Homework 1 Solution

Problem 1 (15 points)

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?

Since the DNA sequence is used as a coding strand, the mRNA sequence is simply a copy of the DNA sequence, with T’s substituted by U’s.

GGAGGCGUAA AAUGCGUACU GGUAUUGCAA ACUAAUGG

• 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?)

There are three start codons, but only one of them has an inframe stop codon (i.e., number of nucleotides between start and stop codons is a multiple of 3), which marks a possible coding region.

AUG CGU ACU GGU AAU GCA AAC UAA

• What is the protein sequence encoded by the gene?

Met - Arg - Thr - Gly - Asn - Ala - Asn

The start codon also encodes Met. So the first amino acid for most proteins are Met. Some proteins undergo post-processing and the Met may be removed.

• If the reverse-complementary strand of the DNA sequence is also transcribed, what will be the mRNA sequence?

5’-CCATTAGTTT GCATTACCAG TACGCATTTT ACGCCTCC-3’

Remember: reverse—complementary. You need to reverse the strand after writing down the complementary base for each nucleotide.

• Do you think the reverse-complementary strand can encode a protein?

This question is somewhat ill-posed given the size of the sequence. This reverse complementary strand has no start and stop codons. So it cannot encode a protein all by itself. However, it could be part of a longer gene that does encode a protein. In fact, long stretches of DNA sequences with no stop codons is one of the signal used for finding genes.

Problem 2 (20 points)

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 the 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?

The optimal score is -1, and there are 2x3x2 = 12 alignments achieving this score (see the paths below). Most of you have got it right. Some of you only show a single optimal alignment. There is nothing wrong with it, but I was hoping you could notice that there are alternative paths during the trace-back.
**Problem 3 (10 points)**

Most of you did a great job in implementing Needleman-Wunsch. A couple of things that I was hoping that you could find:
1. All three sequences start with ATG and end with stop codon (TAA, TGA, and TAA). The lengths of all three sequences are multiples of three (1701, 1701, and 1707). These sequences all encode complete genes and do not contain any non-translated region. A couple of you went a step further and showed that there is no stop codon in the open reading frame. Nice job!

2. The length of the alignment between sequences A and B is 1728, which tells you that there are 27 gaps. As 27 is a multiple of 3, it seemed that NW did a good job in preserving the open reading frame. But there are still places that can be improved if you know that the regions are open reading frames. See for example the first 200 characters in the alignment:

```
ATGAAGGCAAATACTAGATG-TCTGCTATATACATTTGCAACCGCAAATG
   ||||.|||||||.|.||.||-||.|||.|.||||||||.|..||||||
ATGGAGGCAAGACTCTGTGCT-TGTTATGTGAATTTCGAGCTTACAAATG
   ||||.|||||||.|.||.||-||.|||.|.||||||||.|..||||||
CAGACACATTATGTATAGGCTTATCAATACCTAACTCAACAGACACTGTA
   ||||.|||||||.|.||.||-||.|||.|.||||||||.|..||||||
GACACAGTACAAAGAAATGGAACGTAAGAAGACACCTGTAACTTTCTCT
   ||||.|||||||.|.||.||-||.|||.|.||||||||.|..||||||
GACACAGTACTCGAAAAAGAATGCGACGTCGACAGCTCTATTTAATCCCT
   ||||.|||||||.|.||.||-||.|||.|.||||||||.|..||||||
AGAAGAACAG-CATAACGGGAAACTAATGGAAGGTTACGCCCCCA
   .||||||-||-||.|||||.||||||||.|||.|||.|||..|||||||
CGCAAAGC-AGCCACAACGGAAAACATATGAATATTAAAGGAAATGCCCA
```

In the first 50 characters of the alignment, there are two gaps separated by a single match: -TC matched with CT-. This has a score of -1. It would make more biological sense (to preserve reading frame) to replace it with two mismatches (TC vs CT) but will result in a score of -2. There are multiple examples like that in all three pairwise alignment. Between B and C, there are some longer gaps and the size of the gap is not always a multiple of three.

Two possible improvements: (1) Use higher gap opening cost to discourage gap occurring; (2) Translate nucleotides into amino acid sequences before alignment.

**Problem 4 (10 points)**

There were three paragraphs I copied from one article to the other. I also reordered a couple of sentences in one of the paragraphs (not the top scoring). Most of you were able to at least identify the top scoring matched paragraphs.

**Problem 5 (10 points)**

Describe how to achieve linear space for the Smith-Waterman algorithm. (Hint – think about two subproblems: to get the score of the optimal local alignment in linear space, and to find the corresponding alignment in linear space.)

To find the score of the optimal local alignment with linear space is easy. You can run Smith-Waterman with only two rows (call it linear-space SW algorithm), and keep track of the best score you have found so far. To find the corresponding alignment in linear space is tricky. The Hirschberg algorithm for global alignment can not be used directly - since the optimal local alignment might not pass the middle point of either string (for example, see the figure below). The key point, however, is that if we know the start and end points of the optimal local alignment, \((I_1, J_1)\) and \((I_2, J_2)\), we can easily recover the actual alignment with linear space by running the Hirschberg algorithm on the small rectangle constrained by the two points (see figure below).

To find the end point of the optimal local alignment with linear space is easy too. Above we have discussed how to find the score for the optimal alignment using linear space SW. If we also keep track of
the location of the cell associated with the currently best score, we will have the end point of an optimal alignment when the linear-space SW finishes.

So the only tricky part is how to find the starting point of an optimal local alignment. There are two options. First, after we have found the end point \((I_2, J_2)\) of an optimal local alignment, we can run the linear space SW backwards, starting from \((I_2, J_2)\). We terminate this backward linear-space SW when we have reached the same optimal score as found in the forward iteration. The cell \((I_1, J_1)\) with the optimal score is a valid start point, since the alignment between \((I_1, J_1)\) and \((I_2, J_2)\) has the optimal score. In the second option, we run the linear-space SW in the forward direction only. For each cell in the two rows that we do keep track of, we record (1) a score for the best local alignment ending in that cell and (2) the starting point for that best local alignment. Initially the starting point of each cell is just itself. When an alignment path continues, each cell copies the starting point from its preceding cell on the path. The starting point is set to a cell itself whenever a score zero is reached.

When the local alignment is relatively short compared to the sequences, the first option is more efficient. When there are multiple optimal (or near optimal) alignments that need to be traced back, the second option may be better because the first strategy would require a separate backward iteration for each alignment that need to be traced back.