Homework 2 part 2 (type 3) on Pairwise Alignments

Instructions

Pairwise sequences alignment is a "classical" Bioinformatics topic. The following tasks aim to solidify your knowledge of global and local alignments. In addition, the element of applicability is supplied by the last questions employing R and relevant libraries.

Below you will find a series of questions that you should provide answers to. The provided R-code "should" run without any issues.

Author: Kyrylo Bessonov

Help: Please contact me preferential via email or via my office phone. I would be glad to help you out

Questions

1) Explain what are the two major types of sequence alignment? Highlight their commonalities and differences
2) Define in your own words Dynamic-Programming. Why this approach speeds up local and global alignment algorithms?
3) Please align globally using Needleman–Wunsch algorithm the following DNA sequences. Use The following scoring rules: a) gap -5; b) match between two nucleotides +5; c) mismatch between two nucleotides +3;
    - **Sequence A**: `CTTGAA` ; **Sequence B**: `CTT`
        - Show the best alignment and its score.
        - Show the alignment matrix (template given below)
        - Show the trace-back path giving the best alignment (i.e. arrows)

		C	T	T	G	A	A
C							
T							
T							

4) Align locally the following sequences using the following scoring rules: a) gap -2; c) mismatch -1 3) match +2

- Sequence A: *GGTATACC* ; Sequence B: *TATA*
- Show the best alignment with its corresponding score
- Show the alignment matrix and trace-back path

		G	G	T	A	T	A	C	C
T									
A									
T									
A									

5) Do **local** protein alignment **using BLOSUM 62** matrix on the *HEAGAWGHEE* and *PAWHAE* sequence. The scoring rules are a) gap -8; matches and mismatches are given in BLOSUM 62 matrix.
   a. State the best alignment and its score
   b. Show the alignment matrix and trace-back path to arrive to the best local alignment

		H	E	A	G	A	W	G	H	E	E
P											
A											
W											
H											
A											
E											

Aligning sequences via Biostrings R library

6) Manually download the DNA sequences of *Brugia malayi* Vab-3 protein (UniProt accession A0A0J9XT59) and the *Loa loa* Vab-3 protein (UniProt accession E1FTG0) via the following commands. Load them into R via the following key commands.

   Tip: save the FASTA sequences in a text files (A0A0J9XT59.txt and E1FTGO.txt)

```
install.packages("seqinr");
library("seqinr"); #load library
brugia <- read.fasta(file = "A0A0J9XT59.txt"); #read seq into R
brugiaseq <- as.character(brugia[[1]]); #copy only sequence data
loa <- read.fasta(file = "E1FTGO.txt"); #the same for the other
loaseq <- as.character(loa[[1]]);
```

7) What is the alignment score for the optimal global alignment between the *Brugia malayi* Vab-3 protein and the *Loa loa* Vab-3 protein, when you use the *BLOSUM50* scoring matrix, a gap opening penalty of -9.5 and a gap extension penalty of -0.5? Use the `pairwiseAlignment()`. Read the manual here. Paste your alignment below.

```
library("Biostrings");
data(BLOSUM50);
brugiaseqstring <- c2s(brugiaseq); #convert to one string
brugiaseqstring <- toupper(brugiaseqstring); #convert to upper case for input
loaseqstring <- c2s(loaseq);
loaseqstring <- toupper(loaseqstring);
myglobalAlign <- pairwiseAlignment(…);
```

8) What global alignment score do you get for the two Vab-3 proteins, when you use the *BLOSUM62* alignment matrix, a gap opening penalty of -10 and a gap extension penalty of -0.5?

```
data(BLOSUM62)
myglobalAlign2 <- pairwiseAlignment(…)
myglobalAlign2
```

9) What is the statistical significance of the optimal global alignment for the *Brugia malayi* and *Loa loa* Vab-3 proteins made using the *BLOSUM50* scoring matrix, with a gap opening penalty of -10 and a gap extension penalty of -0.5?

   How does the distribution of the random alignment scores looks like (show histogram)?

   What does the *p*-value represents?

   What can be concluded based from this *p*-value with respect to alignment results?

```
generateSeqsWithMultinomialModel <- function(inputsequence, X)
  {
     # Change the input sequence into a vector of letters
     require("seqinr") # This function requires the SeqinR package.
     inputsequencevector <- s2c(inputsequence)
     # Find the frequencies of the letters in the input sequence "inputsequencevector":
     mylength <- length(inputsequencevector)
     mytable <- table(inputsequencevector)
     # Find the names of the letters in the sequence
     letters <- rownames(mytable)
     numletters <- length(letters)
     probabilities <- numeric() # Make a vector to store the probabilities of letters
     for (i in 1:numletters)
     {
        letter <- letters[i]
        count <- mytable[[i]]
        probabilities[i] <- count/mylength
     }
     # Make X random sequences using the multinomial model with probabilities "probabilities"
     seqs <- numeric(X)
     for (j in 1:X)
     {
        seq <- sample(letters, mylength, rep=TRUE, prob=probabilities) # Sample with replacement
        seq <- c2s(seq)
        seqs[j] <- seq
     }
     # Return the vector of random sequences
     return(seqs)
  }
```

```
randomseqs <- generateSeqsWithMultinomialModel(brugiaseqstring,1000);
randomscores=c();
for (i in 1:1000)
  {
     score <- pairwiseAlignment(loaseqstring, randomseqs[i], substitutionMatrix =
"BLOSUM50", gapOpening = -9.5, gapExtension = -0.5, scoreOnly = TRUE)
     randomscores[i] <- score
  }

pvalue = sum(randomscores >= myglobalAlign@score)/1000;
if(pvalue==0){pvalue = 1/1000}
print(pvalue);
```

10) Download the mRNA sequence in FASTA format here of the *Loa loa* Vab-3 protein (NCBI ref.: XM_003139724.1). Calculate the GC content %(G+C)