Primate Phylogenetics Lab Code

Isaac Krone

2025-06-18

# Part A: Morphological Analysis

We’ll use two software packages (Phangorn and pytools) to build phylogenetic trees and measure their properties. Both of these software packages are written in a coding language called “R” that is widely used by biologists.

## 1) Software installations and loading

If you are using the RStudio application or the RStudio environment in Juypter notebooks, run the code block below by clicking the green play button in the upper right-hand corner. This step will take a moment, and you’ll see some normal code ouptut.

#install.packages("phangorn", quiet = TRUE)  
#install.packages("phytools", quiet = TRUE)  
library(phytools)  
library(phangorn)

## 2) Data import and explorations

Now that we have installed the proper software, we can load a character matrix. The code below reads the file “Morphological\_Data.csv”, which is the same character matrix that you see in your lab worksheet.

#read character matrix  
M\_characters<- as.matrix(read.csv("Morphological\_Data.csv", row.names = 1))

To see that this character data matrix is correct, we’ll print it out.

M\_characters

## X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12  
## Solenodon 0 0 0 0 0 0 0 0 0 0 0 0  
## Chimpanzee 1 1 1 1 1 1 1 1 1 1 0 0  
## Galago 1 0 0 0 0 0 0 0 0 0 1 1  
## Gibbon 1 0 1 1 1 0 0 1 1 0 0 0  
## Gorilla 1 1 1 1 1 1 1 1 1 0 0 0  
## Human 1 1 1 1 1 1 1 1 1 1 0 0  
## Lemur 1 0 0 0 0 0 0 0 0 0 1 1  
## Marmoset 1 0 1 1 0 0 0 1 1 0 0 0  
## Orangutan 1 0 1 1 1 1 1 1 1 0 0 0  
## Tarsier 1 0 0 0 0 0 0 0 1 0 0 1

Next, we’ll convert this matrix into a format that the phangorn package can use. It’s still the same data, just arranged a bit differently.

M\_phydata <- phyDat(M\_characters,type = "USER", levels = 0:1)

## 3) Tree building and plotting

Now we’re ready to build a phylogenetic tree. The bab function (“**B**ranch **A**nd **B**ound”) finds all of the most parsimonius trees for a given character matrix. All of these trees will be stored with the name morpho\_trees.

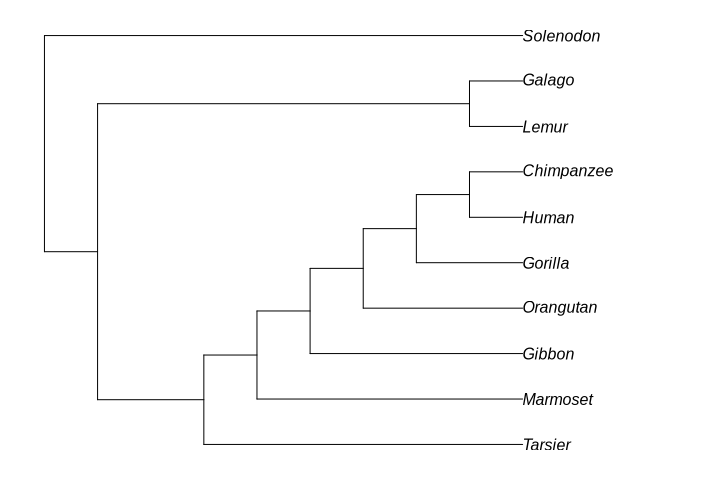
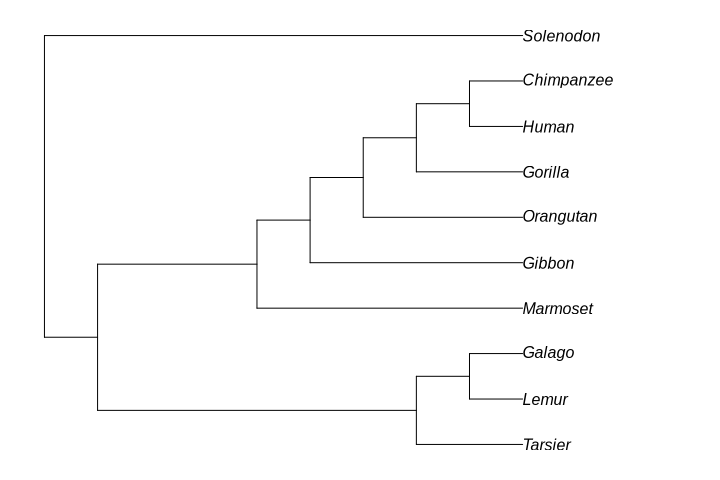
morpho\_trees <- bab(M\_phydata)

Since we are using *Solenodon* as an outgroup, we’ll designate it as such using the root function.

morpho\_trees <- root(morpho\_trees, "Solenodon", resolve.root = TRUE)

Now we’ll plot all of our most parsimonious trees.

par(mai = c(0.2,0.2,0.2,0.2))  
  
plot(morpho\_trees)



We found 2 most parsimonious trees. But how many trees are possible?

The number of possible phylogenetic trees for any number of tips is given by the formula below.

We can use the function howmanytrees to solve this for any given n. We have 10 taxa, but one of them, the outgroup, we already know the position of. So really, we are solving from n=9, since we want to know all of the relationships between the ingroup taxa.

howmanytrees(9)

## [1] 2027025

Use this to answer Question 4A on your lab handout.

The code below measures the parsimony of our two most parsimonious morphological trees.

paste("Tree 1 parsimony score:", parsimony(morpho\_trees[[1]],M\_phydata))

## [1] "Tree 1 parsimony score: 13"

paste("Tree 2 parsimony score:", parsimony(morpho\_trees[[2]],M\_phydata))

## [1] "Tree 2 parsimony score: 13"

Use this to answer question 4B on your lab handout.

**Work on questions 4 - 13 on your lab handout before moving to part B.**

# Part B: Molecular Phylogeny

## 1) Data import and explorations

Using the same method we used for the morphological data, we can build a parsimony tree for the Epsilon Hemoglobin Gene.

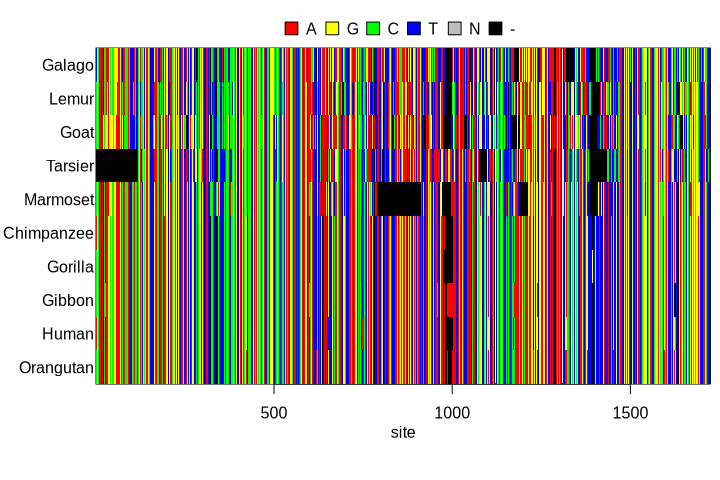
First, load in genetic data. These data are stored a bit differently.

Hemoglobin <- read.phyDat(file = "Epsilon\_Cooked.fasta", format = "fasta")

Since we are working with DNA data, we have 4 character states corresponding to the 4 bases in DNA, “A”,“G”,“C”, and “T”.

Let’s take a look at what these data look like. The code below plots out a visual representation of the character matrix. Think of it as several DNA strands stretched out and laid next to each other. On the x axis, we have **sites**. A **site** is a specific base on a DNA strand.

#set up the plot space  
par(mai = c(1, 1, 0.5, 0.1), cex.lab = 0.1)  
#plot out our alignment  
image(Hemoglobin)  
mtext(text="site", side=1, line=2)



Epsilon Hemoglobin gene alignment.

Notice all the black space: this corresponds to sites where genetic data are missing for certain taxa. It could be missing because of problems sequencing the DNA. But it’s more likely that those data are missing because the animals don’t have that chunk of DNA that some of the others do. Grey sites, marked “N” are places where we know there is DNA, but it’s not clear which base is there. There is only one “N” site in our dataset, so we don’t need to worry about this.

Looking closely at this plot, you can see that there are a lot of sites at which every one of our ten taxa has the same character state. For instance, just before base 500, we can see several uninterrupted vertical bars of color. Since all of the taxa have the same base at these sites, these **invariant sites** can’t give us any useful phylogenetic information. Instead, we need to look at **variable sites**.

paste("Number of sites:", length(attr(Hemoglobin,"index")))

## [1] "Number of sites: 1724"

paste("Number of variable sites:", length(Hemoglobin[[1]]))

## [1] "Number of variable sites: 607"

What does this mean?

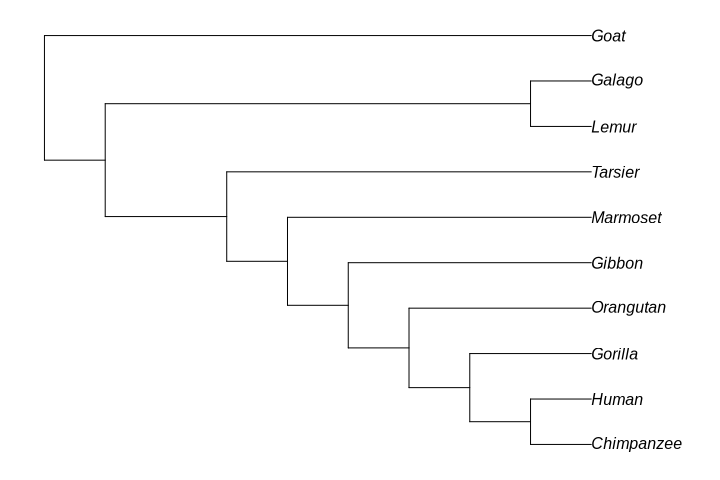
* **the number of sites** is just how many sites are present in our dataset. This is the same as the length of the x-axis in the plot above.
* the number of **variable sites** is the number of sites from the plot above at which at least one taxon has a different base than all of the other taxa.

Pause here and answer Question 15 in your lab worksheet.

## 2) Tree building, plotting, and character mapping

Once again, we’ll use bab to find all the most parsimonious trees. Now, we find only one most parsimonious tree.

Hemoglobin\_tree <- bab(Hemoglobin)  
  
Hemoglobin\_tree <- root(Hemoglobin\_tree, "Goat", resolve.root = TRUE)  
  
par(mai = c(0.2,0.2,0.2,0.2))  
plot(Hemoglobin\_tree)



Epsilon-Hemoglobin tree of primates.

Just as before, we can plot character state changes on the tree to understand the course of character evolution. For instance, for character #160, we can map each time the character state has changed. We can use our program to find the character states of #160 for each of the taxa we study. As before, assume that the character state in the outgroup (here, a goat) is the ancestral state.

matrix(c("a","c","g","t")[sapply(1:10, function(x) Hemoglobin[[x]][160])],  
 dimnames = list(names(Hemoglobin),"character 160"),  
 nrow = 10, ncol = 1, byrow = FALSE  
 )

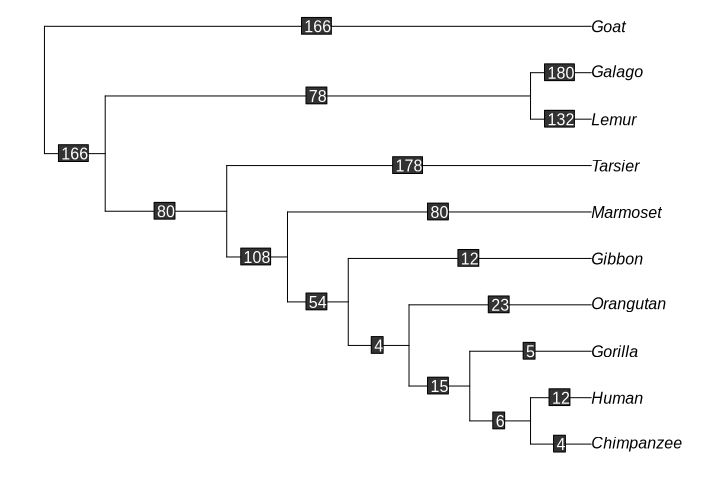
## character 160  
## Galago "t"   
## Lemur "t"   
## Goat "t"   
## Tarsier "g"   
## Marmoset "a"   
## Chimpanzee "g"   
## Gorilla "g"   
## Gibbon "g"   
## Human "t"   
## Orangutan "g"

Just like with our morphological data, we can map every single molecular character onto this tree, and we would find the number of state changes across the tree is equal to the tree’s parsimony score.

Why would we want to do this with molecular data, though? We usually aren’t interested in how individual molecular characters have evolved, but we might be interested in *how much evolution has happened* on a tree. By mapping all of our molecular character state changes onto our tree, we can see how many characters differentiate different taxa, and because we have a large number of characters, this can give us an absolute measure of how different different species are. This measurement is called a **branch length**.

Rather than map all of the characters ourselves, we can do this computationally. The acctran function will count the number of character changes across the phylogeny. In the next tree, we will plot the number of character changes on each branch to indicate the **branch length**.

Hemoglobin\_tree\_branchlengths <- acctran(Hemoglobin\_tree, Hemoglobin)  
par(mai = c(0.2,0.2,0.1,0.2))  
plot(Hemoglobin\_tree\_branchlengths,  
 use.edge.length = FALSE)  
edgelabels(round(Hemoglobin\_tree\_branchlengths[[1]]$edge.length,0),  
 adj = c(0.5, 0.5),  
 frame = "rect",  
 col = "white", bg = "#333333")



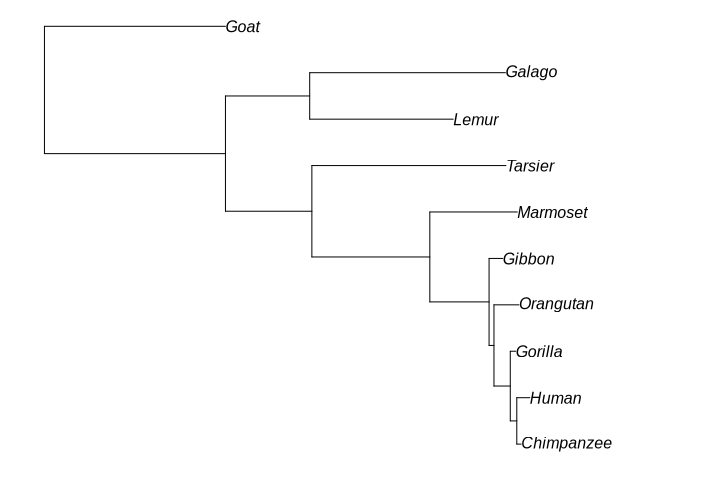
Epsilon hemoglobin tree with numeric branch lengths.

Look at the branch leading to the Gorilla. This branch has a length of 5, meaning that there are 5 characters that changed between the gorilla and the most recent common ancestor of gorillas, humans, and chimpanzees. Likewise, the 6 on the branch leading from that ancestor to the common ancestor of humans and chimpanzees means that there were 6 molecular changes in between these two ancestors. That means that there are a total of 11 (5+6) differences between the gorilla and the most recent common ancestor of humans and chimpanzees.

Use this to answer Question 16 on your worksheet.

Note: often, when you see published phylogenetic trees, the figured branch lengths will all be to scale; so a branch of length 10 will be twice as long on the page as a branch of length 5. The tree above would look like this:

Hemoglobin\_tree\_branchlengths <- acctran(Hemoglobin\_tree, Hemoglobin)  
par(mai = c(0.2,0.2,0.1,0.2))  
plot(Hemoglobin\_tree\_branchlengths,  
 use.edge.length = TRUE)



Epsilon Hemoglobin tree with scaled branch lengths

**Complete all the questions including the synthesis questions on your worksheet as a group. Sign the contribution sheet and turn in your work before you leave the lab.**