Day 1: Linux, Alignment and Parsimony

Jesus Martinez-Gomez

November 4th, 2019

Abstract

This worksheet will walk you through some Linux basics, alignment and parsimony tree inference. These are are step-by-step instruction intended for someone who does not have familiarity with Linux or phylogenetics. As such follow them as closely as you can. However, if do have familiarity, follow the instructions at your leasuire. Make sure you answer the homeworks at the end.

Table of Contents

Linux Basic

Many free bioinformatic programs, including phylogenetics are often made by a one or a few academics who don't have the resources to make nice graphical user interfaces (GUI). GUIs are the way most computer users interact with computers, they are the point and click apps (e.g., Google chrome, adobe reader, Micorosoft word etc), the folder navigation system that we all use everyday. Some bioinformatic GUI exisit but typically need to pay for such them (i.e., Sequencher, Genious). Most bioinformatic programs typically lack the bells and whistles and come in the form of an 'exacutable.' Bioinformatic executables are built using the Linux an operating system (OS), like Mac and Windows but less common. Both Linux and Mac OS are largely based on another operating called Unix, as such if you own a Mac you will have the necessary programs built in to some of the steps below. Windows OS is not Unix based instead it is MS-DOS so you will need to install some additional programs.

Linux, Mac and Windows all have a command line interface, this is a way of navigating your computer without a mouse, and how executables ran. In the first part of this worksheet we will introduce you to the Linux command line interface (it'll make you'll feel like your a hacker). Additionally instead of running things locally (i.e., on your own computer) we will run all our analysis on a remote server, this has a few advantages: 1. Bioinformatics is an increasingly popular dicipline, even if you plan on doing a fully wet-lab based Ph.D, our biased opinion, is learning some of these Linux based skills will be usefull to you. Especially considering the possibilitiy of shut down #Rona 2. If you don't ever use phylogenetics again, you'll just have these programs randomly installed. 3. If you will use phylogenetics, in the age of big data, your computer likely doesn't have the computational resource to run things. You'll likely need to use a server

anyways.

Please please please please please ask us question if you have trouble!

An important tip for first timers If this is your first time using linux/ coding/ scripting, you might experiance some frustration, as you would learning any new skill. My recommendation is to **TAKE IT SLOW**, a single typo misplaced period will cause things to fail and because you will likely lack skills necessary to troubleshoot, you might get super frustred. As such, read everything carefully, take a lot of deep breaths, take a yoga break, eat some pie, whatever you need to chill out. We are here to help you succeed so please please ask us all the questions!

Alrighty then, let get to it!

Remote login

The first step is to log into the server from your computer. Depending on whether you are a Mac/ Linux or Windows user follow the instructions below. Importantly for instructions below you all have a specific user name and password. The user name is your, 'net_id@[IP.address]'. We have two servers since the are about 1/2 Mac and 1/2 Windows user we are splitting you up between them. The IP address will depend on your computer time. Your password is your 'first_name6410', first letter capitalized. For example for me: * jm2722@[IP_Address] * Jesus6410

Windows Users

You will need to download a few programs

1. Download putty, this is a program for remote login
2. Download WinSCP, this is a program to upload and download files.

Logining in

1. Open putty, you'll be promted to type your location

128.253.192.48

Mac/ Linux User

Open the Terminal app. For Mac users it'll be under Applications/Utilities/. This should open a black screen with a flashing light that you can type things in. Type the foll

```
ssh cornell_net_id@128.253.192.48
cornell_net_id6410
```

Basic Linux Command

Now that you have successfully logged you are all in a Linux envrionment. We will run through a few basic linux commands. As I mentioned, Terminal essentially allows you to navigate your computer through a command line rather than a point-and-click folder system. You'll quickly notice that it is not as intuative as point-and-click. It's retro so just pretend you are living in the 80's lol. Linux has a set of built in commands to help you navigate the program. These commands are essentially little built in programs.

Moving around with Linux

In Linux knowing your location (i.e, which folder you are in) is important. Each folder has a unique location more commonly refered to as a 'path' on your computer. Think of the'path' as an address. To find were you currently are type the following.

```
pwd
```

'pwd' stands for 'print working directory'. The word 'directory' is synonyms with computer 'folder,' and will be used interchangably throughout this worksheet. All computers are essentially folders-within-folers, pwd prints the location of the working directory. The path is specified as follow, each folder name is seperated by a forward slah '/'. You should see somethign similar to the follow, except instead of my net id it will be yours.

```
/home/jm2722
```

Next it is useful to show what files and folder are in the directory you are currently in. Using the following command we can show that.

```
ls
```

'ls' stands for 'list'. This command will list all the files and folders in your current location. You should some folders you are likely familiar with including 'Downloads' and 'Desktop.' This is because the 'server' you are logging into is just a regular desktop computer.

The next step we will show you how to move between folders. For this we use 'cd' command which stands for "change directory". Type each line individually.

```
cd Destkop
pwd
```

As we just did, using cd we can move into any subfolder by typing its name. In other words we moved into the subfolder 'Desktop' because it was inside of the folder 'jm2722' (your will be your net id).

'cd' has two other important functions, the first insted of moving down into subfolders it can be used to move back "up". To to this we type the following,

```
cd ..
pwd
```

We are no long in the 'Desktop' folder we are back in the net id folder.

Lastly, we can use 'cd' to move to a directory that is not directly adjacent to our current directory we just need the full folder 'path'. We will do this in the following section,

Making and deleting new directories

An important function is to make new folders to keep your self organized. The command "mkdir" is short for 'make directory', and does just that. Lets try it out, but first lets move into Desktop

```
cd Desktop
ls
mkdir testFolder
ls
```

After you typed 'ls' the first time you'll see that there are no folder in Desktop. After you typed 'ls' the second time you will see that you have a new folder called testFolder (because you just made it). Additionally, you'll notice a backslash "", in Linux this means a line break, this way you can type multiple lines into your termina. Now lets move into that new folder.

We are now in testFolder. If we type 'ls', we will notice that there is nothing. That is because our new folder is empty. Lets go back to our previous directory.

```
mkdir testFolder \
ls
```

The function "mkdir" is short for make directory. After you typed ls you will see that you have a new folder called testFolder (because you just made it). Additionally you'll notice a backslash "", in Linux this means a line break, this way you can type multiple lines into your termina. Now lets move into that new folder.

```
cd testFolder
```

```
cd ..
```

'cd ..' is the command to move into the parent directory. Now lets delete the testFolder.

```
rmdir testFolder
```

Uploading and downloading files

There are many different way you can upload and download files depending on the server you use. For today purposes we are using the function 'scp'

These instructions are basic but will get you started. This is a link to a **Linux cheat sheet**.

Download the GitHub Repository

All the material for this course are found on a Github repository, this is essentially a cloud storage system. You can view this repository using a browser by following the **link**. You can download the repository using your browser, or you can do it in terminal. But Before you do this we will need to decide a location to put the folder. For Mac users I recommend putting it on your Desktop, for Windows users I recommend putting it in your /home/ directory.

1. First you need to naviagte to the location where you want the folder. If you use 'pwd' and you see the following, you are in the correct spot.

```
/Users/<username>/Desktop #mac users
/home/<username>  # Windows users
```

2. The following will download the repository

```
git clone \
https://github.com/Jesusthebotanist/
PLBIO6401_EcologyEvolution_Module_2019.git
```

Type 'ls', you should see your new folder. You can further convince yourself that the new folder is there by navigating to it using a finder.

We will be updating this folder with relevant files for the four modules. There should be two directories in there "Day1" and "Ruhfel_unaligned_fasta". We will primarily be working in Day1 today. At the start of each day you should re-download any updates to the folder that Jacob or Eugenio. To do this just do the following.

```
cd PLBIO6401_EcologyEvolution_Module_2019
git pull origin master
```

Making a Project Directory

In this section we will make a new project directory. Every time you start a new project, it is a good idea to create a new directory to store files you need (data/programs). The following code will create a new folder called "My_first_Parsimony_Analysis" inside of the folder "Day1". We will then copy and paste two programs and a data file into this folder. **Tipe: When you do your Homework Name the folder something different**.

Mac User

```
cd PLBIO6401_EcologyEvolution_Module_2019

mkdir Day1/My_first_Parsimony_Analysis

cp Day1/programs/mac/mafft-7.450-signed.pkg \
Day1/My_first_Parsimony_Analysis

cp Day1/programs/mac/tnt-mac.command \
Day1/My_first_Parsimony_Analysis

cp Ruhfel_unaligned_fasta/ruhf_32_by_5000_unaligned.fas \
Day1/My_first_Parsimony_Analysis

cd Day1/My_first_Parsimony_Analysis
```

Windows User

```
cd PLBIO6401_EcologyEvolution_Module_2019

mkdir Day1/My_first_Parsimony_Analysis

cp Day1/programs/windows/mafft_7.450-1_amd64.deb \
Day1/My_first_Parsimony_Analysis

cp Day1/programs/windows/tnt-linux \
Day1/My_first_Parsimony_Analysis

cp Ruhfel_unaligned_fasta/ruhf_32_by_5000_unaligned.fas \
Day1/My_first_Parsimony_Analysis
```

```
cd Day1/My_first_Parsimony_Analysis
```

Alignment with MAFFT

We will begin by aligning our code using the program MAFFT. You can learn more about the program following this link **LINK**. MAFFT has many options and it is a good idea to read the documentation on the website. For the purposes of today we will run an alignment with default settings.

Mac Instillation

In Finder simply double click on the file called, "mafft-7.450-signed.pkg". This will take you through a program instillation. You can then delete the file.

```
rm mafft_7.450-1_amd64.deb
```

Windows Instillation

```
sudo dpkg -i mafft_7.450-1_amd64.deb
rm mafft_7.450-1_amd64.deb
```

Running a Basic Alignment

If you have successfully aligned the data you should be able to run MAFFT by simplying typing in 'mafft'. To perform a basic alignment we type 'mafft' followed by our input data. Our input data is the unaligned 32 tip tree. The '>' redirects the output of mafft, into a new file. We name the new file.

```
mafft ruhf_32_by_5000_unaligned.fas \
> ruhf_32_by_5000_aligned.fas
```

Convert Aligned File from FASTA format to TNT format

Our newly aligned file is in FASTA format (.fas, .fasta). TNT does not take in this file format, instead we need to convert the file to a .tnt file. Kevin Nixon has an online converter that you can access here (**TNT CONVERTER**)[http://pwww.plantsystematics.org/plbio4400_2019/].

1. Once on the website scroll down to the section that says **UPLOAD FASTA FILE HERE FOR TRANSLATION TO TNT FORMAT**.
2. Once you click 'Submit' a new tab will open up. You will need to copy and paste the content of the tab into a new file. There are a few ways to do this:

- Copy and paste into a text editor such as, Notepad, Text Wrangler, Sublime, BBedit (**DO NOT USE TEXTEDIT**)

3. Name the file to: ruhf_32_by_5000_aligned.tnt
4. Move the file to '/PLBIO6401_EcologyEvolution_Module_2019/Day1/My_first_Parsimony_Analysis'

A Note For Windows Users You might have a hard time finding your folder. This is because Ubuntu is a computer-in-a-computer therefore if you want to see any files using a finder you need to go to this location.

```
C:\Users\<username>\AppData\Local\Packages\
CanonicalGroupLimited.UbuntuonWindows_79rhkp1fndgsc\
LocalState
```

Parsimony with TNT

We will be inferring phylogenies using the program "Tree analysis using New Technology" (TNT). This is a command line program, which means in order to run it you will use the "Terminal" app in Mac/ Linux or the "Command Prompt" app in Windows. It is developed in part by Kevin Nixon a Plant Biology faculty! The following is a worksheet that you can follow to perform a basic parsimony analysis in TNT. To learn more about TNT check out there website (**LINK**)[http://www.lillo.org.ar/phylogeny/tnt/]

Mac Instillation

```
sudo chmod +x tnt-mac.command
./tnt-mac.command
```

Windows Instillation

```
sudo chmod +x tnt-linux
/tmp/phymod_prog/tnt64/tnt
```

Introduction to TNT

First a brief note on syntax. Most TNT commands are a simple word or abbreviation, but in order to get the program to run them they must end with a semi-colon ";". So for example, try typing in the following command "help".

```
help;
```

The "help" command above will give you a list of all the commands that TNT can run. The "help" command can also be used to find information about other commands. For example we will be using the "mult" command later on in this worksheet. To find out more information about "mult". Type the following.

```
help mult;
```

If you are ever unsure about what a particular command does always use "help"

TNT Analysis

Setting Up TNT Analysis

We will now perform a basic parsimony inference. The first step is to specify settings in TNT. The "log" commands will generate a text file, (ruhf_32_by_5000_log.txt) that will keep track of all the subsequent commands you type and the output TNT produces. It is a good coding practice to keep log files, as they are a written history of the analysis you've done (future you will thank you).

```
log ruhf_32_by_5000_log.txt;
```

Look at the folder, a new file called ruhf_32_by_5000_log.txt should be in there. You should be able to open this up by double clicking on it. It should be empty right now, but will fill up after we are done. The second step is to set the RAM memory usage. When TNT runs it takes up memory on your computer (like any program), the default setting for TNT memory usage is low. We'll set it to 200mb which is good for our tree size and no problem for a modern day computer. If you are analyzing a larger tree, consider setting the memory to something larger.

```
mxram 200;
```

The next step is to read in our data.

```
p ruhf_32_by_5000_aligned.tnt;
```

TNT searches through tree space to find the most parsimonious tree or set of most parsimonious trees. TNT searches through many trees most of which are not good (they have a bad parsimony score). Therefore we don't want to save

every single tree, instead we will use the "hold" command to save the top 1000 trees.

```
hold 1000;
```

Basic TNT Analysis

Perform a basic parsimony search. You may notice that instead of a single command there are four in the following line of code.

```
Mult; Rat; Drift; Sect;
```

This analysis may take a few seconds. and you will see some fun stuff pop up on your screen. Once its done you can view your most parsimonious phylogeny in your terminal window using the following code.

```
tp;
```

Next we will perform a bootstrap. Bootstrap is a used as a measure of support for particular topologies. Consult the lecture for more notes.

```
resample replications 100;
```

A tree will be printed with your bootstraps. We will visualize our tree in a different program.

Exporting and Quiting

Now the we have successfully completed an analysis we may want to export our best tree into a file for safe keeping. First we will use the command "taxname=", TNT changes our tip name during this analysis, "taxnames=" changes them back. Next we will use the 'ttags);' command, this will save our boot strap values from our previous analysis. Next we use the "export" command to generate a .nex file, which stand for Nexus. A Nexus file is a standard phylogenetic file.

```
taxname=;
ttags );
export - ruhf_32_parsimony.nex;
```

Double check to see if a file called myPhylogeny.nex was created. Finally we can exit out of TNT using the "quit" command

```
quit
```

Open the ruhf_32_by_5000_log.txt file again (if you already have it open, close it first), it should now be full of all the commands you inputed and the TNT output! Yay phylogenies! A final note. If you rerun the above code it will overwrite any files that exist on your computer. If you are running a different analysis make sure you change the names of the .log file and the .nex file

Viewing your Phylogeny with FigTree

You should now have a .nex file in your folder. This .nex file contains the phylogeny you just inferred and can be viewed in a tree viewer. To view your phylogeny download **FigTree**.

1. Open Figtree
2. Open ruhf_32_parsimony.nex in FigTree, it should propt you to name somthing. Name it 'Bootstrap'. Once you have a phylogeny should appear
3. Click the drop down arrow on 'Node Labels' on the left hand tool par. Then click on the drop down arrow for 'Display:' you should see 'Bootstrap' click on that.

Homework Questions:

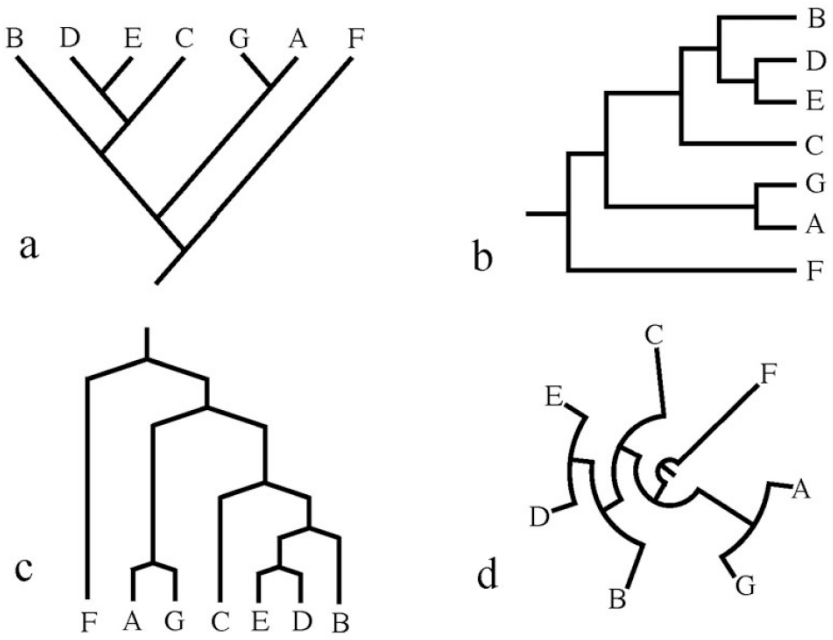
The following is the homework assignment. For question 1 and 2, please write your answers in a word document. For question 3 email me a folder with all your files. **Email: jm2722@cornell.edu**

Homework Question 1:

In less then 3 sentences explain why a human hand and a Gorrilla hand look alike?

Homework Question 2:

Which of the four trees below depicts a different pattern of relationships than the others? In less than 3 sentences explain why using at least three words from the "Tree Thinking Reading Trees" section of lecture.



Homework Question 3:

Rerun through the worksheet using a different gene file from the "Ruhfel_unaligned_fasta" folder. Do everything inside a new directory named, [YourLastName_GeneName]. Note for Jacobs lab you will be infering a maximum likelihood phylogeny and baysian phylogeny using this same gene

About This Doucment

This document was written in markdown and knitted into a PDF with Pandoc, using the following code.

```
pandoc Day1_Worksheet_LinuxAlignmentParsimony.md \
-f gfm \
-t latex \
--toc \
-V toc-title:"Table of Contents" \
-V linkcolor:blue \
-M title="Day 1: Linux, Alignment and Parsimony" \
-M author="Jesus Martinez-Gomez" \
-M date="October 19th, 2020" \
```

```
-M abstract="This worksheet will walk you through some \
Linux basics, alignment and parsimony tree inference. \
These are are step-by-step instruction intended for \
someone who does not have familiarity with Linux or \
phylogenetics. As such follow them as closely as you \
can. However, if do have familiarity, follow the \
instructions at your leasuire. Make sure you answer \
the homeworks at the end." \
--extract-media ./images \
--highlight-style tango \
-o Day1_Worksheet_LinuxAlignmentParsimony_Oct2020.pdf
```