Algorithms in Computational Biology – Homework Exercise 1

Publication date: *Thursday, November 12*
Due date: *Sunday, November 29 (9pm IST)*

Problem 1:

Denote by numAlign(n,m) the number of valid pairwise (global) sequence alignments of two sequences of length m and n. Prove that:

$$\binom{m+n}{m} < \text{numAlign}(m,n) < \binom{m+n}{m}^2$$

Hint: use the formal definition of alignments as pairs of gapped sequences S', T' and define two 1-1 mappings: one from the set of all alignments to pairs of gapped sequences and one from the set of single gapped sequences to a subset of all alignments.
S

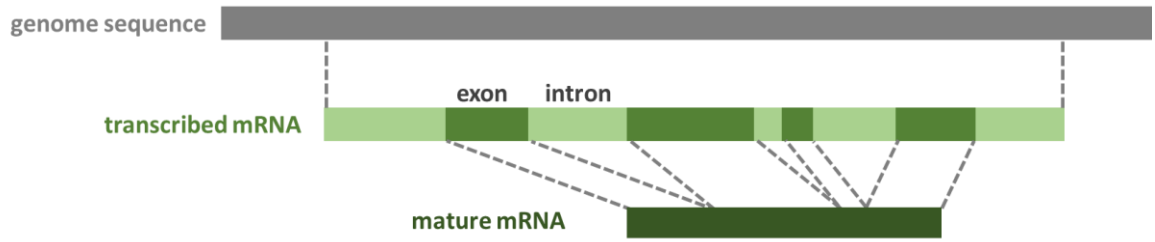
Problem 2:

We saw in class a linear-space implementation of the Needleman-Wunsch dynamic programming algorithm for finding a maximum score global alignment between two sequences (Hirschberg's trick). In this question, you will develop a space efficient version for the Smith-Waterman algorithm for finding a maximum score local alignment.

a. Explain shortly why the technique used for global alignment cannot be used **as is** in the local alignment case.
b. Suggest a linear-space version for the Smith-Waterman algorithm. Describe your algorithm in full detail.
c. Prove the correctness of your suggested algorithm and analyze its complexity. You may assume correctness and use the complexity bounds proven for all algorithms taught in class.

Problem 3:

This question deals with the problem of ***transcript alignment***. In most Eukaryotic organisms (ones where cells have a nucleus that contains all DNA), the mRNA transcript is processed after it is transcribed from DNA and before it is shipped out of the nucleus to be translated into an amino-acid sequence in the ribosome. The main post-transcriptional editing step is called ***splicing***, in which parts of the mRNA are cut and removed from the mature RNA. The sequences retained in the mature RNA are called ***exons*** (they are expressed) and the sequences spliced out of the mRNA are called ***introns*** (between exons). The process is illustrated below:



Transcript alignment is the problem of aligning sequences corresponding to mature RNA to genome sequences. In this problem, a potential mature mRNA sequence (T) is aligned to a genome sequence (S).

a. In the 'gapless' version, any gap in the alignment is associated with an intron and substitutions may occur inside the exons (due to sequence errors). We are interested in finding an alignment of the mRNA to the genome sequence that would **minimize** the *number of implied introns + α x number of substitutions* (for some pre-specified value of α). Suggest an efficient algorithm for solving this problem. Describe your algorithm in detail, formally prove its correctness and analyze its complexity.

b. We are interested in extending this idea to allow indels inside exons. Indels can occur from sequencing errors, post-transcriptional edits, or differences between the genome sequence of the individual that the mRNA was taken from and the genome sequence S. Indels are allowed by finding an alignment of T and S with **maximum score**, under a general scoring scheme σ (as in standard versions of alignment), with the restriction that there are no more than k introns in the mRNA. The max number of introns k is part of the input and is typically much smaller than the length of the mRNA.

   Suggest an efficient algorithm for solving the problem of transcript alignment with gaps. Explain your algorithm and analyze its complexity. No need for formal proof in this case, but do make sure your algorithm is correct and that there are no loopholes in your informal arguments.

Problem 4:

Write a short computer program (in your language of choice) that computes the Smith-Waterman dynamic programming matrix for **local alignment** of the two sequences:

S = **ATAAGGCATTGACCGTATTGCCAA**

T = **CCCATAGGTGCGGTAGCC**

Under the scoring function σ defined as follows:

$\sigma(x,x) = 2$, $\sigma(x,y) = -2$ ($x \neq y$), and $\sigma(x,-) = \sigma(-,x) = -3$.

The values of S, T, and σ can be hard-coded in your program (the program does not have to accept them as parameters), but write the program in a way which would allow you to easily change the input.

a. Print the 25x19 matrix of values.
b. Use this matrix to trace back the path representing a max-score alignment. Do not show all 25x19 pointers. Just trace the ones that belong to an optimal alignment. You can do that manually on top of the program output.
c. Write down the alignment corresponding to the path you traced in (b).
d. Notice that there are several optimal local alignments of S, T. Specify another optimal local alignment that does not overlap the alignment you specified in (c) above.
e. Are there also optimal alignments of S, T that overlap each other? If so, give an example. If not, then explain why.
f. Modify your code to compute a maximum score local alignment in <u>linear space complexity</u> ($O(n)$) using Hirschberg's technique, and your algorithm from Problem 2. Minimize the space complexity of your implementation should be linear in the length of the shorter sequence (n), and the time complexity of the implementation should be $O(mn)$, as guaranteed by Hitrschberg's technique..

Provide the results for (a-f) and required explanations for (d-f) in the main solution document of your assignment, and submit your code as appendix to the exercise. Document your code clearly to indicate which parts are relevant for each section.

Submission Instructions:

- Submit your work on the course **Moodle website** by Sunday, Nov 29 @21:00.
- Type your solutions or write legibly and scan. **If you scan, make sure the scan came out fine.**
- **Submit your work in pairs!** One student should submit a solution file with both of your student IDs specified. The other student should submit a simple text file with your two student ids, to help us match back the grade to both of you.
- If you consult with other pairs on ideas, specify their names clearly on the first page and make sure that they **acknowledge your collaboration** as well.
- You have two weeks to complete the assignment. **Plan your time wisely**. Extensions due to special circumstances, will be granted only upon **request by e-mail at least 48 hours** before the deadline. No last minute extensions!
- Please post any questions that you have on the course Piazza website: https://piazza.com/idc.ac.il/fall2020/cs3571/.