Lab 1: Introduction to PHYLIP

What's due at the end of lab, or next Tuesday in class:

1. **Print out of Caminicules outfile and outtrees from pars**
2. **Print out of Azolla consensus output**

Introduction

Today we will be learning about some of the features of the PHYLIP (PHYLogeny Inference Package) software package. PHYLIP was developed by the famous evolutionary biologist Joe Felsenstein, works on most operating systems, and is available for free online. It is widely used, but slightly less popular than PAUP*. Like Mesquite, PHYLIP is an open source package, and you can make changes yourself if you program in C++.

Methods available in the package include parsimony, distance matrix, and likelihood, as well as bootstrapping and consensus trees. Data types that can be handled include molecular sequences, gene frequencies, restriction sites, distance matrices, and discrete-state characters.

What is PHYLIP?

PHYLIP consists of about 30 programs that perform different algorithms on various types of data, and collectively are able to do most things you might want to do when it comes to inferring phylogenies. In PHYLIP, there is a program for each method you might want to use, and each type of data you might want to work with:

Molecular sequence methods

Program:	What it does:
protpars	protein parsimony
dnapars	DNA sequence parsimony
dnapenny	DNA parsimony branch and bound
dnamove	interactive DNA parsimony
dnacomp	DNA compatibility
dnaml	DNA maximum likelihood
dnamlk	DNA maximum likelihood with clock
proml	Protein sequence maximum likelihood
promlk	Protein sequence maximum likelihood with clock
dnainvar	DNA invariants
dnadist	DNA distance
protdist	Protein sequence distance
restdist	Restriction sites and fragments distances
restml	Restriction sites maximum likelihood
seqboot	Bootstrapping/Jackknifing

Distance matrix methods

Program:	What it does:
fitch	Fitch-Margoliash distance matrix method
kitsch	Fitch-Margoliash distance matrix with clock
neighbor	Neighbor-Joining and UPGMA method

Gene frequencies and continuous characters

Program:	What it does:
contml	Maximum likelihood continuous characters and gene frequencies
contrast	Contrast methods
gendist	Genetic distance

Discrete characters methods

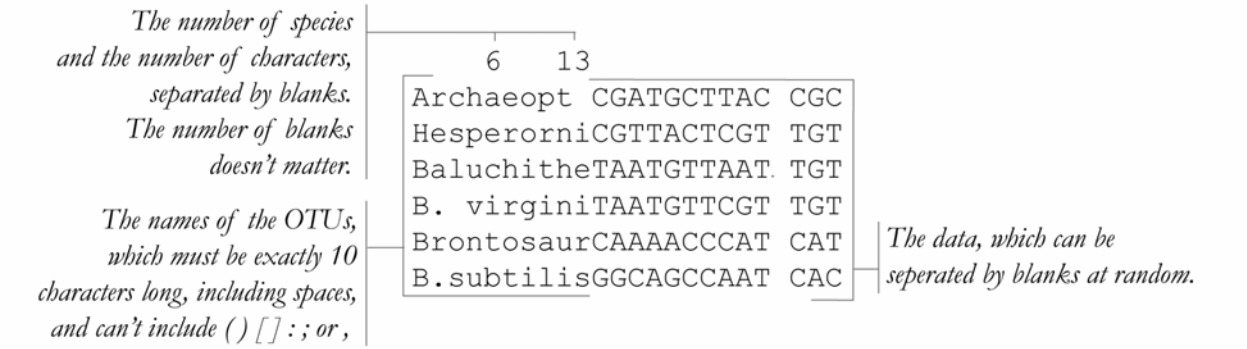
Program:	What it does:
pars	Unordered multistate parsimony
mix	Mixed method parsimony
penny	Branch and bound mixed method parsimony
move	Interactive mixed method parsimony
dollop	Dollo and polymorphism parsimony
dolpenny	Dollo and polymorphism branch and bound parsimony
dolmove	Dollo and polymorphism interactive parsimony
clique	0/1 characters compatibility method
factor	Character recoding program

Tree drawing, consensus, tree editing, tree distances

Program:	What it does:
drawgram	Rooted tree drawing program
drawtree	Unrooted tree drawing program
consense	Consensus tree program
treedist	Tree distance program
retree	interactive tree rearrangement program

Getting ready to use PHYLIP

Input files: PHYLIP uses a standard format for its input. Here's what it looks like, and what the parts mean:



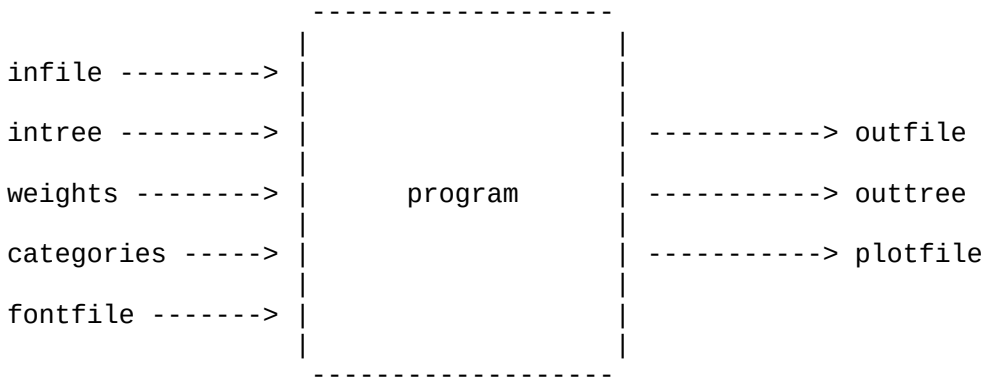
If you forget to extend the names to ten characters in length, using spaces or other characters, the program will not read the file in correctly and an error message will result.

This is a rather persnickety format, but luckily, you don't have to type it by hand. Mesquite can export this format for you. Go to File > Export… > Phylip (categorical data). A dialog will pop up. Set "Minimum length of taxon names" to 9 and "End of line character" to "Current System Default." Then give your file a name that is both easy to remember and easy to type. Save it to the **exe** folder inside the Phylip folder. Step-by-step instructions for making a Phylip input file from scratch can be found at the end of this lab handout.

[Step 1 – Make a PHYLIP input file] Last week, you and your lab partner made a matrix using Mesquite and the Caminalcules model data. You emailed it to me, so hopefully you have a record of it and you can download it now. Save it to the desktop and open it in Mesquite. **File > Export… > Phylip (categorical data)**. A dialog will pop up. Set "Minimum length of taxon names" to 9 and "End of line character" to "Current System Default." Then give your file a name that is both easy to remember and easy to type. Save it to the **exe** folder inside the Phylip folder.

Using PHYLIP

PHYLIP is not hard to use, but it is not intuitive either. You'll need your "I'm comfortable with using command-line driven programs" hat. You'll find PHYLIP folder under **Start menu > Programs > phylip**. (If you download PHYLIP yourself, you can place it in any folder—it is self-contained.) There are three files inside, **doc**, **exe**, and **src**. These contain documentation files, executables (ie, actual programs) and source code. You'll want to keep all the files you're working with in the **exe** folder, because to the best of my knowledge, PHYLIP can't switch working directories. Each program in PHYLIP takes input in the form of a file, operates on the data, and produces output in the form of two new files. The figure below shows the default name for each of the files:



This means if you include a file called `infile` in the **exe** folder, the program you are using will automatically read it in. If there is no file called infile, it will prompt you for the name of the file you want to use. Same goes for the output; all of the PHYLIP programs store their results in a file called `outfile` and a corresponding tree file called `outtree`. Phylip will warn you before overwriting them and give you a chance to name it something else. The easiest way to produce and read these files is probably with a text editor.

Today we're going to use a few different programs, so that you can see what PHYLIP can do. The first one is **pars**, which allows you to use unordered, multistate parsimony to infer phylogenies using discrete characters… the kind of characters that you made in lab last week. The second one is **dnapars**, which does the same thing, but for DNA data. We'll also learn how to output tree drawings (**drawtree** and **drawgram**) and make consensues trees (**consense**.)

pars

[Step 2 – Analyze your Caminalcules file in pars] Ok, let's experiment with pars. Open the **pars** program, **start menu > phylip > exe > pars.exe**. A window will pop up, and you'll see the following:

```
pars.exe: can't find input file "infile"
Please enter a new file name>
```

Enter the name you gave your file. You'll see the following:

```
Discrete character parsimony algorithm, version 3.6

Setting for this run:
  U                 Search for best tree?  Yes
  S                       Search option?  More thorough search
  V              Number of trees to save?  100
  J     Randomize input order of species?  No. Use input order
  O                       Outgroup root?  No, use as outgroup species 1
  T              Use Threshold parsimony?  No, use ordinary parsimony
  W                      Sites weighted?  No
  M           Analyze multiple data sets?  No
  I            Input species interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                       Print out tree  Yes
  4         Print out steps in each site  No
  5  Print character at all nodes of tree  No
  6       Write out trees onto tree file?  Yes

  Y to accept these or type the letter for one to change
```

This is the PHYLIP menu, which allows you to se various options relating to the things the program can do. I'll cover a few of the options, but you can check out the rest in the appendix at the end of this lab handout, and also the online documentation at http://evolution.genetics.washington.edu/phylip/phylip.html. (Be patient when reading the online documentation; there are a lot of places where you will be referred to other documents for the full story.)

Today, we're just going to change the Jumble option. First, we will randomize the input order of the species. This will allow pars to search a greater portion of tree of tree-space, giving us more confidence that we have found the most parsimonious result. Type j and hit enter. Enter any odd number to act as a random number seed. Enter 10 as the number of times to jumble (this will tell the program to randomly add the taxa together to form a starting tree 10 times; each of the 10 starting trees will then be used in its searches for a most parsimonious cladogram).

Now hit Y. When the analysis is finished, the program window will close, and two new files, 'outfile' and 'outtree' will be added to your PHYLIP folder. You can open these files in Word or Text Edit to check the results. The outfile shows the cladogram(s) you have inferred, and gives some details of the results. The outtree gives you a simple, parenthetical representation of your cladogram(s). Rename these files to something that is easy to remember and easy to type. **[Step 3 – Save these files so you can print them out to turn in later]**

drawtree and drawgram

Now, you just created a tree from you Caminalcules data and I bet you want to look at it. PHYLIP offers you two ways to do this, drawgram, which draws rooted cladograms, and drawtree, which draws unrooted trees. You can try either one. When you double click on drawtree or draw gram, the terminal window will open and it will ask you for an input file. Type the name you used to save your file (outtree if you didn't change it.) The program will read in your tree, then it will ask you what font to use to draw it. PHYLIP has 6 font files:

File name	What it looked like on Steph's computer in the preview
font1	a sans-serif machine font
font2	a bold sans-serif machine font
font3	a serif font
font4	an italic serif font
font5	an italic serif font
font6	Russian? (It looked like Times in the postscript output)

Type the name of the font file that you want to use, then press enter. Once again, you will get a menu that allows you to change some of the settings, go ahead and change whatever you want, then press y. Another window will pop up with a preview drawing of your tree. The window will have a **File** menu, which allows you to choose either **Change Parameters**, or if you are ready to make the final plot, **Plot**. Choosing **Plot** outputs a postscript file, which you can view and print with a free postscript reader such as GhostView or convert to a .pdf with an application such as Adobe Acrobat. Changing the extension to .ps may help your computer recognize it.

dnapars

[Step 4 – Analyze DNA data] Ok, now we'll try working with some DNA data. This is from a data set that I worked on for my 200A project, on the miniature floating fern Azolla. (Azolla is a very weird plant that has a few interesting features, such as hosting symbiotic nitrogen fixing bacteria, that make it pretty cool. It has been widely touted as green alternative to synthetic fertilizers in rice-growing areas, and some have even suggested it is the ideal flat crop for future space stations! But I digress…)

Go to the 200A website (http://ib.berkeley.edu/courses/ib200a/IB200A_SyllabusHandouts.shtml) and download the file AzollaITS.txt. Save it in the Phylip **exe** folder. Now, go to the **exe** folder and open **dnapars**. When prompted for the input file, enter **AzollaITS.txt**. If you already have a file called **outfile**, the program will next prompt you to choose a new name for your output file. Choose F and enter any name you like. If not, it will go strait to the menu. From the menu, choose J to jumble, and enter a random number seed and a number of times to jumble. You do not need to choose a new outgroup, as the outgroup, *Azolla nilotica*, is taxon 1. Enter Y to continue. If you already have a file called **outtree**, the program will prompt you to decide what to do with it, and you can overwrite it or choose a good name for your output tree – remember to put tree in the name though to differentiate it from the summary output file. Next, the program will run and close, and you will be left with two new files: the outfile (or whatever you named it) and the outtree (or whatever you named it.)

Open the outfile using notepad. You will see that the algorithm found 12 equally parsimonious trees. What to do? Well, you can use a program called consese to see which relationships all the trees agree on.

consense

[Step 6 – Learn to make a consensus tree]We'll talk more about consensus cladograms in lecture. For now, be aware the consense will compute strict and majority-rule consensus cladograms. Here's how to use consense to compute a majority-rule cladogram of the Azolla results:

-Open CONSENSE, type the name of your tree file.

(You may be prompted to enter a new name for your output summary. Give it a name like acs.txt, for azolla consensus summary)

-You'll see the usual PHYLIP style menu. For now, just type y and press enter. We'll make the first consensus summary just so we can see what numbers PHYLIP has assigned to all the taxa, and figure out what number it gave the outgroup.
(You may be prompted to enter a new name for your output tree. Give it a name like ignoreme, since you won't be using this file.
- Press enter to quit.
- Open the acs.txt file using notepad. At the top of the file there is a list of taxa, which are numbered. The outgroup, nil82, (short for *Azolla nilotica*,) is taxon number 26.
-Open CONSENSE again, and type the name of your tree file again.
-A screen of options should appear again. You can change outgroup rooting by entering the O, then 26.
-The CONSENSE results will be written into the outtree file and outfile. If there you already have files called outtree and outfile, you will be given the opportunity to choose whether to overwrite them or give the output files a different name. **[Step 7 – Save these files to print out and turn in later]** You can check your results by opening the files in Word or notepad, and you can look at your consensus tree using plottree or plotgram.

Some other things you can use the PHYLIP package to do:

mix – you can use mix to analyze discrete data that is formatted as 0 or 1 with no gaps. It can use Wagner or Camin-Sokal parsimony to create cladograms. Wagner parsimony is the default, and it allows character state transitions from 0 to 1 and from 1 to 0. Camin-Sokal parsimony allows character state transitions from 0 to 1, but not from 1 to 0 (i.e., reversals are prohibited). Both methods search for the most parsimonious cladogram or cladograms.

Other old fashioned methods – from various distance methods (**neighbor joining, UPGMA**) to choosing the a phylogenetic tree based on the largest set of exclusively compatible characters (**clique**).

Appendix

Where to get & how to install PHYLIP:
http://evolution.genetics.washington.edu/phylip/getme.html
More information about how to use PHYLIP:
http://evolution.genetics.washington.edu/phylip/phylip.html

To create files directly for Phylip:

1. First, open Text Edit (or Notepad or a similar program) on your computer. They can be prepared in any editor, but it is important that they be saved in Text Only ("flat ASCII") format, not in the format that word processors want to write (in Microsoft Word, make sure that the data encoding used is "US ASCII.")
2. On the first line, put in some spaces. Then, add the number of taxa in the data set.
3. Insert another four spaces, and then put in the number of characters in the data set. You can always come back and add this later, but don't forget!
4. On the next line, enter your first taxon. The taxon name must be 10 characters long, including spaces and punctuation. After the taxon name put the character states.
5. An example finished entry should look something like: `Ancestor  000000`.
6. Enter the rest of your taxa on separate lines. A finished data matrix with 5 taxa and 6 characters should look like this:

```
     5    6
Ancestor  000000
L.longipes000000
L.leyseroi111111
L.tennella111111
L.gnaphalo111100
```

7. Save your file as a text file in the following PHYLIP exe folder. Using a `.txt` extension will make it easy to look at later.

What the PHYLIP menu options do

```
Setting for this run:
  U                 Search for best tree?  Yes
  S                       Search option?  More thorough search
  V             Number of trees to save?  100
  J     Randomize input order of species?  No. Use input order
  O                       Outgroup root?  No, use as outgroup species 1
  T             Use Threshold parsimony?  No, use ordinary parsimony
  W                      Sites weighted?  No
  M           Analyze multiple data sets?  No
  I            Input species interleaved?  Yes
  0   Terminal type (IBM PC, ANSI, none)?  ANSI
  1    Print out the data at start of run  No
  2  Print indications of progress of run  Yes
  3                       Print out tree  Yes
  4         Print out steps in each site  No
  5  Print character at all nodes of tree  No
  6       Write out trees onto tree file?  Yes

  Y to accept these or type the letter for one to change
```

This is the PHYLIP menu, which allows you to se various options relating to the things the program can do.

The Weights (W) option takes the weights from a file whose default name is "weights". The weights follow the format described in the main documentation file, with integer weights from 0 to 35 allowed by using the characters 0, 1, 2, ..., 9 and A, B, ... Z.

The User tree (option U) is read from a file whose default name is `intree`. The trees can be multifurcating. They must be preceded in the file by a line giving the number of trees in the file.

The J (Jumble) option. This causes the species to be entered into the tree in a random order rather than in their order in the input file. The program prompts you for a random number seed, and for how many times you want to restart the process.

```
      5    6
Alpha      CCACCA
Beta       CCAAA
Gamma      CAACCA
Delta      AACAAC
Epsilon    AACCCA
5    6
Alpha      CACACA
Beta       CCAACC
Gamma      CAACAC
Delta      GCCTGG
Epsilon    TGCAAT
```

The M (Multiple data sets) option. In menu programs there is an `M` menu option which allows one to toggle on the multiple data sets option. The program will ask you how many data sets it should expect. The data sets have the same format as the first data set. Here is a (very small) input file with two five-species data sets:

The main use of this option will be to allow all of the methods in these programs to be bootstrapped.

The O (outgroup) option. This specifies the number of the particular species which will be used as the outgroup in rerooting the final tree when it is printed out. It will not have any effect if the tree is already rooted or is a user-defined tree.

The T (threshold) option. This sets a threshold such that if the number of steps counted in a character is higher than the threshold, it will be taken to be the threshold value rather than the actual number of steps. The user is prompted for the threshold value.

The S (search) option controls how, and how much, rearrangement is done on the tied trees that are saved by the program. If the "More thorough search" option (the default) is chosen, the program will save multiple tied trees, without collapsing internal branches that have no evidence of change on them. It will subsequently rearrange on all parts of each of those trees. If the "Less thorough search" option is chosen, before saving, the program will collapse all branches that have no evidence that there is any change on that branch. This leads to less attempted rearrangement. If the "Rearrange on one best tree" option is chosen, only the first of the tied trees is used for rearrangement. This is faster but less thorough. If your trees are likely to have large multifurcations, do not use the default "More thorough search" option as it could result in too large a number of trees being saved.

The M (multiple data sets option) will ask you whether you want to use multiple sets of weights (from the weights file) or multiple data sets. The ability to use a single data set with multiple weights means that much less disk space will be used for this input data. The bootstrapping and jackknifing tool Seqboot has the ability to create a weights file with multiple weights.

The O (outgroup) option will have no effect if the U (user-defined tree) option is in effect. The T (threshold) option allows a continuum of methods between parsimony and compatibility. Thresholds less than or equal to 1.0 do not have any meaning and should not be used: they will result in a tree dependent only on the input order of species and not at all on the data!