Homework 1
Due: Saturday Feb 8, before class starts

Note: the official language for this HW assignment is MATLAB. However, if you have some trouble with MATLAB, you can use another language (c/c++/java/python). Those who use MATLAB for implementation will receive 5 additional points for each of the two implementation assignments. Including a README with instructions to compile/run your program is required for non-MATLAB implementations.

Answers to Question 1, 3, and 5 can be turned in as hard copy in class. Source code for Q2 and Q4 should be uploaded in blackboard.

Problem 1 (25 points): Biology

You are given the following DNA sequence, which is believed to contain a small protein-coding gene.

GGAGGCGTAA AATGCGTACT GGTAATGCAA ACTAATGG

- If this sequence is fully transcribed (used as a coding strand), what is the corresponding mRNA sequence?
- Which region of the mRNA do you think can be translated into a protein (hint: Can you identify the start codon and stop codon from the mRNA sequence?)
- What is the protein sequence encoded by this gene?
- If the reverse-complementary strand of this DNA sequence is also transcribed, what will be the mRNA sequence?
- Do you think the reverse-complementary strand of this DNA sequence can encode a protein by itself (see hint above) ?

Problem 2 (25 points): Open reading frame finder.

An Opending Reading Frame (ORF) is a continuous stretch of codons (nucleotide triplets) that contain a start codon (i.e., ATG) at the beginning and a stop codon (i.e., TAA, TAG or TGA) **at the end only**, i.e., with no stop codon in the middle (https://en.wikipedia.org/wiki/Open_reading_frame). Note that there are three different ways (frames) that you can convert a DNA sequence into triplets, each shifting one nucleotide from another.

Problem 2.1 (5 points): Reverse complementary strand.

Write a MATLAB function getReverseComp that takes a string as a DNA sequence and return its reverse complementary strand. Save as file getReverseComp.m.

Test it on the command line by typing in: getReverseComp('ACGTGCA') or run the corresponding cell in hw1q2script.m.

Problem 2.2 (2 points): Identify possible start codons.

Write a MATLAB function findStartCodon that takes a string as a DNA sequence, and returns the indices of all possible start codons. You may need the function strfind (type "help strfind" on matlab command line for usage.) Save as file findStartCodon.m.

Test it on the command line by typing in: findStartCodon('AATGTATGA') or run the corresponding cell in hw1q2script.m.

Problem 2.3 (3 points): Identify possible stop codons.

Write a MATLAB function findStopCodon that takes a string as a DNA sequence, and returns the indices of all possible stop codons. The indices should be sorted. Save as file findStopCodon.m.

Test it on the command line by typing in: findStartCodon('ATAAGTAGGA') or run the corresponding cell in hw1q2script.m.

Problem 2.4 (15 points) Identify the longest open reading frames(ORF)

Write a MATLAB function that takes as input a DNA sequence, and returns the longest ORF, which is described as two numbers (the start index of the start codon, and the END index of the stop codon). Save as file findLongestORF.m.

To test your function on the command line, type in:
findLongestORF('GGAGGCGTAAAATGCGTACTGGTAATGCAAACTAATGG')
or run the corresponding cell in hw1q2script.m.

Test

To Test the functions, use the attached HW1q2script.m script. It reads a sequence from a sequence file sequence.fa, in FASTA format (which is one of the most popular and simplest format) and tests each of the functions above and output some statistics.

Problem 3 (25 points): Global and local alignment - manually without coding

Consider the sequences v = TACGGGTAT and w = GGACGTACG. Assume that the match score is +1, and the mismatch and gap penalties are -1.

- Fill out a dynamic programming table for a global alignment between v and w. Draw arrows in the cells to store traceback information. What is the score of the optimal global alignment and what alignment(s) achieves this score?
- Fill out the dynamic programming table for a local alignment between v and w. Draw arrows in the cells to store traceback information. What is the score of the optimal local alignment in this case and what alignment(s) achieves this score?

Problem 4 (25 points): Global alignment

Use MATLAB to implement the Needleman-Wunsch algorithm with m = 1, s = -1, d = -1. The input and output of your function should be as follows.

Input: two DNA sequence strings.

Output:

(1) the optimal global alignment score between the two sequences.

(2) the number of matched characters

(3) the number of mismatched characters

(4) the number of insertion / deletions in the alignment.

(5) Optional (10 pts): the actual alignment, which is specified as below:

The output alignment should have three lines as shown in the example below, where matching characters are shown by a | character, mismatches by a dot (.), and gaps by a dash (-). For longer sequences, break the alignment into lengths of 50.

```
ACGTACGTAG--GACGTAAGCAGAGAACGAGAACCCGGGAAC-ACGAGGC
||.||. ||| |||.|||||..||||.||.||||| ||||| |||||||
ACCTAG-TAGCGGACTTAAGCGTAGAAGGACAACCC-GGAACGACGAGGC
TGGTCGGCTT
|.||||.|||
TGGTCGTCTT
```

To test:

First try your implementation on the sequences used in Problem 3 to make sure it works correctly. Then download hw1prob4.fa and use your program to align each pair of sequences in the file, and output the relevant alignment information returned by your function. (Use HW1q4script.m or modify.)

FYI, the sequences encode the hemagglutinin (HA) protein for different strains of the inuenza viruses.

Bonus (5 points): feedback

How much time did you spend on this homework? Who did you discuss with and what was the discussion about? How is the difficulty level? Do you have any comments about the course?