# Lab 2 R: Assignment

If using your own computer, before class:
Download and install R https://www.r-project.org/ and R-studio: https://www.rstudio.com/
You will also need to install bioconductor, which is easiest to do from inside R-Studio.

1. Start R studio
2. At the prompt, enter, one line at a time:

   ```
   source("https://bioconductor.org/biocLite.R")
   biocLite()
   ```

   Capitals are important, so type exactly.

Download the zip file from the class Google Drive linked to on Canvas.

If using the computer lab computers:
First activate and become acquainted with the UDrive: https://itconnect.uw.edu/wares/online-storage/u-drive-central-file-storage-for-users/
Download the zip file from the class Google Drive linked to on Canvas into your UDrive.

In class:
I will demonstrate the use of R-studio and some of the basics of R. I will also point out some resources in case you want to use more R.

**Assignment is to answer questions in each section in full sentences. More information is at the end of the assignment.**

## Section 0: How to use RStudio

1. Open RStudio
2. Using the … symbol on the far right, navigate to the unzipped alignments_lab folder.
3. Using the gear wheel button, set this as your working directory. You can also do this on the command line by typing:

   ```
   setwd("<<path to this directory>>")
   ```

## Section 1: Playing around with edit distance calculations in R

1. By clinking on them in the directory on the right, open the following programs:

   globalEditdist.R, traceBack.R, and globalAlign.R

   We will look through these before you run them.

2. Run the program globalAlign.R as written by either hitting the source button, or typing: source('globalAlign.R')
3. Matrix a calculates the alignment score. Is it the size you would expect? How large is it for these sequences? How large would it be for sequences of length n and m?
4. Look at matrix d. What does this give us?
5. Does this alignment look like a good one to you?
6. How many potential alignments are there for these sequences?
7. Find scoring values that give you a different final alignment. What are they, and how do they change the alignment?

**Section 2: Aligning hemoglobin RNA sequences**

1. Run the program readin.R.
2. Type View(seqs) and see what this gives us.
3. In the globalEditdist.R, comment out the short sequence input data, and uncomment the lines that read in the table we just generated.
4. Open the chooseAlignSeq.R file. How does it determine which 2 sequences we are aligning?
5. Uncomment the chooseAlignSeq source instruction in globalAlign.R. Run globalAlign.R to compare the human and mouse versions of hemoglobin β.
6. How large is matrix a?
7. What is the alignment score you got?
8. Is the human HBB more similar to the mouse HBB or the human HBA? How did you determine this?
9. Which two human hemoglobin RNAs in this file are most similar to each other (you may work with a partner to get the answer to this)?
10. If you want to look at the alignment, open makeFASTA.R. This will make an alignment out of the 2 sequences that result from the globalAlign.R and traceBack.R. You should be able to edit it to suit your needs. You can view the resulting FASTA file at: https://toolkit.tuebingen.mpg.de/alnviz

## Section 3: Aligning RNA to genome

1. Modify the program readin.R to read in the HBXRNADNA.txt file.
2. Run globalAlign.R on these sequences. Describe your results.
3. Use the makeFASTA.R to create a fasta file for this alignment. Using an alignment viewer (like the one above), look at the alignment. Given that this is a genome to transcript alignment, what would you expect to find? Is that what you found?
4. How would you make this a better alignment?
5. Using globalEditdist.R as a template, write a program that will generate a matrix of a local alignment of these sequences. (submit with your report)
6. EXTRA CREDIT: write a trace back program that will give you an alignment of these two sequences based on the local alignments.
7. How many exons would you expect in this alignment? Is that approximately what you found?

## Section 4: Using Bioconductor to do local alignment and to perform multiple sequence alignment

As you can see, the Needleman-Wunsch algorithm we used for global alignment, ans the Smith-Waterman we used for local are accurate, and are guaranteed to find the optimal alignment, however they are very slow. Also, I am not an excellent software engineer, so these programs are not optimally designed to minimized calls to memory, increasing their time. Therefore, we are going to look at a few programs that are built into some of the R libraries in Bioconductor. In particular, we'll be using Biostrings and DECIPHER[1,2].

To load Biostrings and DECIPHER do the following:

1. biocLite('Biostrings')
2. biocLite('DECIPHER')
3. library(Biostrings)

4. library(DECIPHER)

Perform global alignment of human HBB RNA to DNA.

Since you have already read in the text file, we'll use those sequences to perform the alignment. To perform a global alignment type the following:

1. galign<- pairwiseAlignment(pattern=c(seqs[1,2]), subject=seqs[2,2])

   This calls the Biostrings function pairwiseAlignment. It will align a vector of strings in the pattern array to the subject string. It sets the result equal to galign, so we can use this again. Let's see how this did for an alignment. To do that we need to turn this into a FASTA file. Follow the following steps to do that.

   ```
   r = BStringSet( c( toString( subject(galign) ), toString(pattern(galign))))
   writeXStringSet(r, file= "global.fasta")
   ```

2. Look at this using the same tool as above. Is this better than our previous alignment? What criteria are you using to determine that? This algorithm differs slightly from the one in out global alignment program. This has an additional penalty. It has both a gap opening penalty and a gap extending penalty. How does this affect the alignment? You can change these penalties with the following commands.

   ```
   pairwiseAlignment(pattern = c(seqs[1,2]), subject = seqs[2,2],
    gapOpening = 0, gapExtension = 1)
   ```

3. We can switch this to a local alignment by adjusting the input as follows:

   ```
   lalign<- pairwiseAlignment(pattern = c(seqs[1,2]), subject = seqs[2,2], type = "local")
   ```

   How does this change the alignment? What do you get in this alignment? Is this better than the local alignment? Why? What criteria are you using to make this judgement?

Let's try a multiple sequence alignment. This is most often done by multiple pairwise alignments. As this is even more computationally expensive, heuristics are often used. As the alignment is often somewhere near the diagonal, one way to decrease the computational expense is to decrease the search space by limiting the distance from the diagonal that one computes (Fig 1.)[3]. The DECIPHER package uses this heuristic to shorten the search.

We are going to start by looking hemoglobin β molecules from many different organisms.

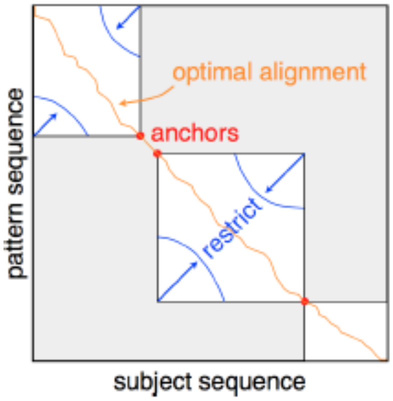

**Figure 1: Limiting search to sequence space near the diagonal can decrease computational time. Adapted from Wright, 2014.**

4. In the folder there is a file name HBBs.fasta. Read this fine in using the following command.
   HBB <- 'HBBs.fasta'

5. To parse this into the data type that we can use for alignments, use the following command.
   dnaHBB<- readDNAStringSet(HBB)

6. Now we can align these. There are two options. You can align the translated protein sequences, or align the DNA sequences. For now we'll align the DNA, though it's slower. It will give us a more accurate sense of the evolution of these RNAs. To start the alignment, use the following commands.
   HBBalign<-AlignSeqs(dnaHBB)
   If you wanted to align by protein sequences, you would use AlignTranslation().

7. To view the alignment in a web browser, use the following command.
   BrowseSeqs(HBBalign)
   How good is the alignment? What can you learn about evolution from these hemoglobins?

8. Repeat this for the file HBX.fasta, using HBX in place of HBB, which has different types of hemoglobin from may different organisms.

9. Let's make a phylogenetic tree for these RNAs. First we'll need a distance matrix, which calculates how different each sequence is from each other sequence. We can do that with the following command.
   dm<-DistanceMatrix(dnaHBX)

10. We can use this distance matrix to make a dendrogram of these sequences, indicating which are more closely related, evolutionarily. You can export the dendrogram from R-Studio to either an image or a pdf.
    IdClusters(myDistMatrix=dm, myXStringSet = dnaHBX, method='NJ', cutoff=-Inf, showPlot=TRUE, type='dendrogram')

11. What does this tell you about evolution? Are the types of hemoglobin from the same species more similar to each other, or to the same type of hemoglobin in different organisms? What could you do with these alignments next?

## Section 5: BLAST

1. Using Blast[4], identify the 4 most closely related sequences to the *Bos Taurus* γ–hemoglobin. Sequence is below. The RNA name is NM_001014902.3, you can also use it in BLAST.

```
ATGCTGAGCGCTGAGGAGAAGGCTGCCGTCACCTCCCTATTTGCCAAGGTGAAAGTGGATGAAGTTGGTG
GTGAGGCCCTGGGCAGGCTGCTGGTTGTCTACCCCTGGACTCAGAGGTTCTTTGAGTCCTTTGGGGACTT
GTCCTCTGCCGATGCCATTTTGGGAAACCCTAAGGTGAAGGCCCATGGCAAGAAGGTGCTGGACTCCTTC
TGTGAGGGCCTGAAGCAACTTGATGACCTCAAGGGTGCCTTTGCTTCGCTGAGTGAGCTGCACTGTGATA
AGCTGCACGTGGATCCTGAGAACTTCAGGCTCCTGGGCAACGTGCTGGTGGTTGTGCTGGCTCGCCGCTT
TGGCAGTGAATTCTCCCCGGAGCTGCAGGCTAGCTTTCAGAAGGTGGTGACTGGTGTGGCCAATGCCCTG
GCCCACAGATATCACTAA
```

2. Using the human HBB gene (not mRNA) and HBA1 gene (sequences available from NCBI), identify the other human hemoglobins. You will likely need to change to more dissimilar or somewhat similar sequences in BLAST. You may also want to consider which set of DNAs you are blasting against. Download these sequences.
3. Align these sequences using Clustal Omega[5] on the web or using the R programs we just used.
4. Using these results, generate a phylogenetic tree that demonstrates the evolution of our multiple hemoglobin genes.

## References

1. Wright, E. S. Using DECIPHER v2.0 to Analyze Big Biological Sequence Data in R. *R J.* **8,** 352–359 (2016).
2. Pages H, Aboyoun P, Gentleman R, DebRoy, S. Biostrings: String objects representing biological sequences, and matching algorithms. *R Packag. version 2.40.2* (2016). doi:R package version 2.26.3
3. Wright, E. S. The Art of Multiple Sequence Alignment in R. 1–7 (2014).
4. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215,** 403–410 (1990).
5. Sievers, F. *et al.* Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.* **7,** 539–539 (2014).

**Instructions for lab report:**

There should be 5 sections, as in the lab. As before, these should be written in paragraph form, not as bullet points. This is worth 90 points. It should be no more than 7 pages.

Section 1:

Answer the questions.

Section 2:

Answer the question in steps 6-10. Show a diagram of at least a portion of the alignment. Be sure to label this figure with an appropriate caption.

Section 3:

Provide descriptions and answer questions indicated. Be sure to include your modified alignment program. Provide images if they are useful for understanding your results.

Section 4:

Answer questions. Provide visual representations of your results that allow the reader to understand the differences you found.

Section 5:

Describe your results and answer the questions. Also provide diagrams as appropriate.