

Homework 1

Due: September 18, 8:30pm

Problem 1 (5 points)

- What is the complimentary sequence to the following string of nucleotides? Be sure to label the 5' and 3' ends.
5'- GCATATCGTAATGCCATA - 3'
- Show the mRNA transcript using the above sequence as a coding strand.
- Show the final protein sequence.

Problem 2 (15 points)

In the lecture, I have mentioned that there are a few different ways of defining “distinct” alignments. Depending on the definition, the number of distinct alignments will be different. Consider for example the three alignments between abc and wxy: $\begin{array}{c} abc \\ xyz \end{array}$, $\begin{array}{c} a-bc \\ xy-z \end{array}$, and $\begin{array}{c} ab-c \\ x-yz \end{array}$. They can be handled in two ways:

(a) the three alignments are all different from each other; (b) the latter two are equivalent but distinct to the first one. The argument for (b) is that in the first alignment, a, b, c are aligned to x, y, and z respectively, while in the latter two cases, a is aligned to x, c is aligned to z, and b, y are aligned to gaps.

Given two strings S , T of lengths m and n , let $f(m, n)$ be the number of distinct alignments between them according to the second definition. In the lecture, we have discussed that $f(m, n)$ can be obtained analytically using probability theory. The formula is $f(m, n) = \binom{m+n}{n} = \frac{(m+n)!}{m!n!}$. (Durbin book).

In this assignment, you are asked to obtain the number of distinct alignments, $F(m, n)$, according to the first definition. There are two ways you can do it. The first is by using probability theory directly. The second is by using dynamic programming. You are free to choose either one, but the second is more related to the lectures.

For this assignment, you don't need to write a program.

- (1) Write down the recurrence or the formula.
- (2) Draw a table and fill in values from $F(0, 0)$ to $F(10, 10)$.
- (3) List all possible alignments between ab and xyz. See if the number agrees with the value $F(2, 3)$ in your table.

Problem 3 (15 points)

In real sequence alignment, alternating gaps are usually prevented by properly defining the mismatch and gap penalties such that $-s > -2d$. With this scoring scheme, it is easy to see that a gap in one string will not be followed immediately by another gap in the other string (i.e., $\begin{smallmatrix} a- \\ -b \end{smallmatrix}$), because the score of having two gaps will be lower than simply aligning a and b.

This restriction significantly reduces the number of possible alignments. Write down a recurrence for $F(m, n)$ in this case. You may need more than one state (similar to affine-gap penalty).

- (1) Write down the recurrence.
- (2) Use dynamic programming (on paper) to count the number of alignments for $m = 3$ and $n = 3$. Write down the table(s) you used during your calculation.
- (3) Enumerate all alignments between ab and xyz, and all alignments between abc and xyz. Confirm that your table is correct.

Problem 4 (20 points)

Implement the Needleman-Wunsch and the Smith-Waterman algorithms with $m = 1$, $s = -1$, $d = -1$. (There are many implementations available on the Internet, but please write one by yourself.)

The input and output of your programs should be as following.

Input: two sequence files. Each file contains one or more sequences. Each line contains a single sequence.

Output: compute the optimal (global or local) alignment between every seq in file1 against every seq in file2. Save the optimal scores to a file. It's not necessary to implement the trace-back part for this assignment. But you may want that to test the correctness of your program anyway.

Download file1.txt, file2.txt, file3.txt, file4.txt from the course website. Each file contains a set of random sequences of length 60. File1 and file2 were generated with equal base frequencies ($p_A = p_C = p_G = p_T = 0.25$), while file3 and file4 were generated with $p_A = p_T = 0.38$, and $p_C = p_G = 0.12$.

Use your programs to align file1 vs file2 and file3 vs file4. Save your results into four files: (1) f1.f2_global.txt, (2) f1.f2_local.txt, (3) f3.f4_global.txt, and (4) f3.f4_local.txt.

Plot the distributions of the four sets of scores. What can you say from these plots? Turn in the plots, and a brief discussion of your observations. Do NOT turn in your program or the alignment scores!

Bonus (5 points)

How much time have you spent? Who did you discuss with and what was the discussion about? What programming language did you use?