Programming for Biologists (MMG1002H)

Assignment 4
10 Points

Assignment goals:
- Practice R concepts learned in class
- Practice regular expressions

Part I: Practice regular expressions (4 marks)
Write regular expressions for each of the following cases. Assume that the regular expression would be used in the following context, where we'd like to check if a string has a particular pattern, and if so, execute a set of statements:

```
if (grepl("regular expression", stringVariable)) {
        <do something here with a set of statements>
}
```

a. Check to see if a string contains a valid yeast gene name, e.g. YJL045W, YKR034W, YLL023C (hint: a yeast systematic gene name consists of a Y, a letter between A and P indicating chromosome number, an R or L for right or left chromosome arm, 3 numbers indicating chromosome position, and then a W or C to designate the strand, either Watson or Crick)

b. Check to see if a yeast gene is encoded on the Crick strand (Hint: YHR143W-C is not encoded on the Crick strand)

c. Check a sequence string for Mot3's (a yeast transcription factor) binding site motif: AAGG(G or T)T

d. Check a sequence string for Hap1's (a yeast transcription factor) binding site motif. '?' implies that there is a single arbitrary nucleotide.
GG???TA?CGG

e. Explain in English what the following regular expression matches:
CGG(A|C|G|T){3}T(A|C|G|T)(A|G)(A|C|G|T){8,12}CCG

E.g. "It matches CGG followed by…"

Write your answers in a text file called assignment4_part1.txt (make sure this is a regular text file, not a rich text, Word or other formatted file). Each regular expression should be in a line by itself. E.g.
a)
regular expression a here
b)
regular expression b here…

Note: You do not need to write a script for this section, though you can test your regular expression in your own script that you don't submit.

Part II: Finding transcription factor binding sites in yeast DNA sequence (6 marks)

Gene transcription in eukaryotes is controlled by transcription factor proteins, which bind specific sequence motifs near the gene they regulate. In yeast, the transcription factors are well-characterized, such that we often know the specific sequence motif bound by each factor.

We would like to write a script to answer the following question:

How often does each transcription factor motif occur in gene promoter sequences across the whole genome? (you'll need to print a list of transcription factors and a corresponding count of the number of genes whose promoter contains that motif for each transcription factor)

Here are the data that we have:
(1) A tab-delimited list of a subset of yeast transcription factors and their corresponding sequence binding sites (motifs). Note: there are multiple binding sites for some transcription factors—you can count frequencies independently for each of these binding sites. yeast_tf_motifs.txt
(2) A FASTA file containing the DNA sequence of all yeast genes including 1000bp of upstream and downstream sequence originally downloaded from https://www.yeastgenome.org/seqTools orf_genomic_1000_554.fasta
Note: this contains only the first 554 sequences to keep the file smaller for the purposes of the assignment.

Write R code that reads the FASTA file and converts it to a data frame where the ORF name is stored in column 1 and the corresponding upstream and downstream sequence is stored in column 2. To support learning, you are not allowed to use an R package to read FASTA files, though such a package would be useful if solving this problem in your research. Once you create your data frame, you should be able to find the upstream and downstream sequence using code like:

```
sequenceDatabase$sequences[sequenceDatabase$sequenceNames=="YAL002W"]
```

This would print the upstream 1000bp and downstream 1000bp DNA sequence, concatenated, for gene YAL002W. Note: the concatenated sequence is present in the input file.

Your finished script should output the fraction of the number of genes with at least one occurrence of each motif in its promoter sequence (defined as the 1000 bases upstream of the start site) over the total number of genes.

Submitting your assignment

When you're finished with the assignment, make a report of any questions you answered plus any requested output, and gather the scripts that you modified. Place them all in a folder, and create a single archive file called assignment4.tar.gz. Remember, you'll need to use a command like this:

```
tar -czf assignment4.tar.gz <your name>-Assignment4
```

When you're finished, upload the `assignment4.tar.gz` file to the course website. You only need to upload your single `.tar.gz` file that contains all your answers and output for this assignment.

You're finished with Assignment 4!