CSE 181: Assignment 1

Due: April 12, 2011 at *noon*

Problem 1 (8 marks)

Suppose that we have sequences $v = \{v_1 \dots v_n\}$ and $w = \{w_1 \dots w_m\}$, where v is longer than w. We wish to find a substring of v that best matches all of w. Global alignment will not work because it would try to align all of v. Local alignment will not work because it may not align all of w. Therefore this is a distinct problem which we call the *Fitting Problem*. Fitting a sequence w into a sequence v is a problem of finding a substring v' of v that maximizes the score of alignment $s(v', w)$ among all substrings of v. For example, if v = GTAGGCTTAAGGTTA and w = TAGATA, the best alignment might be

	global	local	fitting
v	GTAGGCTTAAGGTTA	TAG	TAGGCTTA
w	- TAG - - - - A - - - T -A	TAG	TAGA - - TA
score	-3	3	2

The scores are computed as 1 for match, -1 for mismatch or indel. Note that the optimal local alignment is not a valid fitting alignment. On the other hand, the optimal global alignment contains a valid fitting alignment, but it achieves a suboptimal score among all fitting alignments.

Give an algorithm which computes the optimal fitting alignment. Explain how to fill in the first row and column of the dynamic programming table and give a recurrence to fill in the rest of the table. Give a method to find the best alignment once the table is filled in. The algorithm should run in time $O(nm)$.

Problem 2 (12 marks)

In the edit distance problem, a pattern consists of a set of ordered symbols. Each symbol is a feature vector. If these symbols are letters then the pattern is a word. Such problems arise in automatic editing and text retrieval applications. A string is read and matched against a set of known strings (or, dictionary). Then the task is to find the best match. For example the edit distance problem arises in the spell checker of a word processor. There are three basic types of errors:

- Wrong symbol (i.e. beκuty instead of beauty)
- Insertion error (bearuty)
- Deletion error(beuty)

Suppose that you have dictionary of size n containing all the correct words and m input words. Determine an algorithm that will take each of the input words and finds the closest 5 matches of that word in the dictionary. Your solution should include the following (a) pseudocode describing the algorithm; (b) a brief proof of the correctness of the algorithm; and the running time of the algorithm.

Problem 3 (10 marks)

Given a set of n strings $S = \{s_1, \ldots, s_n\}$, each of length m, and input parameters d and ℓ, the *closest string* problem aims to determine whether there exists a set $S' = \{s'_1, \ldots, s'_n\}$ that contains a ℓ-length substring s'_i in each $s_i \in S$, and a length-ℓ string x such that $d(x, s'_i) \leq d$ for all $s'_i \in S'$. $d(u, v)$ corresponds to the Hamming distance between strings u and v. What is the probability that an instance of the closest string problem has no solutions? You can assume that $m \geq \ell$, ℓ and d are non-negative integers, and that the alphabet size is of length 4 (i.e. the alphabet is $\{A, C, G, T\}$).

Problem 4 (8 marks)

Devise an algorithm to compute the number of distinct optimal local alignments (optimal paths in local alignment edit graph) between pairs of strings. Your solution should include a brief description and the running time.

Problem 5 (12 marks)

Many genomes contain an approximately equal proportion of `A`s, `C`s, `G`s, and `T`s. Plasmodium falciparum, the parasite responsible for causing malaria, has a particularly (`A`+`T`)-rich genome that is, there are significantly more `A`s and `T`s in the Plasmodium genome than `C`s and `G`s. How would this fact affect the ability of the following motif finding techniques to return significant hits? As a way to be specific about your descriptions, compare how motif finding would function over the Plasmodium genome as compared with motif finding over a genome with equal proportions of bases.

a. Consensus (Hertz and Stormo)

b. GibbsDNA (Lawerence et al.)

c. MEME (Bailey and Elkan)