Lab 04:
Distance and parsimony inference using PAUP*;
UPGMA, neighbor-joining, bootstrap, and jackknife
Updated by Carrie Tribble

1 Before you begin

Please download:

1. PAUP*:
   `http://people.sc.fsu.edu/~dswofford/paup_test/`
2. Primate mitochondrial DNA:
   `http://ib.berkeley.edu/courses/ib200/labs/04/primate-mtDNA.nex`

2 Introduction

Today we will cover both distance and parsimony inference using the software PAUP* and introduce two different methods of measuring uncertainty over the phylogeny: the bootstrap and jackknife. PAUP* (Phylogenetic Analysis Using Parsimony [*and other methods]) is supposedly pronounced pop star, and it can infer phylogenies using distance, parsimony, and likelihood. It is widely used (with over 40 *thousand* citations), however its author, David Swofford, has recently (2015) released the Bayesian phylogenetics software Phycas with Paul Lewis and Mark Holder.

3 Basic PAUP*

Open PAUP*. PAUP* takes a nexus (.nex) file as input. If you have your own sequence data you can use that, otherwise use the `primate-mtDNA.nex` file. First set the working directory (like in R) by typing `cd <directory>` where `<directory>` is the full path to the directory in which you saved the `primate-mtDNA.nex` file. Now type `execute primate-mtDNA.nex`.

PAUP* will tell you a few things about the file (how many taxa and characters it has in it, what nucleotide ambiguity codons are being used). For instance, right now R is treated as either A or G for analytical purposes. You can type `?` after a command to see a brief description of it. Check out some other useful commands:

```
cstatus
showdist
showmatrix
tstatus
```

Question 1:
Use the commands above to explore the dataset. What is the optimality criterion currently set to? Are these characters ordered or unordered? What is the distance between chimpanzees and humans?

4 Distance Methods

Distance-based methods are mostly hierarchical clustering algorithms. In fact, many major early contributions to clustering algorithms were developed by phylogeneticists/pheneticists, for example single-linkage hierarchical clustering was first introduced in Sneath [1957].

From lecture you have learned that distance-based methods are not the most theoretically justified of methods for inferring phylogenies, although clustering methods do have many other uses in statistics and computer science. 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;
```

4.1 UPGMA

UPGMA (Unweighted Pair Group Method with Arithmetic Mean) 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.If you want to learn more about the UPGMA algorithm, check out the associated wikipedia page here. Generate a UPGMA tree with branch lengths and save it:

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

4.2 Neighbor Joining

Neighbor joining (NJ) is another clustering algorithm, but it does not assume the molecular clock. NJ has the mathematical property that if the distance matrix is correct, then the output tree will be correct. 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, compared to UPGMA it is more complex. NJ uses the distance matrix to calculate a Q matrix that we won't go over here. Read more about the neighbor joining algorithm here if you feel so inclined. Generate a NJ tree with branch lengths and save it:

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

Question 2:
What is the total branch length sum for your UPGMA tree? For your NJ tree? [HINT: Scroll back up through your PAUP window to find this information]

5 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
```

Because searching through all of tree space can be computationally untenable, we will use a heuristic search process to search through some trees rather than all possible trees. It is possible to use a variety of heuristic searches through PAUP*. To see some of these options, check out `Analysis > Heuristic Search`. For now, let's use the default settings:

```
paup> hs
```

Then save your trees:

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

Question 3:
Examine your parsimony, UPGMA, and NJ trees in FigTree. Do the tarsiers fall out in the same place in all your trees?

6 Estimate Support by Bootstrapping

Now let's figure out how well supported these groupings are. One measure of support is called the **bootstrap**. It works by choosing columns randomly from the character matrix until it is has chosen the same number of columns as were in the original matrix. Because it returns to the original matrix each time it chooses a new column, some characters may be represented several times in the bootstrap matrix, while others are omitted. This is known as resampling the data with replacement. In practice, although it is possible to randomize taxa, bootstrapping almost always randomizes characters. Bootstrapping calculates a support value for each node based on the fraction of samples that support that node. Bootstrap provides a number on each node $(0 - 100)$. The highest support value is 100, while values below 70 are usually considered weak. Values below 50 aren't shown. In fact, branches below 50 are collapsed and shown as a polytomy. Bootstrap support is somewhat sensitive to the number of replicates used, but not terribly so.

Let's run a bootstrap analysis with a hundred replicates, using a heuristic search for each replicate, and randomizing the order of taxa while building the trees on your data. In practice, you'll likely want to run many more than 100 replicates, but for expediency in this lab we will stick with 100. When it's done, PAUP will display a 50% majority rule consensus tree from all 100 bootstrap runs. This type of consensus tree only shows clades that are present in more than 50% of the primary trees.

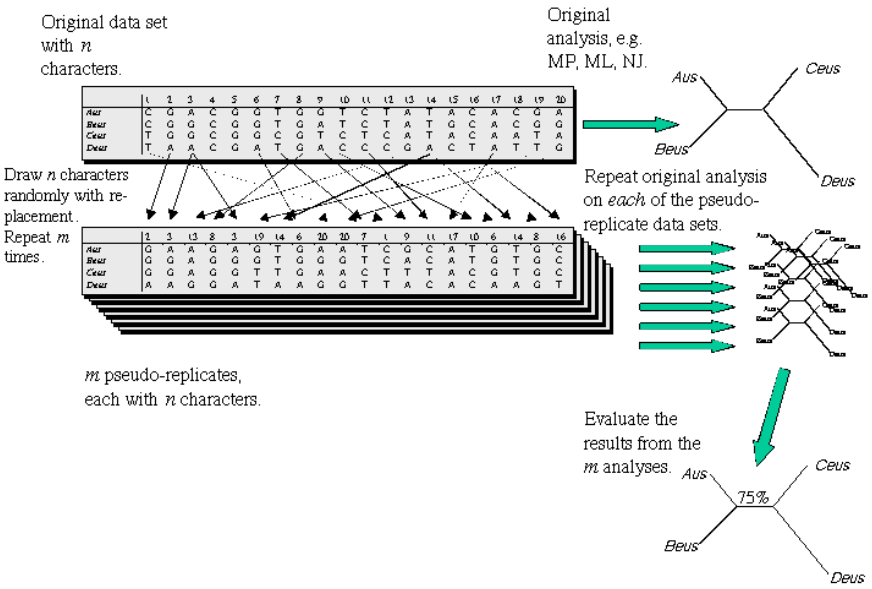

Figure 1: Bootstrap replicates. Image from
http://artedi.ebc.uu.se/course/X3-2004/Phylogeny/Phylogeny-Credibility/Phylogeny-Credibility.html

```
paup> bootstrap nreps=100 search=heuristic /addseq=random;
```

PAUP will also display a list showing the frequency of different taxon bipartitions (that is, how often out of the 100 replicates the taxa were grouped together.) If you look at the tree, youll see that Pan has a (3) after it and Gorilla has a (4). Look down the chart until you see a line where there is a * under 3 and 4, but not under any of the other numbers. The frequency on this line will be something like 47.70, which means these two taxa were grouped together in about 47.7 out of the 100 runs. Some of these numbers are fractional because some runs produced more than one most parsimonious tree.

Now export the fancy new bootstrap values you just came up with:

```
paup> savetrees from=1 to=1 savebootp=nodelabels file=bootstrap.tre;
```

Note that you must supply the tree numbers that you want to save (here, from 1 to 1, since there is only one bootstrap tree) and the file name you want to save to. You can view these node labels on the tree in FigTree or Mesquite.

Question 4
Figure out a way to show your tree with bootstrap support in FigTree. Note: The program can be a bit glitchy for some reason so you may need to open the program first and then go to `File - Open` to select your file. Highlight the weakly supported *Pan* clade and take a screenshot.

7 Estimate Support by Jackknifing

Jackknifing is very similar to bootstrapping, but rather than resample the data, it uses subsets of the data. This is also described as resampling without replacement to create

a smaller dataset. The purpose of this is to see if excluding certain characters has a big effect on the shape of the tree. (You can imagine that some 'outlier' character might have a disproportionate influence on the relationships that are reconstructed; jackknifing is an attempt to get around this.) Jackknifing is much less common in the literature than bootstrap support.

```
paup> jackknife nreps=100 search=heuristic /addseq=random;
```

Export the jackknife values the same way you exported the bootstrap ones:

```
paup> savetrees from=1 to=1 savebootp=nodelabels file=jackknife.tre;
```

Question 5
Open your jackknife tree file in FigTree and show the jackknife values on the tree. Then highlight the weakly supported clade *Pan* and take a screen shot.

Please email me the following:

1. The answers to questions 1-5.
2. Screengrabs of your bootstrap and jackknife trees with support values and the *Pan* clades highlighted.

8 Bonus

Using your skills from last lab, find a gene from a taxon of interest on NCBI (or elsewhere). BLAST that gene and find around 10 other sequences of that gene in different taxa. Download each of those sequences as a separate fasta file and save the files in a directory on your computer.

Open Terminal and change your working directory to the directory where your fasta files are saved. Now create a new empty file named `sequences.fasta` and save all the information from your 10 fasta files in the new file. Still in Terminal, align the sequences in `sequences.fasta` using either Mafft or Muscle and save the alignment as `aligned.fasta`.

Open PAUP* and generate a parsimony-based tree. Save the tree and open the treefile in FigTree. Send me 1. your code for concatenating and aligning the sequences, 2. your treefile, and 3. a screenshot of your tree.

References

Peter HA Sneath. The application of computers to taxonomy. *Journal of general microbiology*, 17(1):201–226, 1957.