

Lab 8 - Advanced PAUP*

Today we're going to be doing some more advanced things in *PAUP**. Command line *PAUP** is very powerful, but of course has the disadvantage that you have to remember all the commands. Of course, no one, not even David Swofford, (who wrote *PAUP*) actually knows all the commands by heart. Normal humans tend to refer quite often to the documentation, which is lucky and searchable. We're just going to use the example nexus files, but if you want to use your own data go ahead. If you are bored of *PAUP* and want to work on your own data, I can help you with that too. Just raise your hand and call me over.

*Basic Command Line PAUP**

Copy the **angiosperm** and the **primate-mtdna** files from the **example nexus files** folder in the **PAUP** folder to a folder that you create on your desktop. The *PAUP* program can be accessed through the start menu. The *PAUP* folder is in **C:/Program Files/IB200A/PAUP**.

Open *PAUP** in the **IB200** folder in the Start Menu.

It will ask you to open a file, but **cancel** that step for now.

As you can see there are a few pull down menus, but they don't do a lot. The one thing that these menus do OK is deal with files. You can use these menus to open and save files, although you have to use the command line to save trees. The edit and window menus have the basic features that you would expect.

Open a file **File > Open > PAUP4b10**. Select the **angiosperm.nex** file and hit **Open/Execute**.

Now, go back to **C:/Program Files/IB200A/PAUP** and open up the **Cmd_ref_v2**. This is a PDF file in the **PAUP** folder. Open up the **Bookmarks**. You can use the Bookmarks to find commands. Open up the **Command Reference** folder in the **Bookmarks** window. You can also search for commands using the search tool.

You will see a bunch of different folders of commands for different blocks. All nexus files are made up of a combination of these different blocks. Each block starts **begin (type of block)** and finishes **end;**. Between are a number of commands that tell the program what to do, and the commands that are allowed in each block are described in the appropriate part of the document.

The **PAUP block** can be added to a nexus file. The commands allowed in the **PAUP block** can also be typed into the command line of the running program. It is nice to put them in a **PAUP block** because then you don't have to repeatedly type the same

commands in, on the other hand the command line allows you the flexibility to put commands in as the program runs.

Let's run a simple heuristic search. Open up the **Commands used in the PAUP block** folder in the **Bookmarks** window. You will see a list of commands. Click on **Hsearch**, so that you go to that part of the document.

This is a description of the **Hsearch** command. Under syntax is the description of how you type the command. In this case it says **HSEARCH [options] ; .** This means that you type in **hsearch** or **hs** and then type in any of the options that you want to change. The options and their settings are listed under **Available options** and described under **Description of options**.

For the initial run let's just stick with the default settings to get an idea what the most parsimonious tree is. Type **hsearch** and hit **Enter**.

Did you get multiple most parsimonious trees? Now we should construct a consensus tree. Go to the command **contree** in the manual. The command syntax is now a little different: **CONTREE [tree-list] [/options] ; .** This means that after typing **contree** you list the trees you want to do a consensus of, then you put a "/" before you type in the options. We want to find a majority rule consensus tree of all the trees. The default for the tree-list is **All**, so if we don't list any trees it will compute a consensus of all of them. On the other hand the default is a strict consensus tree, so we're going to use the **Majrule** command to get a majority rule tree.

Type: **contree / majrule=yes** and hit **enter**. You will get a strict consensus and a majority rule tree.

Decay Index

To find Bremer support values we need to retain more trees than just the most parsimonious. When we ran the original search we found that the shortest tree had 284 steps. To find which nodes have a decay index of 1 we need to find all the trees with 285 steps or less. To do this we will use the **keep** option.

Type: **hsearch keep=285** and hit **enter**. You will have to increase the maximum numbers of trees saved. Make it 1000.

Now construct another strict consensus tree. This tree is much less resolved than the one we got before. All the clades that are now unresolved do not appear in one of the trees of length 285, so they have a decay index of 1.

To find higher Bremer support values increase the number of steps in the minimum trees. Rather than doing a new search each time, you can just search for trees with say 5 extra steps. Then to figure out which node have a decay index of say 3 only construct consensus trees of the trees with less than 4 extra steps.

Batch commands

I said before that often it is best to put a lot of commands in the nexus file. It is always good to define sets of taxa and characters in the nexus file. This allows you to easily turn on and off different sets of characters and taxa to focus your analysis.

Go to the **Open** window, select edit, and edit the **primate-mtDNA** file.

Pan down and look at the **Assumptions** block. The first two commands divide the sequence into coding and non-coding regions by defining those two **charsets**. The next three lines divide the coding regions into first second and third codon positions. The **exset** command defines sets of taxa that can be excluded together from the analysis. The two **usertypes** are user defined cost matrices of changes for a parsimony analysis. The last command defines a set of taxa.

Now erase all the commands in the **PAUP** block. We're going to use that space to enter our own commands. Make sure that you put a semicolon after each command line.

First type: **exclude noncoding 3rdpos;** This will limit our characters to just the first two very stable codon positions of the sequence by eliminating these two character sets.

Type: **delete hominoids;** to limit our analysis to the non-ape primates.

Type: **outgroup 12;** This sets the twelfth taxon in our analysis as the outgroup, which in this case is the tarsier. Is that the right one?

Type: **set criterion=likelihood;** This is self explanatory. The important thing to note here is the **set** command, which controls many options.

Type: **lset basefreq=equal;** Normally you would get this model from *Modeltest*, but I just chose an arbitrary one here to speed up the calculation.

So your entire PAUP block should look like this:

```
begin paup;  
exclude noncoding 3rdpos;  
delete hominoids;  
outgroup 12;  
set criterion=likelihood;  
lset basefreq=equal;  
end;
```

Pull down the **File** menu and select to execute your nexus file.

It should give a list showing all the changes that you made. Type **cstatus** to confirm that many of the characters are excluded, and that your criterion is likelihood. Type **tstatus full=yes** to see which taxa have been deleted and which are in the outgroup.

Now type **hsearch** to run the search.

When the search is done type **showtrees** to display the trees.

Constraint trees

We are now going to run this data file in *PAUP** using a constraint tree. There are many different situations in which using a constraint tree might be useful. For example if you want to see how much a certain arrangement affects your optimality score or if you want to include a tree from another source.

First let's run an analysis on all the taxa in this data set.

Type **undelete all**. This will put your deleted taxa back into the analysis. It should tell you that all your taxa are nondeleted now.

Type **hsearch** and when the search is done type **showtrees** to see the best tree. Note the topology and the likelihood score of the best tree.

Now we are going to create a constraint tree using *Mesquite*. The same thing could be done in *Maclade* if we were using a Mac.

Open *Mesquite* and then open your data file.

From the **Character Matrix** window, pull down the **Taxa&Trees** menu and select **New Tree Window**. Select **Default trees**.

Go to the second tree which is totally unresolved. Put some taxa together which do not go together in your maximum likelihood tree, while leaving the rest of the taxa unresolved. For example humans and chimps don't go together in my maximum likelihood tree, so I could put them together to see how that affects my hypothesis. Or I could do something totally arbitrary and wrong like putting an unresolved clade of macaques as sister to the gibbon.

Pull down the **Taxa&Trees** menu and select **Save Copy of Tree Block**. Give the tree a name you can remember and save it to the same folder where your data file is, as that is currently *PAUP**'s default directory. Shut down *Mesquite*.

In *PAUP** type **loadconstr file='(the name of your constraint tree file)'**. This will load that tree as your constraint tree.

Type **showconstr** to look at the topology and make sure that this is the right tree.

Type **hsearch enforce=yes**. This will do a heuristic search while enforcing the constraint tree that you loaded.

Type **showtrees**. Does the new tree have the topology that you enforced? Did the constraints affect the rest of the topology? How does the likelihood score compare?