Homework 3

Due Date: Friday, Sep 27, 2024 before class

Total Points: 130

Problem 1

(a) Reverse and complement of DNA (20 points)

Write a `RevComp` function that returns the reverse and complement of a DNA sequence string. Include an argument that will allow to return only (i) the reversed sequence, (ii) the complemented sequence, or (iii) the reversed and complemented sequence. The following R functions will be useful for the implementation: Generate a short test DNA sequence

```
x <- c("ATGCATTGGACGTTAG")
x
```

```
## [1] "ATGCATTGGACGTTAG"
```

Vectorize sequence

```
x <- substring(x, 1:nchar(x), 1:nchar(x))
x
```

```
##  [1] "A" "T" "G" "C" "A" "T" "T" "G" "G" "A" "C" "G" "T" "T" "A" "G"
```

Reverse sequence

```
x <- rev(x)
x
```

```
##  [1] "G" "A" "T" "T" "G" "C" "A" "G" "G" "T" "T" "A" "C" "G" "T" "A"
```

Collapse sequence back to character string

```
x <- paste(x, collapse="")
x
```

```
## [1] "GATTGCAGGTTACGTA"
```

Form complement of sequence

```
chartr("ATGC", "TACG", x)
```

```
## [1] "CTAACGTCCAATGCAT"
```

(b) Write an export function (20 points)

Write a function that applies the `RevComp` function to many sequences stored in a vector. In addition, write an export function that saves the sequences generated under in 1(b) to a txt file.

Problem 2 Perform the following steps in R:

(a)

Simulate a string of 10,000 characters drawn uniformly and independently from the set {A, C, G, T} [Hint: sample] (7 points)

(b)

Create a frequency table of the string [Hint: table] (3 points)

(c)

Write a function to create a contingency table of adjacent k-tuples. For example, with k=3 and with the string "CAGACAAAAC", you would want to produce the following table: [Only use for loops and paste(, collapse=" "), Do not use embed, substr or do.call] (30 points)

AAA	AAC	ACA	AGA	CAA	CAG	GAC
2	1	1	1	1	1	1

Problem 3 Write your own factorial function

x! $= 1 \times 2 \times 3 \ldots \times$x; 0!=1. x is an integer ≥ 0. Write your own function to perform the calculation. (20 points) [Do not use the function prod and factorial in R]

Problem 4 Translate DNA into Protein

Write a function that will translate one or many DNA sequences in all three reading frames into proteins. (30 points) The following commands will simplify this task:

Import lookup table of genetic code

```
AAdf <- read.delim2(file="https://people.stat.sc.edu/hoyen/STAT718/Homework/AA.txt", header=TRUE)
AAdf[1:4,]
```

```
##   Codon AA_1 AA_3 AA_Full AntiCodon
## 1   TCA    S  Ser  Serine       TGA
## 2   TCG    S  Ser  Serine       CGA
## 3   TCC    S  Ser  Serine       GGA
## 4   TCT    S  Ser  Serine       AGA
```

Generated named vector of relevant components

```
AAv <- as.character(AAdf[,2])
names(AAv) <- AAdf[,1]
AAv
```

```
## TCA TCG TCC TCT TTT TTC TTA TTG TAT TAC TAA TAG TGT TGC TGA TGG CTA CTG CTC CTT CCA CCG CCC CCT CAT
## "S" "S" "S" "S" "F" "F" "L" "L" "Y" "Y" "*" "*" "C" "C" "*" "W" "L" "L" "L" "L" "P" "P" "P" "P" "H"
## CAC CAA CAG CGA CGG CGC CGT ATT ATC ATA ATG ACA ACG ACC ACT AAT AAC AAA AAG AGT AGC AGA AGG GTA GTG
## "H" "Q" "Q" "R" "R" "R" "R" "I" "I" "I" "M" "T" "T" "T" "T" "N" "N" "K" "K" "S" "S" "R" "R" "V" "V"
## GTC GTT GCA GCG GCC GCT GAT GAC GAA GAG GGA GGG GGC GGT
## "V" "V" "A" "A" "A" "A" "D" "D" "E" "E" "G" "G" "G" "G"
```

Tripletize sequence and translate by name subsetting/sorting of AAv

```r
y <- gsub("(...)", "\\1_", x)
y <- unlist(strsplit(y, "_"))
y <- y[grep("^...$", y)]
AAv[y]
```

```
## GAT TGC AGG TTA CGT
## "D" "C" "R" "L" "R"
```