Lab 02:

Hypotheses of homology
How to handle phylogenetic data and trees;
Introduction to command line + R;
Introduction to Nexus and Newick formats;
Introduction to FigTree and Mesquite

Updated by Ixchel González-Ramírez

1 Before you begin

1.1 Lab grading

These labs are graded based on participation. The way I'll keep track of your participation is by asking you to keep copies of various files or answer questions. At the end of the exercise you'll email me the files and answers to your questions.

1.2 Software needed

Please download and install the following software:

1. Mesquite: `https://www.mesquiteproject.org/Installation.html`
2. FigTree: `http://tree.bio.ed.ac.uk/software/figtree/`
3. A command line interface:
   - For Macs, make sure you can find the Terminal program on your computer
   - For PCs, make sure you can find command prompt
   - Common commands for both Linux/ Mac and PC: `https://goo.gl/LyqjHr`
4. A plain text editor such as:
   - Sublime Text: `http://www.sublimetext.com/`
   - TextWrangler (Mac only): `http://www.barebones.com/products/textwrangler/`
   - Notepad ++ (PC only): `http://notepad-plus-plus.org`

1.3 Files needed

Please download the following files:

1. Amblygnathus.nex `http://ib.berkeley.edu/courses/ib200/labs/02/Amblygnathus.nex`. Right click > Save Link As to download file.

2 Introduction

Today we will experience the process of coding morphological data. Then we will learn about the primary ways to visualize and represent trees, as well as the most essential programs we will use throughout the course. We will review common ways to manipulate files through the **command line** and **RStudio**. We will also become familiar with **Nexus** and **Newick** file formats, which are widely used by phylogenetic programs. By having common file formats the results of one program can be viewed or analyzed in another (although this does not always work as well as one would hope). You can manipulate these file formats directly in a text editor, but often it is better to use a program like **Mesquite** or **FigTree** and save the results to be used in another program. By the end of the lab, we will built a phylogeny with our own generated data.

3 Building a morphological matrix

The process of building a morphological data matrix is complex. By designating characters and character states we are constructing hypothesis of homology under the assumption that shared character states are due to common ancestry. Later in the semester we will discuss that the same assumptions are true for molecular data. Today we will use the **caminalcules**, a group of imaginary organisms created by the professor Joseph H. Camin, to explore the way a biologist would approach the study of morphological traits in living organisms.

1. Individually, observe the caminacules. All of them are living organisms. We want to infer their evolutionary history based on their morphology. Identify three characters that you consider are helpful to infer their relationships. Assign character states for **all** of them.

2. Find a fellow (Try to work with a person you don't know!). Discuss your character choices. Are they the same? Are they contradictory?

3. Work together to agree on 10 characters that will be useful to build a caminacules phylogeny. Keep a written record of the characters and character states, you will use it in the last section of the lab.

4 Intro to the Command Line

Feel free to skip this section if you already know how to create directories, copy files, and delete things using Terminal.

The command line gives you more direct access to the actions your computer is performing. You can do everything from basic actions such as copying a file to complex tasks like running an entire analysis. Some programs used in phylogenetic analyses can be accessed only through the command line, such as RevBayes, and many others can be accessed either through the command line or through a Graphical User Interface (GUI). GUIs are what we think of as programs; they facilitate our interactions with computers through buttons, words, pictures, etc. The command line is an extremely powerful tool, and the goal for today is just for you to become more comfortable with using the command line and provide you with some resources for moving forward. Since Macs and PCs run on different systems, the command line prompts are a bit different. I wrote different instructions for mac and windows users. In either case, check this site for a handy list of translations between systems for common commands.

4.1 Mac/Linux user

Open the Terminal. Note that I will indicate commands you should type with a $ symbol- do not type the $, just what follows it. Typing `$ pwd` prints your current working directory. This tells you where you currently are in your computer. If you run an analysis that outputs a file, it will be saved to your working directory. You can change your working directory with the command `$ cd`. Change your working directory to the folder where the Amblygnathus.nex Nexus file is located by typing (for example):

```
$ cd /Users/ixchel/ib200/labs/lab02
```

Once in this new directory, the command `$ ls` shows you all the files you have in that folder (make sure that the Amblygnathus.nex file is there). Create a new folder with `$ mkdir test_folder` and use `$ ls` to make sure it worked. Now copy your Nexus file into your new folder using the copy command:

```
$ cp Amblygnathus.nex test_folder
```

In this command, we tell our computer to copy with `cp`, we tell it what to copy with `Amblygnathus.nex`, and we tell it where to copy with `test_folder`. Note that this command will only work if you have set your working directory as instructed. Your computer will look for the file `Amblygnathus.nex` in whatever directory you are currently in. Now let's delete the folder we just created (and the copied Nexus file inside).

```
$ rm test_folder
```

Uh oh! This didn't work because we are not trying to delete a document, but rather a folder or directory. Try this instead:

```
$ rmdir test_folder
```

Oh no! This didn't work either because the directory we are trying to delete contains items. Try this instead:

```
$ rm -r test_folder
```

That's all for now, but there are many useful tutorials available online, such as this one. We will use the command line a bit later in the semester. A word of caution: it is possible to delete every file (including your own operating system) using the command line. Be very careful when trying out new things, especially if they involve the `rm` command!!

4.2 Windows user

Open Command Prompt. Note that I will indicate commands you should type with a $ symbol- do not type the $, just what follows it. Typing `$ chdir` prints your current working directory. This tells you where you currently are in your computer. If you run an analysis that outputs a file, it will be saved to your working directory. You can change your working directory with the command `$ cd`. Change your working directory to the folder where the Amblygnathus.nex Nexus file is located by typing (for example):

```
$ cd C:/Users/ixchel/Documents
```

Once in this new directory, the command `$ dir` shows you all the files you have in that folder (make sure that the Amblygnathus.nex file is there). Create a new folder with `$ mkdir test_folder` and use `$ ls` to make sure it worked. Now copy your Nexus file into your new folder using the copy command:

```
$ copy Amblygnathus.nex test_folder
```

In this command, we tell our computer to copy with `cp`, we tell it what to copy with `Amblygnathus.nex`, and we tell it where to copy with `test_folder`. Note that this command will only work if you have set your working directory as instructed. Your computer will look

for the file `Amblygnathus.nex` in whatever directory you are currently in. Now let's delete the folder we just created (and the copied Nexus file inside).

```
$ del test_folder
```

The computer will ask you if you want to delete the folder, tell it yes. Did it work? Try `$ dir` command. For PC, this command deleted the file inside the folder, but not the folder itself. This didn't work because we are not trying to delete a document, but rather a folder or directory. Try this instead:

```
$ rmdir test_folder
```

That's all for now, but there are many useful tutorials available online, such as this one. We will use the command line a bit later in the semester. A word of caution: it is possible to delete every file (including your own operating system) using the command line. Be very careful when trying out new things, especially if they involve the `del` command!!

5 Making (Probably Ugly) Trees in R

R is a very powerful and popular coding language developed for statistical computing and graphics. It is widely used in phylogenetics for performing comparative methods analyses, but is rarely used for building phylogenies. We will have a longer introduction to R later on in the semester, but here's a little to get started.

In the next section you will use FigTree to make pretty trees. Why, then, are we going to bother plotting trees in R? It is less user friendly and in general less pretty than FigTree. Often, you may run analyses in R and want to make a figure to accompany the analysis. If you write a script (a file with many lines of code), you can easily rerun the same analysis even if your data changes. The same goes for making figures. Making your figures in R makes your entire process more streamlined and reproducible (even if it is more of a pain initially).

Open RStudio. RStudio is a free and open source Integrated Development Environment (IDE) for R. Basically, it makes coding in R a bit more intuitive and user friendly. Look in the left of the RStudio screen - it should say Console in the corner. This is where you type your commands. If you want to generate a script (you always do) with a record of what commands you ran, open a new R script by going to `File>New File>R Script`. This creates a new text file where you can write commands and send them to the Console.

Like we did in the Command Line, first set your working directory using the command `> setwd("working_directory")`. In R, I'll use the `>` symbol instead of the $ to indicate a line of code. Don't type the `>` symbol! Set your working directory to the folder where your Amblygnathus.nex file is saved. Run the line of code by pushing the **Run** button in the top right corner of your R Script, or by pushing **command + return** on a mac or **control + Enter** on a PC. This 'sends' the line of code to the Console and executes the command. I'll ask you to send me your R script at the end of lab, so be sure to write all of these commands in that file before sending them to the Console. Load the library ape, a commonly used phylogenetics package:

```
> library(ape)
```

Unless you have used ape before, you will get an error. This means that you need to first install the package to your computer before you can load it into your current R Session. `> install.packages("ape")`

Now rerun the first line:

```
> library(ape)
```

We will use a command from the ape package to read in the information in the Nexus file.
`trees <- read.nexus("Amblygnathus.nex")`
This command tells R to save the information in the Nexus file into an object named trees. If we want more information about the object, we can ask R what class it is:
`class(trees)`
and what the object contains: `ls(trees)`
Note the difference between the `ls` command in R and in the Command Line! Now let's plot the trees:
`plot(trees)`
Notice that this goes through and plots all of the trees. For now, focus on just the first tree:
`plot(trees[1])`
You can get more information on the command we are using by type a question mark before the command:
`?plot.phylo`
This shows you what additional arguments the function has (i.e. what else you can change in the plot). For example, here's a really ugly tree:

```
> plot(trees[1], font = 2, edge.lty = 4,
edge.width = 10, tip.color = c("red", "green"))
```

With a little bit of work, you can make really nice figures for publication in R; it just takes some time to learn the parameters and options. Play around a bit with these arguments and make your own ugly (or pretty) tree in R. You can save a PDF of your tree by sandwiching the plot command between two additional lines of code, like this:

```
> pdf("ugly_r_tree.pdf")
> plot(trees[1],...) (Put the plot command for your customized tree here)
> dev.off()
```

When you are done, save your R script by going to `File>Save As`.

6 Making Trees Pretty in FigTree

FigTree is developed by Andrew Rambaut's research lab, who are well known for developing the the Bayesian Evolutionary Analysis by Sampling Trees (BEAST) software package. FigTree is a highly useful way to quickly view phylogenetic trees and to produce publication-ready figures. Here we'll just cover some quick basics.

Open the `Amblygnathus.nex` file in FigTree. `Amblygnathus.nex` is a Nexus file, and it contains multiple phylogenetic trees in the same file. Use the arrow buttons at the top of the screen to scroll through the trees. Try experimenting with the options under the `Layout` tab. Use the `Selection Mode` tab (at the top of the screen) to select a `Clade`. Color one of the clades on the trees. Under the `Trees` tab (left side of window) click `Order nodes` to *ladderize* the tree. Using the `Node Labels` tab (left side of window) add the node ages to the tree. Explore other options to modify the appearance of the tree. Under the `File` menu, export the ladderized tree with the colored clade and node labels (and any other changes you feel like making) as a pdf. You'll email me the pdf file when you complete the lab.

Now let's convert the Nexus file into a Newick file. Remember, the Nexus file contains multiple trees, so to export the single tree you've been editing follow these steps: under the `Edit` menu select `Copy`, and then under the `File` menu select `New`. Then under the `Edit` menu select `Paste`. Now select `Export Trees` under the `File` menu and choose `Newick` as tree file format. Name your file `Amblygnathus.new`. Note: if you have a clade selected when

you do this, you will export a file that only contains that clade! Be sure that you do not have a clade selected when you copy.

7 Newick and Nexus File Formats

In a plain text editor, open the Newick file you just saved. Newick format is a commonly used way of representing tree topologies as text. Put simply, monophyletic clades are surrounded by parentheses and sister clades are separated by commas. For example, a simple tree could be written as (((A,B),C),(D,E)). Newick format also contains information about branch lengths (after colons) and node names (after closed parentheses).

Now let's take a look at the Nexus file `Amblygnathus.nex` by opening it in a text editor. Every Nexus file starts out with `#NEXUS` and then is followed by a brief description of the file surrounded by brackets. This is followed by the actual data of the Nexus file, which are organized into several blocks. Each block starts with a line `BEGIN BLOCK NAME;` and finishes with an `END;`. The lines between, which hold the actual data, are often indented. Each program creates and uses different blocks with information and commands that are particular to it, but there are several block types that are almost universally used and contain the most fundamental information for phylogenetic analysis.

The data block (ie. `BEGIN DATA`) contains your data matrix. Some Nexus files use a taxa block (ie. `BEGIN TAXA`) and a character block (ie. `BEGIN CHARACTERS`) instead of the data block. Most programs, but not all, are flexible and can use either format. The data block basically contains all the same information as the other two blocks.

The data block must contain a `DIMENSIONS` and a `FORMAT` line, which describe the data in the matrix. Datatype determines the basic class of data: discrete; continuous; protein; or DNA. After that there are several commands, which describe what symbols are allowed and what they represent. For example, does a `-` mean unknown or missing. The characters block may also contain other information such as `charlabels` (names for the different characters) and `statelabels` (names for the different character states for each character) or `charstatelabels` (both types of info combined in one command). Next is the actual data matrix `MATRIX`. Each line of the matrix starts with the name of the taxon represented and is followed by a series of symbols representing the character states for the various characters in the matrix. Each taxon line must have exactly the same number of characters in the same order.

You will also see `BEGIN TREES`, which is the start of the trees block that contains your phylogenetic tree information. A Nexus file will only contain a tree block, when it is necessary to import a tree into a program. Tree blocks often start with a `TRANSLATE` command, which is required for a number of programs, but not all. It is a list of consecutive numbers followed by the names of the taxa that those numbers will represent. This is followed by a `TREE` line (in Amblygnathus.nex there are 21 different trees), which describes the tree in Newick format using the numbers assigned in the translate command for the names of the taxa.

8 Intro to Mesquite

Mesquite was developed by Wayne and David Maddison (twin brothers who are both phylogeneticists!) as a tool for interpreting phylogenetic information. The strengths of Mesquite are creating and editing data matrices, examining the distribution of features on a phylogeny, and testing hypotheses about character evolution. This program will be a great way to explore the data you collect for your final project, especially if you are not quite yet com-

fortable with R. In Mesquite it is also possible to reconstruct ancestral states for continuous and discrete characters under different parsimony models (for example unordered states). However, Mesquite is NOT the program you should use for any of your tree-building. You need to implement maximum parsimony, maximum likelihood, or Bayesian inference in another program such as PAUP* or MrBayes to build your trees.

Mesquite is modular, which means that the program is set up as a bunch of modules that all do different functions, such as draw a picture or do a parsimony analysis. Some modules use other modules and the modules are used in combination to perform an analysis. Thus Mesquite is very flexible and capable of doing analyses that it was never intended to do. Unfortunately it also means that Mesquite can be difficult to use, because it is not always clear where to find the appropriate command in its menus.

9 Editing a Data Matrix in Mesquite

Open Mesquite and select `File>Open` to open the `Amblygnathus.nex` file. A *Project* window will open showing the *Character Matrix*. Here you can edit and add characters and character states. There are several tools along the left side that allow you to manipulate the matrix. When you hold the cursor over each of the buttons, a description of what it does appears at the bottom of the window.

To the far left is a panel showing some of the other viewing options. Click the `Taxa>List & Manage Taxa`. The *Taxa Block* can be used to edit information about taxa. Change some of the taxon names.

Now go back to the *Character Matrix* by clicking the tab at the top of the window, or the button on the far left. Now let's add some data to the matrix. Go to the tab `Matrix>Add characters`. Select the number of characters you want to add. At the bottom left are five small buttons next to a blue i that look like little windows. Select the one second from the right, the `Show State Names Editor Window`. In this window you can update the names of characters and character states, just make up some characters. Now go back to the *Character Matrix* by clicking the tab at the top of the window. Make sure you have selected the `Edit tool` on the left panel and entry the values for your character matrix. Feel free to explore more functionalities of Mesquite, in particular under the `Display` menu. You will find functionalities such as color the character states, that are helpful when constructing a matrix.

Now save the Nexus file with your new characters. When you finish the lab you'll email me a copy of the modified Nexus file.

Final excercise: Create a matrix of 10 morphological characters for the Caminalcules. For this, go to `file>new`, save the new file as `caminalcules`. Once your matrix is ready we will build a tree using Mesquite (I know! I told you not to do that, but this is just for you to have a tree at the end of your first lab. We will spend a lot of time later in the class learning how to infer trees.) Go to `Analysis>Tree Inference>Tree search>Mesquite Heuristic`, select `Treelength` as the criterion for tree search, and accept all the rest of the default settings. Mesquite will search for the tree that implies the least number of changes in the character matrix you built. Once it has finished a tab called `trees from Mesquite` will appear on your left panel. At the top of the window select `Analysis:tree>Trace character history` and accept the defaults. A small tab will appear at the top left of your screen and you will be able to explore the history of each character in the caminalcules phylogeny. Save your Nexus file.

Please email me the following:

1. Your R script file.
2. PDF file of the ugly tree you made in R.
3. PDF file of the pretty tree you made in FigTree.
4. The Nexus file you modified by adding a new character in Mesquite.
5. The Nexus file with the Caminalcules data matrix and trees.