CS466: Introduction to Bioinformatics **Fall 2018**

Problem Set 2

Handed out: September 28, 2018 *Due: October 5, 2018*

Instructions: This homework assignment consists of four questions worth a total of 100 points. These questions are based on the material covered in lectures 5 to 8. Please use the space provided to write your answers.

1. **Linear Space Alignment** [25 points]

   Consider two sequences $\mathbf{v} = \text{CT}$ and $\mathbf{w} = \text{GCAT}$ of length $m = |\mathbf{v}| = 2$ and $n = |\mathbf{w}| = 4$, respectively. In this exercise, we will compute an optimal global alignment of the two sequences using the Hirschberg algorithm. We will use a score of +1 for a match, -1 for a mismatch, and -1 for a insertion/deletion (i.e. a gap penalty of 1).

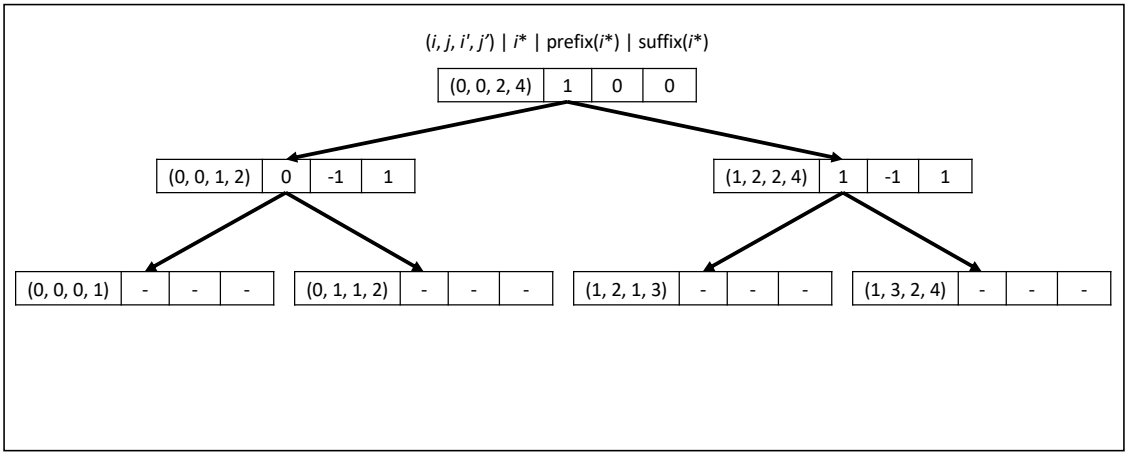
   a. The initial call is $\textsc{Hirschberg}(0, 0, m = 2, n = 4)$. We need to identify the middle vertex $(i^*, n/2 = 2)$. Fill out the following table for this initial call and indicate i^*. [5 points]

i	prefix(i)	suffix(i)	wt(i)
0	-2	0	-2
1	0	0	0
2	-1	-2	-3

   b. What are the two recursive calls that are made in this initial invocation $\textsc{Hirschberg}(0, 0, m, n)$? [5 points]

   $\textsc{Hirschberg}(0, 0, 1, 2)$ and $\textsc{Hirschberg}(1, 2, 2, 4)$

   c. Give the recursion tree, where each vertex corresponds to an invocation of Hirschberg. See slide 44 of lecture 1 for an example of a recursion tree. Label each vertex of this tree by the used arguments (i, j, i', j'). In addition, label each *internal* vertex by the value of i^*, prefix(i^*) and suffix(i^*). [10 points]



d. Indicate the reported vertices in the table and give the final alignment. [5 points]

Reported vertices ('X'):

	0	G	C	A	T
0	X	X			
C			X	X	
T					X

Final alignment:

-	C	-	T
G	C	A	T

2. **Sub-quadratic Time Alignment** [15 points]

Global pairwise sequence alignment of two sequences of length m and n takes $O(mn)$ time using dynamic programming. This corresponds to quadratic time. In this question, we will explore algorithms that run in sub-quadratic time.

a. We start with banded alignment, which was introduced in Lecture 5. Briefly, the idea is to restrict alignments, which are paths from $(0, 0)$ to (m, n), to only occur in a band of width k around the diagonal. Modify the recurrence of global sequence alignment to achieve this change. [5 points]

With 0-based index, i.e. the case where only the main diagonal is allowed corresponds to $k = 0$, we have

$$s[i,j] = \max \begin{cases} 0, & \text{if } i = 0 \text{ and } j = 0, \\ s[i-1,j] + \delta(v_i, -), & \text{if } i > 0 \text{ and } j - i < k, \\ s[i,j-1] + \delta(-, w_j), & \text{if } j > 0 \text{ and } i - j < k, \\ s[i-1,j-1] + \delta(v_i, w_j), & \text{if } i > 0 \text{ and } j > 0. \end{cases}$$

With 1-based index, i.e. the case where only the main diagonal is allowed corresponds to $k = 1$, we have

$$s[i,j] = \max \begin{cases} 0, & \text{if } i = 0 \text{ and } j = 0, \\ s[i-1,j] + \delta(v_i, -), & \text{if } i > 0 \text{ and } j - (i-1) < k, \\ s[i,j-1] + \delta(-, w_j), & \text{if } j > 0 \text{ and } i - (j-1) < k, \\ s[i-1,j-1] + \delta(v_i, w_j), & \text{if } i > 0 \text{ and } j > 0. \end{cases}$$

b. In class we learned how to solve the block alignment problem in time $O(n^2/\log n)$ using the Four Russians Technique. Specifically, in the case of of an alphabet of $|\Sigma| = 4$ letters, we pre-computed *all* pairwise alignments of *all* strings of length $t = \log_2(n)/4$. Protein sequences have an alphabet of $|\Sigma| = 20$ letters. How should we choose length t to achieve the time bound of $O(n^2/\log n)$ if $|\Sigma| = 20$? Motivate your answer. [5 points]

Set $t = \log_2(n)/(2\log_2(20))$. The number of sequences of length t is

$$\begin{aligned} 20^t = 20^{\log_2(n)/(2\log_2(20))} &= (2^{\log_2(20)})^{\log_2(n)/(2\log_2(20))} \\ &= (2^{\log_2(n)})^{\log_2(20)/(2\log_2(20))} = n^{1/2}. \end{aligned}$$

Thus, the number of alignments that we need to precompute is

$$20^t \cdot 20^t = n^{1/2} \cdot n^{1/2} = n.$$

The rest of the analysis is the same as described in Lecture Notes 5, and the time bound of $O(n^2/\log n)$ follows.

c. In their STOC 2015 paper, Backurs and Indyk proved that the edit distance problem cannot be solved in time $O(n^{2-\epsilon})$ where $\epsilon > 0$ under the Strong Exponential Time Hypothesis. Show that the same result also holds for pairwise global sequence alignment by choosing an appropriate scoring function $\delta : (\Sigma \cup \{-\}) \times (\Sigma \cup \{-\}) \to \mathbb{R}$ such that an optimal edit distance alignment is also an optimal global sequence alignment. [5 points]

Bonus: +5 points if you give an actual proof of why the reduction works.

Define scoring function δ as

$$\delta(v_i, w_j) = \begin{cases} 0, & \text{if } v_i = w_j, \\ -1, & \text{if } v_i \neq w_j, \\ -1, & \text{if } v_i = -, \\ -1, & \text{if } w_j = -. \end{cases}$$

Proof: Alignments in the edit distance and global sequence alignment problem are defined identically: gaps '$-$' are inserted into $\mathbf{v}$ and $\mathbf{w}$ such that (i) the resulting gapped sequences $\mathbf{v}'$ and $\mathbf{w}'$ have the same length k and (ii) there is no column $i \in [k]$ where $v'_i = w'_i = -$. In edit distance, each mismatch, insertion and deletion column contributes a weight of 1 to the distance, whereas a match does not contribute to the distance. In the above scoring function δ for global sequence alignment, each mismatch, insertion and deletion column contributes a weight of -1 to the score, whereas a match does not contribute to the distance. Let d denote the cost of an alignment using the edit distance scoring and let s denote the score of that same alignment using δ. We have that $d = -s$. In the edit distance problem we are minimizing the distance d, while in the global sequence alignment problem we are maximizing the score s. Since $d = -s$ for any alignment, both problems have the same optimal alignments. Thus, an $O(n^{2-\epsilon})$ algorithm for pairwise global sequence alignment allows us to solve the edit distance problem in the same time bound.

3. **Nussinov Algorithm** [45 points]

   Consider the RNA sequence $\mathbf{v}$ = GCAAGACU . In this exercise, we use the Nussinov algorithm to find the pseudoknot-free secondary structure of $\mathbf{v}$ with the maximum number of complementary base pairings. We will use three different algorithm configurations. In each case, show the structure (in the dot-parenthesis format discussed in class), the optimal score and the dynamic programming table with the backtrace clearly indicated.

   If there are multiple optimal structures, you can report any one of them.

   a. Consider a G-C and A-U match score of 1 with no constraints on minimum hairpin loop length. [15 points]

	G1	C2	A3	A4	G5	A6	C7	U8
G1	0	1	1	1	1	1	2	3
C2	0	0	0	0	1	1	1	2
A3		0	0	0	0	0	1	2
A4			0	0	0	0	1	2
G5				0	0	0	1	1
A6					0	0	0	1
C7						0	0	0
U8							0	0

Score = 3

G	C	A	A	G	A	C	U
(	)	.	(	(	.	)	)
(	)	(	.	(	.	)	)

b. Consider a G-C and A-U match score of 1 but prevent hairpin loops with 0 residues (i.e. minimum length ℓ is now set to 1). [15 points]

	G1	C2	A3	A4	G5	A6	C7	U8
G1	0	0	0	0	1	1	2	2
C2	0	0	0	0	1	1	1	2
A3		0	0	0	0	0	1	2
A4			0	0	0	0	1	2
G5				0	0	0	1	1
A6					0	0	0	1
C7						0	0	0
U8							0	0

Score = 2

G	C	A	A	G	A	C	U
(	(	.	.	)	.	.	)
(	(	.	.	)	.	)	.

c. Consider a G-C match score of 4 and A-U match score of 2 but prevent hairpin loops with 0 residues (i.e. minimum length ℓ is set to 1). [15 points]

	G1	C2	A3	A4	G5	A6	C7	U8
G1	0	0	0	0	4	4	8	8
C2	0	0	0	0	4	4	4	6
A3	0	0	0	0	0	0	4	6
A4	0	0	0	0	0	0	4	6
G5	0	0	0	0	0	0	4	4
A6	0	0	0	0	0	0	0	2
C7	0	0	0	0	0	0	0	0
U8	0	0	0	0	0	0	0	0

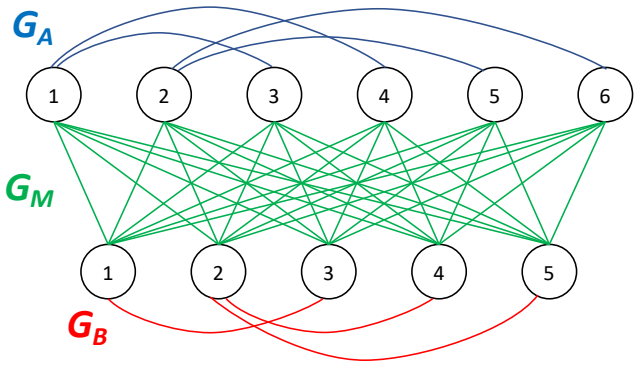
Score = 8

G	C	A	A	G	A	C	U
(	(	.	.	)	.	)	.

4. **Contact Map Overlap** [15 points] Consider protein sequences A = NLITRH and B = HLNTK with the following two contact maps.

$$C^A = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad C^B = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

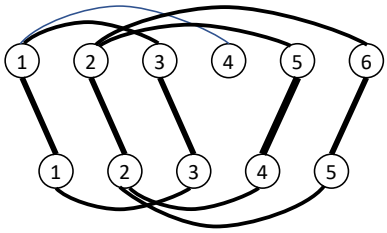
a. Give the matching graph representation of C^A and C^B, consisting of three graphs $G_A = (V_A, E_A)$, $G_B = (V_B, E_B)$ and $G_M = (V_A \cup V_B, E_M)$. Clearly indicate the three graphs. [5 points]



b. Explain why the maximum number of conserved contacts is *at most* $\min\{|E_A|, |E_B|\} = 3$. [5 points]

In an alignment, each residue in one input protein has at most one counterpart in the other input protein. As such, each contact in one input protein can only be paired up with at most one contact in the other protein. Thus, the number of conserved contacts is at most the minimum number of input contacts. Here, protein A has $|E_A| = 4$ contacts and protein B has $|E_B| = 3$. Hence, there can be at most $\min\{|E_A|, |E_B|\} = 3$ preserved contacts.

c. Give an alignment with the maximum number of preserved contacts. State the number of preserved contacts. [5 points]



Three preserved contacts