Homework Assignment

Pair-wise Sequence Alignment

03-04-2023

Task A

```
# Accession numbers
ac_1 <- "AY884001"
ac_2 <- "MH940245"

# Retrieve the data
q1 <- query("q1", paste("AC=", ac_1))
q2 <- query("q2", paste("AC=", ac_2))

# Get the sequences
seq1 <- getSequence(q1$req[[1]])
seq2 <- getSequence(q2$req[[1]])

# Print the sequences
print(DNAString(c2s(seq1)))
```

```
## 29815-letter DNAString object
## seq: GAGCGATTGACGTTCGTACCGTCTATCAGCTTACGA...TGATTGAAATTAATTATAGCCTTTTGGAGGAATTAC
```

```
print(DNAString(c2s(seq2)))
```

```
## 29811-letter DNAString object
## seq: GATTGACGTTCGTACCGTCTATCAGCTTACGATCTC...TGATTGAAATTAATTATAGCCTTTTGGAGGAATTAC
```

Task B

```
p1 <- seqinr::translate(seq1)
p2 <- seqinr::translate(seq2)

seq1_count <- table(p1)
seq2_count <- table(p2)
print(seq1_count)
```

```
## p1
##    *    A    C    D    E    F    G    H    I    K    L    M    N    P    Q    R
## 1006  310  729  268  196  858  343  141  588  356 1037  172  377  176  152  442
##    S    T    V    W    Y
##  779  414  612  251  731
```

```
print(seq2_count)
```

```
## p2
```

```
##    *    A    C    D    E    F    G    H    I    K    L    M    N    P    Q    R
##  388  395  370  413  250  678  406  185  787  472 1287  301  490  240  269  274
##    S    T    V    W    Y
##  697  429  968  157  481
```

```
seq1_prop <- proportions(seq1_count)
seq2_prop <- proportions(seq2_count)
print(seq1_prop)
```

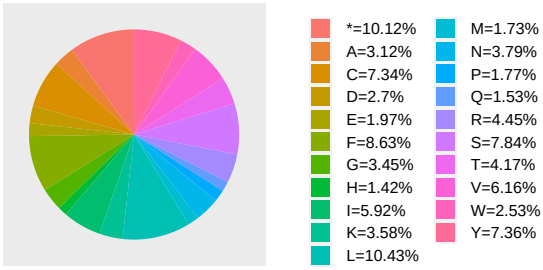
```
## p1
##          *          A          C          D          E          F          G
## 0.10122761 0.03119340 0.07335480 0.02696720 0.01972228 0.08633528 0.03451399
##          H          I          K          L          M          N          P
## 0.01418797 0.05916683 0.03582210 0.10434695 0.01730731 0.03793520 0.01770980
##          Q          R          S          T          V          W          Y
## 0.01529483 0.04447575 0.07838599 0.04165828 0.06158181 0.02525659 0.07355605
```

```
print(seq2_prop)
```

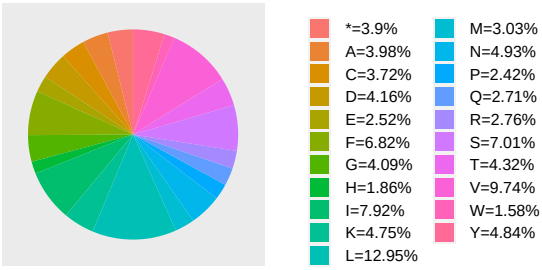
```
## p2
##          *          A          C          D          E          F          G
## 0.03904599 0.03975043 0.03723458 0.04156184 0.02515850 0.06822985 0.04085740
##          H          I          K          L          M          N          P
## 0.01861729 0.07919895 0.04749925 0.12951595 0.03029083 0.04931066 0.02415216
##          Q          R          S          T          V          W          Y
## 0.02707054 0.02757371 0.07014189 0.04317198 0.09741371 0.01579954 0.04840495
```

```
theme_update(legend.key.size = unit(3, "mm"), legend.text = element_text(size = 6),
    axis.text = element_blank(), axis.ticks = element_blank(),
    panel.grid = element_blank())
df1 <- data.frame(slices = seq1_prop, amino_acids = paste(names(seq1_prop),
    "=", round(seq1_prop * 100, 2), "%", sep = ""))
df2 <- data.frame(slices = seq2_prop, amino_acids = paste(names(seq2_prop),
    "=", round(seq2_prop * 100, 2), "%", sep = ""))
plot1 <- ggplot(df1, aes(x = "", y = slices.Freq, fill = amino_acids)) +
    geom_bar(stat = "identity", width = 1) + coord_polar("y",
    start = 0) + ggtitle(ac_1) + xlab("") + ylab("") + guides(fill = guide_legend(title = NULL))
plot2 <- ggplot(df2, aes(x = "", y = slices.Freq, fill = amino_acids)) +
    geom_bar(stat = "identity", width = 1) + coord_polar("y",
    start = 0) + ggtitle(ac_2) + xlab("") + ylab("") + guides(fill = guide_legend(title = NULL))
grid.arrange(plot1, plot2, ncol = 2)
```

AY884001



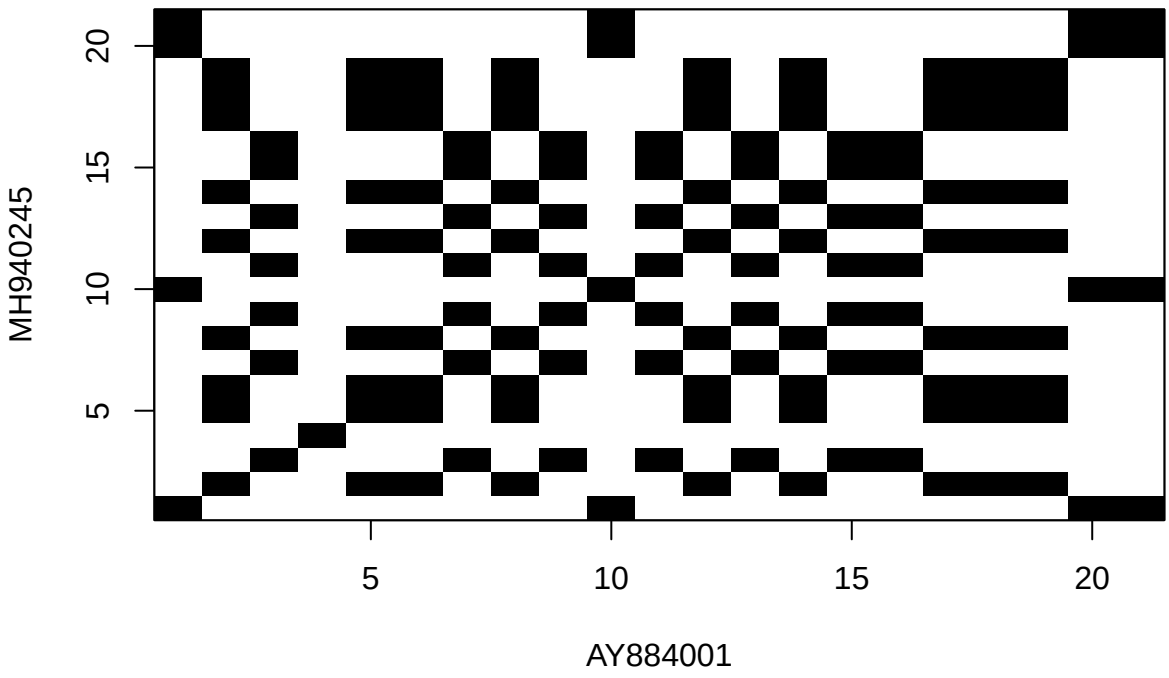
MH940245



Task C

```
seq1_orfs <- findORFs(c2s(seq1))
seq2_orfs <- findORFs(c2s(seq2))

dotPlot(s2c(seq1_orfs[1, "orf.sequence"]), s2c(seq2_orfs[1, "orf.sequence"]),
    xlab = ac_1, ylab = ac_2)
```



The symmetric nature of the plot reveals that the two sequences are almost identical. In fact, by doing a direct comparison we can see that they are exactly the same!

```
print(seq1_orfs[1, "orf.sequence"] == seq2_orfs[1, "orf.sequence"])
```

```
## orf.sequence
##         TRUE
```

Task D

```
opt_ga <- pairwiseAlignment(DNAString(c2s(seq1)), DNAString(c2s(seq2)),
    type = "global", substitutionMatrix = nucleotideSubstitutionMatrix(match = 2,
        mismatch = -1, baseOnly = TRUE), gapOpening = 0, gapExtension = -2)
print(opt_ga)
```

```
## Global PairwiseAlignmentsSingleSubject (1 of 1)
## pattern: GAGCGATTGACGTTCGTACCGTCTATCAGCTTAC...ATTGAAATTAATTATAGCCTTTTGGAGGAATTAC
## subject: GA----TTGACGTTCGTACCGTCTATCAGCTTAC...ATTGAAATTAATTATAGCCTTTTGGAGGAATTAC
## score: 59611
```

```
cat("Alignment for first 20 nucleotides:", paste(substr(pattern(opt_ga),
    1, 20), substr(subject(opt_ga), 1, 20), sep = "\n"), sep = "\n")
```

```
## Alignment for first 20 nucleotides:
## GAGCGATTGACGTTCGTACC
## GA----TTGACGTTCGTACC
```

Task E

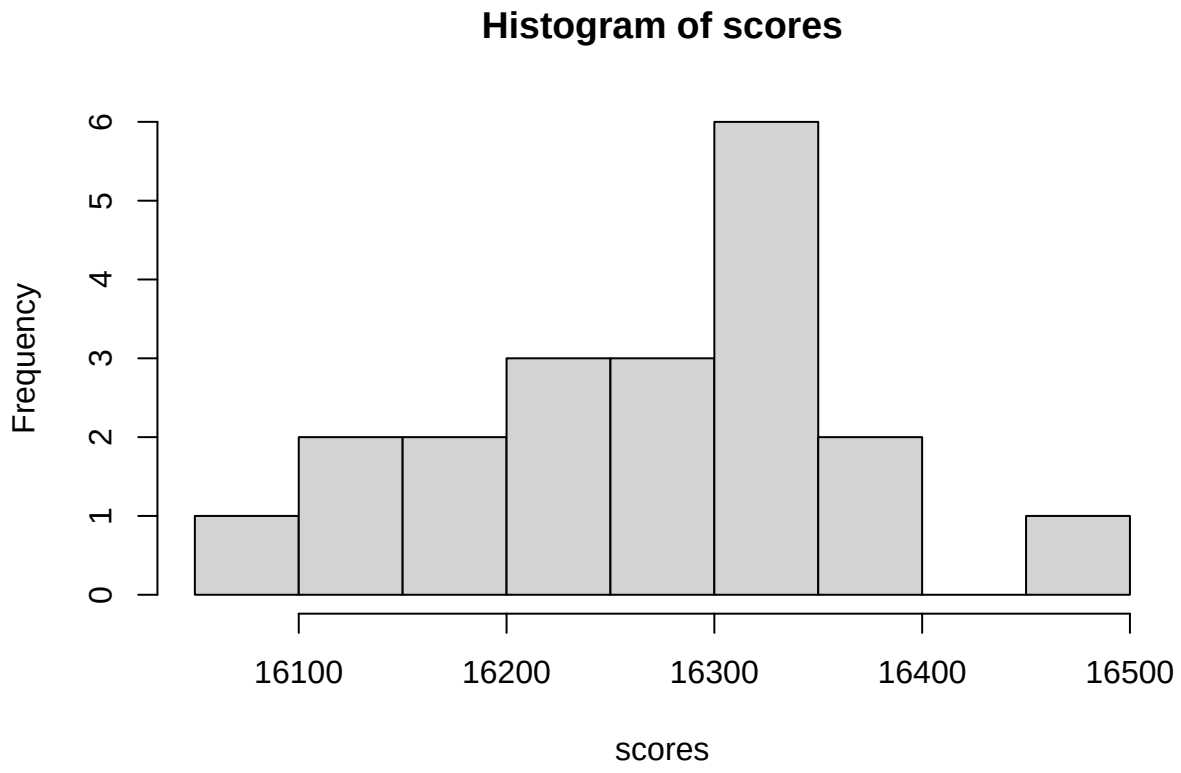
```
seq1_random <- generateSeqsWithMultinomialModel(c2s(seq1), 20)
seq2_random <- generateSeqsWithMultinomialModel(c2s(seq2), 20)

scores <- c()
for (i in 1:20) {
    score <- pairwiseAlignment(DNAString(c2s(seq1_random[[i]])),
        DNAString(c2s(seq2_random[[i]])), type = "global", substitutionMatrix = nucleotideSubstitutionMa
            mismatch = -1, baseOnly = TRUE), gapOpening = 0,
        gapExtension = -2, scoreOnly = TRUE)
    scores <- append(scores, score)
}

print(scores)
```

```
##  [1] 16291 16333 16072 16315 16336 16201 16312 16162 16252 16390 16342 16210
## [13] 16219 16150 16390 16108 16189 16453 16324 16258
```

```
hist(scores)
```




```
p_value <- sum(scores > score(opt_ga))/length(scores)
print(p_value)
```

```
## [1] 0
```

The global alignment is statistically significant as the p-value is below 0.05. This means that the odds of this alignment happening by chance are extremely low.

Task F

```
opt_la <- pairwiseAlignment(DNAString(c2s(seq1)), DNAString(c2s(seq2)),
    type = "local", substitutionMatrix = nucleotideSubstitutionMatrix(match = 3,
        mismatch = -2, baseOnly = TRUE), gapOpening = -4, gapExtension = -2)
print(opt_la)
```

```
## Local PairwiseAlignmentsSingleSubject (1 of 1)
## pattern: [5] GATTGACGTTCGTACCGTCTATCAGCTTACGA...TGAAATTAATTATAGCCTTTTGGAGGAATTAC
## subject: [1] GATTGACGTTCGTACCGTCTATCAGCTTACGA...TGAAATTAATTATAGCCTTTTGGAGGAATTAC
## score: 89428
```

```
print("Length of the alignment:")
```

```
## [1] "Length of the alignment:"
```

```
print(nchar(pattern(opt_la)))
```

```
## [1] 29811
```