# Homework Assignment

Protein Sequence Alignment

03-17-2023

## Task A

```
choosebank("swissprot")
ac_1 <- "A0A1I9G1P5"
ac_2 <- "A0A1S0U2K5"

q1 <- query("q1", query = paste("AC=", ac_1, sep = ""))
q2 <- query("q2", query = paste("AC=", ac_2, sep = ""))

seq1 <- getSequence(q1$req[[1]])
seq2 <- getSequence(q2$req[[1]])
```

```
print(AAString(c2s(seq1)))
```

```
## 208-letter AAString object
## seq: MKLIVDSGHTGVNQLGGVFVNGRPLPDSTRQKIVDL...LAQLPPLTQANHLNKKDSGKLLWYSNYKGGWKELLI
```

```
print(AAString(c2s(seq2)))
```

```
## 133-letter AAString object
## seq: MLGDKKYSGHTGVNQLGGVFVNGRPLPDSTRQKIVD...EQPSIFAWEIRDKLLHEKVCSPDTIPSIHVGFSIHF
```

## Task B

```
count_seq1 <- table(seq1)
count_seq2 <- table(seq2)

print(count_seq1)
```

```
## seq1
##  A  C  D  E  F  G  H  I  K  L  M  N  P  Q  R  S  T  V  W  Y
## 12  5 10  7  3 15  4 13 15 19  3 10 12 12 14 19  7 15  5  8
```

```
print(count_seq2)
```

```
## seq2
##  A  C  D  E  F  G  H  I  K  L  M  N  P  Q  R  S  T  V  W  Y
##  5  5  7  5  4 12  5 12  9  8  1  3  8  5 10 13  5 11  1  4
```

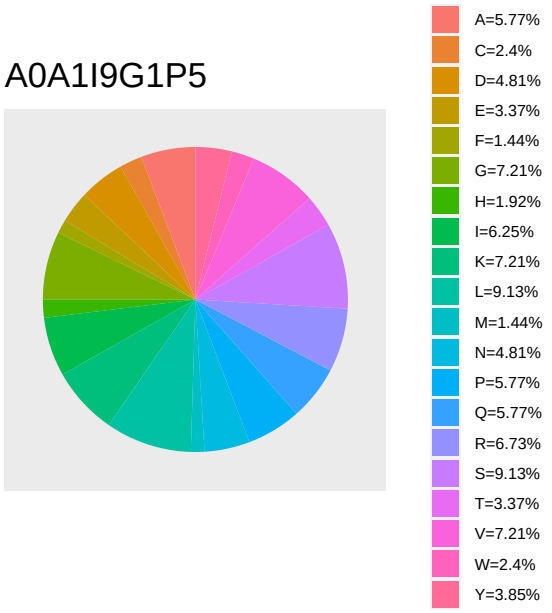
```
prop_seq1 <- proportions(count_seq1)
prop_seq2 <- proportions(count_seq2)

print(prop_seq1)
```

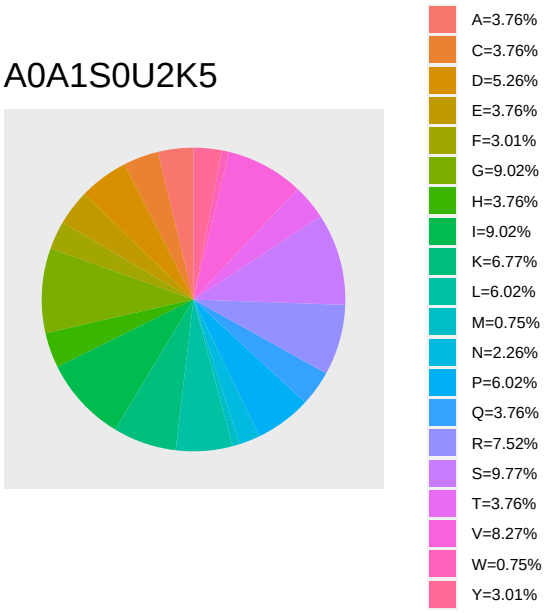
```
## seq1
##          A          C          D          E          F          G          H
## 0.05769231 0.02403846 0.04807692 0.03365385 0.01442308 0.07211538 0.01923077
##          I          K          L          M          N          P          Q
## 0.06250000 0.07211538 0.09134615 0.01442308 0.04807692 0.05769231 0.05769231
##          R          S          T          V          W          Y
## 0.06730769 0.09134615 0.03365385 0.07211538 0.02403846 0.03846154
```

```
print(prop_seq2)
```

```
## seq2
##           A           C           D           E           F           G
## 0.037593985 0.037593985 0.052631579 0.037593985 0.030075188 0.090225564
##           H           I           K           L           M           N
## 0.037593985 0.090225564 0.067669173 0.060150376 0.007518797 0.022556391
##           P           Q           R           S           T           V
## 0.060150376 0.037593985 0.075187970 0.097744361 0.037593985 0.082706767
##           W           Y
## 0.007518797 0.030075188
```

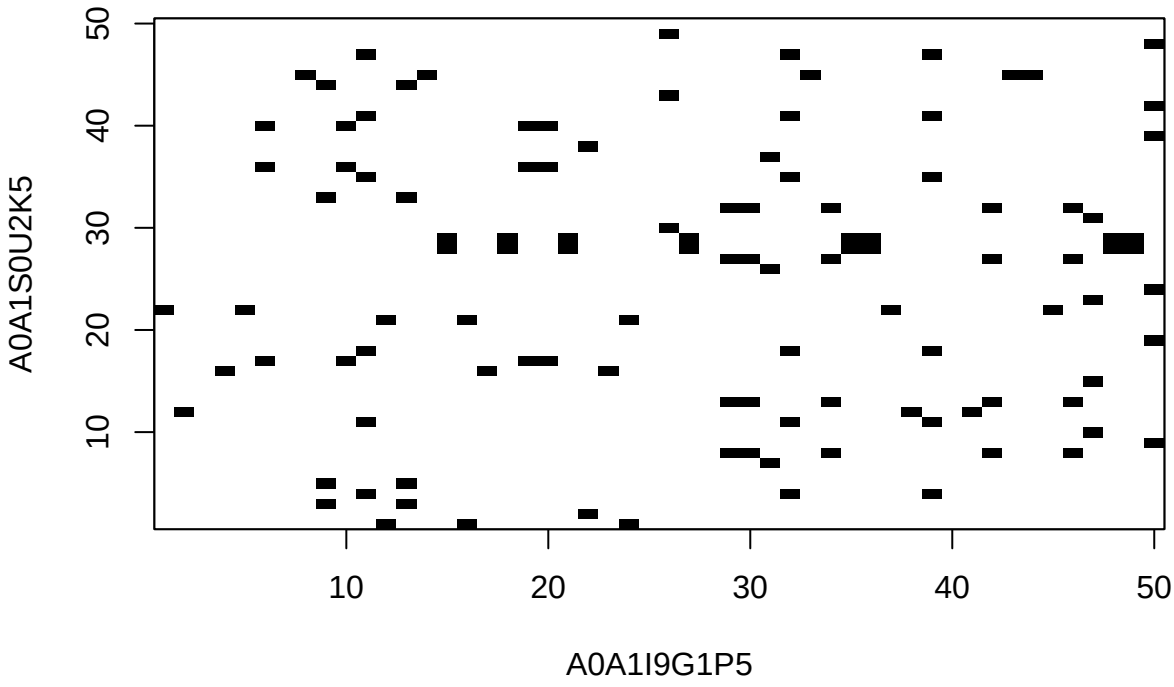
```
theme_update(legend.key.size = unit(4, "mm"), legend.text = element_text(size = 6),
    axis.text = element_blank(), axis.ticks = element_blank(),
    panel.grid = element_blank())
df1 <- data.frame(slices = prop_seq1, amino_acids = paste(names(prop_seq1),
    "=", round(prop_seq1 * 100, 2), "%", sep = ""))
df2 <- data.frame(slices = prop_seq2, amino_acids = paste(names(prop_seq2),
    "=", round(prop_seq2 * 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)
```




## Task C

```
dotPlot(tail(seq1, 50), tail(seq2, 50), xlab = ac_1, ylab = ac_2)
```




There is no similarity between the last 50 characters of the two sequences, evidenced by the lack of a diagonal line on the graph.

## Task D

```
printPairwiseAlignment <- function(alignment, chunksize = 60,
    returnlist = FALSE) {
    require(Biostrings)  # This function requires the Biostrings package
    seq1aln <- pattern(alignment)  # Get the alignment for the first sequence
    seq2aln <- subject(alignment)  # Get the alignment for the second sequence
    alnlen <- nchar(seq1aln)  # Find the number of columns in the alignment
    starts <- seq(1, alnlen, by = chunksize)
    n <- length(starts)
    seq1alnresidues <- 0
    seq2alnresidues <- 0
    for (i in 1:n) {
        chunkseq1aln <- substring(seq1aln, starts[i], starts[i] +
            chunksize - 1)
        chunkseq2aln <- substring(seq2aln, starts[i], starts[i] +
            chunksize - 1)
        # Find out how many gaps there are in chunkseq1aln:
        gaps1 <- countPattern("-", chunkseq1aln)  # countPattern() is from Biostrings package
        # Find out how many gaps there are in chunkseq2aln:
        gaps2 <- countPattern("-", chunkseq2aln)  # countPattern() is from Biostrings package
        # Calculate how many residues of the first sequence
```

```
        # we have printed so far in the alignment:
        seq1alnresidues <- seq1alnresidues + chunksize - gaps1
        # Calculate how many residues of the second
        # sequence we have printed so far in the alignment:
        seq2alnresidues <- seq2alnresidues + chunksize - gaps2
        if (returnlist == "FALSE") {
            cat(paste0("\\textcolor{blue}{", chunkseq1aln, " ",
                seq1alnresidues, "}", "\\newline"))
            cat(paste0("\\textcolor{red}{", chunkseq2aln, " ",
                seq2alnresidues, "}", "\\newline"))
            cat(paste0("  \\newline"))
            # print(paste(chunkseq1aln, seq1alnresidues))
            # print(paste(chunkseq2aln, seq2alnresidues))
            # print(paste(' '))
        }
    }
    if (returnlist == "TRUE") {
        vector1 <- s2c(substring(seq1aln, 1, nchar(seq1aln)))
        vector2 <- s2c(substring(seq2aln, 1, nchar(seq2aln)))
        mylist <- list(vector1, vector2)
        return(mylist)
    }
}
```

```
ga <- pairwiseAlignment(AAString(c2s(seq1)), AAString(c2s(seq2)),
    substitutionMatrix = PAM250, type = "global", gapOpening = -10,
    gapExtension = -0.5)
print(ga)
```

```
## Global PairwiseAlignmentsSingleSubject (1 of 1)
## pattern: MKLIVD---SGHTGVNQLGGVFVNGRPLPDSTRQ...QLPPLTQANHLNKKDSGKLLWYSNYKGGWKELLI
## subject: M--LGDKKYSGHTGVNQLGGVFVNGRPLPDSTRQ...----------------------------------
## score: 534.5
```

```
printPairwiseAlignment(ga, chunksize = 30)
```

MKLIVD—SGHTGVNQLGGVFVNGRPLPD 27
M–LGDKKYSGHTGVNQLGGVFVNGRPLPD 28

STRQKIVDLAHQGARPCDISRILQVSNGCV 57
STRQKIVDLAHQGARPCDISRILQVSNGCV 58

SKILCRYYESGTIRPRAIGGSKPRVATVSV 87
SKILCRYYESGTIRPRAIGGSKPRVATVSV 88

CDKIESYKREQPSIFAWEIRDKLLHEKVCS 117
CDKIESYKREQPSIFAWEIRDKLLHEKVCS 118

PDTIPSVSSINRVLRNLAAKKEQQAMQNDF 147
PDTIPSI——————-HVGF 129

YDRALRY 177
—SIHF 156

## Task E

```
la <- pairwiseAlignment(AAString(c2s(seq1)), AAString(c2s(seq2)),
    substitutionMatrix = BLOSUM62, type = "local", gapOpening = -10,
    gapExtension = -0.5)
print(la)
```

```
## Local PairwiseAlignmentsSingleSubject (1 of 1)
## pattern: [7] SGHTGVNQLGGVFVNGRPLPDSTRQKIVDLAH...SYKREQPSIFAWEIRDKLLHEKVCSPDTIPSV
## subject: [8] SGHTGVNQLGGVFVNGRPLPDSTRQKIVDLAH...SYKREQPSIFAWEIRDKLLHEKVCSPDTIPSI
## score: 622
```

```
printPairwiseAlignment(la, chunksize = 30)
```

SGHTGVNQLGGVFVNGRPLPDSTRQKIVDL 30
SGHTGVNQLGGVFVNGRPLPDSTRQKIVDL 30

AHQGARPCDISRILQVSNGCVSKILCRYYE 60
AHQGARPCDISRILQVSNGCVSKILCRYYE 60

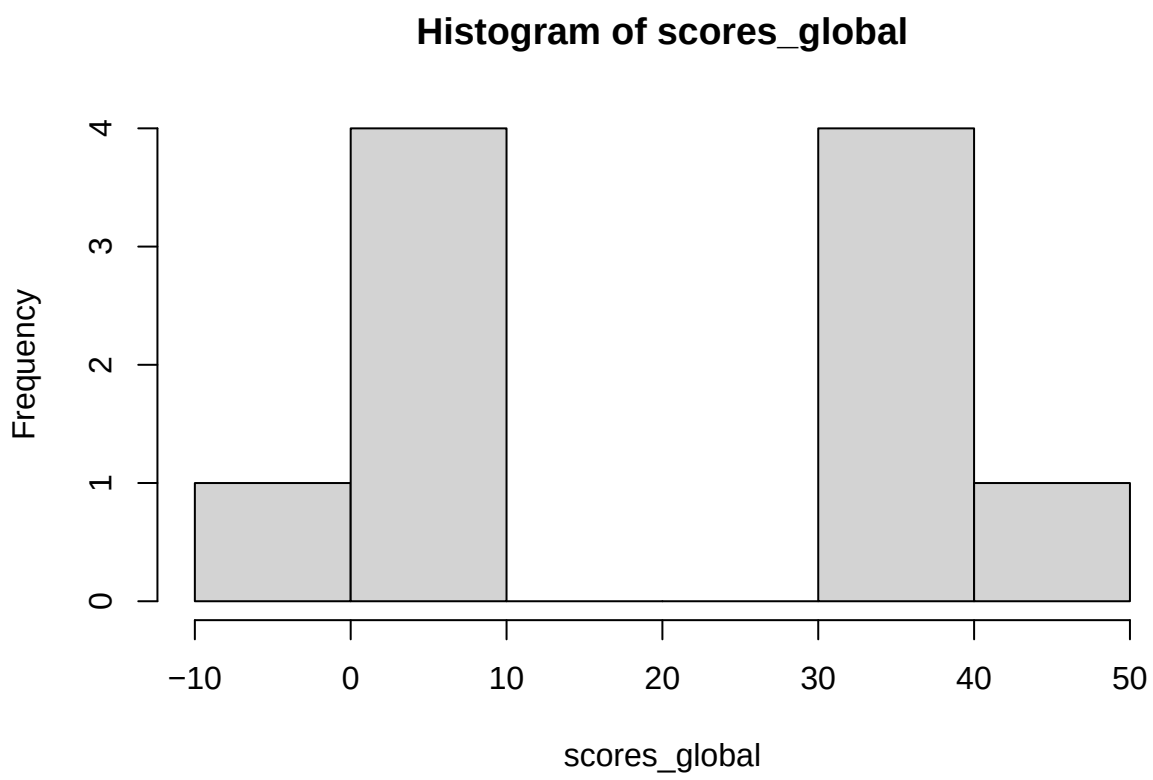
SGTIRPRAIGGSKPRVATVSVCDKIESYKR 90
SGTIRPRAIGGSKPRVATVSVCDKIESYKR 90

EQPSIFAWEIRDKLLHEKVCSPDTIPSV 120
EQPSIFAWEIRDKLLHEKVCSPDTIPSI 120

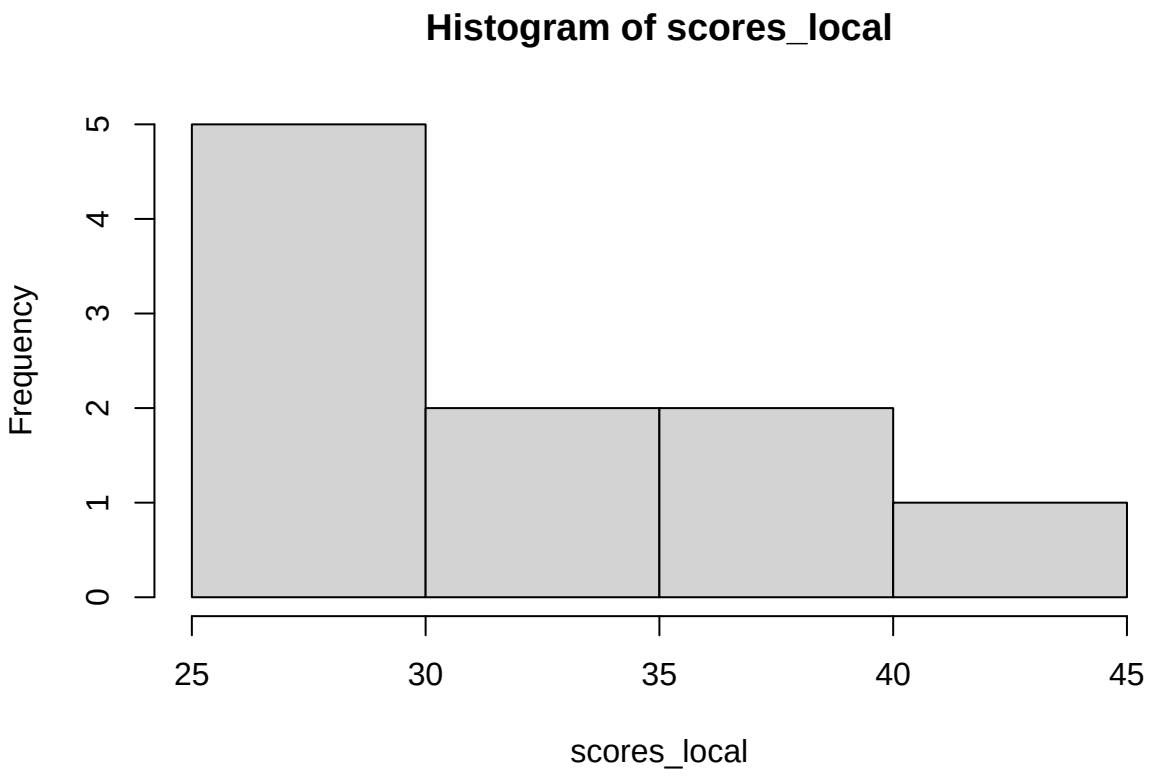
The local alignment's score is higher than the global alignment. With the pairwise printed output we can see that there is just a single mismatch.

## Task F

```
rand_seq1 <- generateSeqsWithMultinomialModel(c2s(seq1), 10)
rand_seq2 <- generateSeqsWithMultinomialModel(c2s(seq2), 10)

scores_global <- c()
scores_local <- c()

for (i in 1:length(rand_seq1)) {
    score_global <- pairwiseAlignment(pattern = AAString(c2s(rand_seq1[[i]])),
        subject = AAString(c2s(rand_seq2[[i]])), substitutionMatrix = PAM250,
        gapOpening = -10, gapExtension = -0.5, type = "global",
        scoreOnly = TRUE)
    score_local <- pairwiseAlignment(pattern = AAString(c2s(rand_seq1[[i]])),
        subject = AAString(c2s(rand_seq2[[i]])), substitutionMatrix = BLOSUM62,
        gapOpening = -10, gapExtension = -0.5, type = "local",
        scoreOnly = TRUE)
    scores_global <- append(scores_global, score_global)
    scores_local <- append(scores_local, score_local)
}

hist(scores_global)
```




```
hist(scores_local)
```




```
p_global <- sum(scores_global > score(ga))/length(scores_global)
p_local <- sum(scores_local > score(la))/length(scores_local)

print(p_global)
```

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

```
print(p_local)
```

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

Both the alignments are statistically significant as the 'p' values for both are less than 0.05.