Homework 5
Working with sequence data in R

1. Download the *egfr_flank.fasta* file from the Buffalo textbook GitHub repository (chapter 10). https://github.com/vsbuffalo/bds-files/tree/master/

2. Go to GenBank and download some nucleotide files (pick your favorite species and/or gene). For example, you could select a species and search for it in GenBank. Then, pick a gene and download at least five files from separate records. Or you could search for a gene and download files from separate records but different species.
   To download, click "fasta", then (in the top right) "Send to:" -> File -> Create File. I have a short video on Canvas about how I downloaded a sample of the COI gene from a bird called *Diomedea exulans*. You need to pick your own gene/species!

3. GitHub has a dumb feature that it downloads all sequence files using the same file name. Make sure to rename the files so that there are no spaces in the file name (use an underscore or dash instead) and remove any special characters like parentheses. This makes it easier to read into R.

4. Add all of these sequence files to your GitHub repository. It is smart to create new sub-folders for each project. You could, for example, create one for your species of interest, or for this class session. Sync (commit and push) your repository to GitHub. Make sure to push your data at the end of every work/class session.

5. Open R studio. Create a new script in R (File -> new file -> R script). Save your script to your GitHub repository. Load the libraries that we installed last week. Also load (and install if you have to) the "Biostrings" and "seqinr" packages.

```
if (!require("BiocManager", quietly = TRUE))
   install.packages("BiocManager")

BiocManager::install("Biostrings")

install.packages("seqinr")
```

6. The reading this week was the 'msa.pdf' document sections 1-3. Work through section 3 of the document ('msa for the impatient') by typing the commands into R, using your sequence files. Ignore the stuff about LATEX. They run the commands on the example 'system file' that they provide with the package. I want you to import and align the fasta files that you just downloaded from GenBank.

7. To import your sequence files, use the readDNAStringSet() function from Biostrings. To see how to use this function, type a "?" before the R function, e.g.: ?msa or ? readDNAStringSet
   if you want to use the seqinr package, the function is read.fasta()

8. You will need to read in each of the fasta files separately to R, and then combine them into one file using the "c" (combine) command. e.g.,
   sequences <- c(seq_1, seq_2, seq_3)
   Print out your sequences to the screen. Are the sample names very long? If so, change the names to something more logical with this command:
   names(seqs) <- c("new_name_1", "new_name_2")

9. Run ClustalW (in R!) on your combined set of sequences. How would you have run an msa analysis with a different alignment tool (such as MUSCLE)? (hint: check the 'msa' function details)

   I would have ran
   res=msa(dnaSet,method = "MUSCLE")
   print(res)

   *I used then term "dnaSet" to denote the dataset instead of "YourAlignment"

10. Are there any gaps in your alignment? How many? Use:
    print(YourAlignment, show="complete")

88 gaps

11. How long is your alignment? Check the Biostrings package manual for the command to check this.

499 base pairs

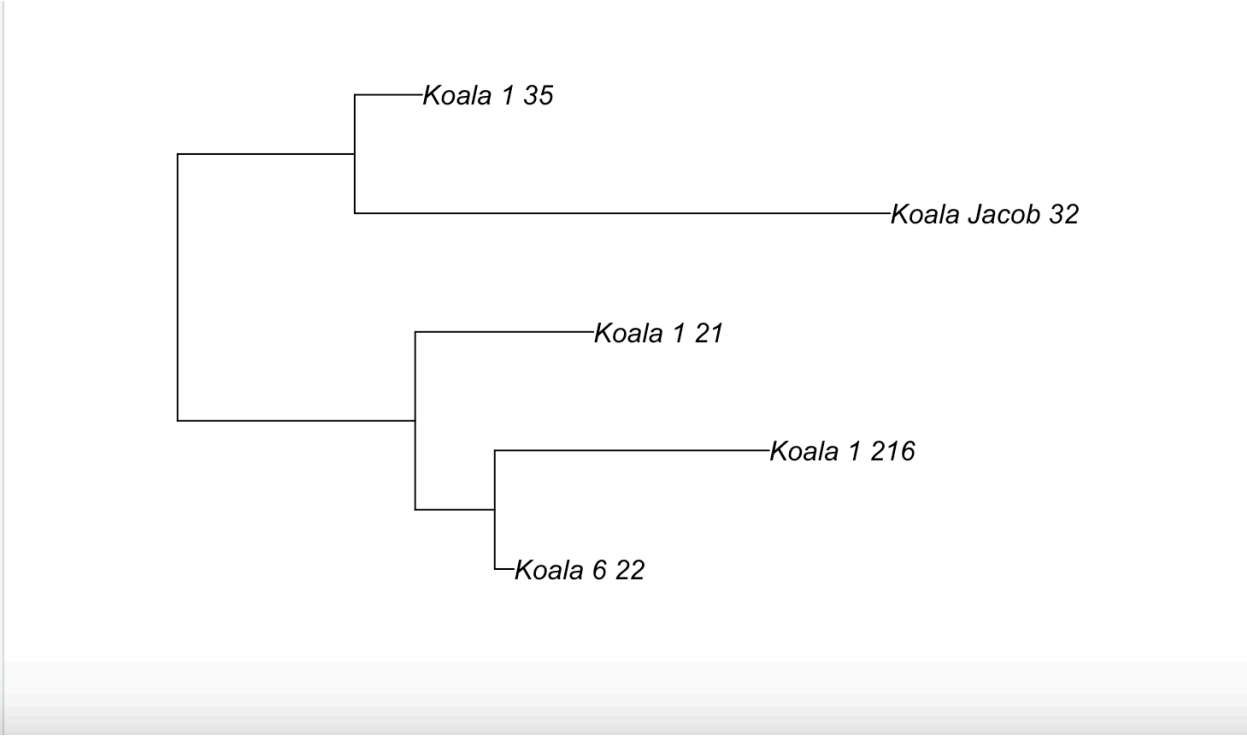
12. Calculate the GC content. Check the Biostrings package manual for the command to "Tabulate the letters" in the alignment. Recall that the GC content is the percentage of the total alignment that is G or C. Alternatively, there is a function in seqinr called GC() but you would need to read in the file using that package.

GC content = 57.2%

13. Now, convert your alignment to the seqinr format (see section 6.3 of the msa.pdf document) and compute a distance matrix.

14. What are the two most distantly related samples in your alignment? What about the two most closely related? Recall that this is what we did on the GenBank website last week in class, but now you've successfully done this as a programmer!

It's a relatively small dataset but the two most closely related are Koala 1 216 and Koala 6 22. The most distantly related groups are between the aforementioned Koalas and the branch containing Koala Jacob 32 and Koala 1 35.



15. See if you can figure out how to translate one of your sequences to amino acids.

translate()
This is through Biostrings.

16. I had planned on asking everyone to figure out on your own how to write your alignment to a file, but it was harder than I expected to figure out how to do this using the

packages that we have already installed. We need to convert our alignment to a format used in the ‘phangorn’ package (so you’ll need to install that package, which is thankfully easy). Use the following commands, modified for your data:

```
Alignment_phyDat <- msaConvert(Alignment, type="phangorn::phyDat")
write.phyDat(Alignment_phyDat, "alignment.fasta", format = "fasta")
```

17. Push your data to GitHub and you’re done!