorious for their slow computational speed. Make sure you run a heuristic search!

```
paup> hs
```

Now let's look at our tree(s):

```
paup> showtrees
```

PAUP will show you the tree that it found in an ascii display. Then you can use savetrees to write your tree to file and include branch lengths.

```
paup> savetrees brlens file=jcmltree.tre
```

A new tree file called "jcmltree.tre" should appear in your designated folder.

Repeat the maximum analysis using different models of nucleotide substitution introduced today. For each of these models run a heuristic search and save the tree(s) and branch lengths to a file. Specify the **F81** model:

```
paup> lset nst=1 basefreq=empirical;
```

Specify the **K2P** model:

```
paup> lset nst=2 basefreq=equal;
```

Specify the **HKY** model:

```
paup> lset nst=2 basefreq=empirical variant=hky;
```

Specify the **GTR** model:

```
paup> lset nst=6 basefreq=empirical;
```

You should now have five maximum likelihood tree files – one for each of the substitution models that you examined.

Question #5: **Examine these trees in FigTree. Do you notice any differences? It might be easier to compare the simplest model (JC69) to the most complex (GTR) to look for differences. If you find an ambiguous relationship in your trees between the two model selections, highlight that clade and take a screen shot.**

You have probably noticed that there can be significant differences between your trees based on the nucleotide substitution model. But how do you choose between these models? In general you want to use the simplest model that adequately explains your data. But, if a more complex model yields a greater improvement in tree score than would be expected if applied to random data, then use the more complex model. We'll discuss how to systematically do this in Friday's lab.

Question #6: **What are the different tree scores for the maximum likelihood models evaluated today? Look back through your PAUP window to find these. Based on these scores alone, which model should you choose?**