Computational Genomics
© Benny Chor and Amos Tanay
School of Computer Science
Tel Aviv University
November 11, 2004

Computational Genomics: Assignment No. 2
due on November 29th, 2004

General Guidelines:
This assignment is part of your final grade in the course. It should be done *independently*, either individually or in *pairs*. Duplicated and copied works will be given zero grade. Using articles or books is perfectly acceptable as long as you include the reference in your relevant answer.

Credit: Solve all items for 125 points + one golden star. **Bonus** sections are harder, but you can accumulate the full 100% grade even without solving them.

1. (20 pts) Recall that the Nussinov algorithm for RNA folding assumes all pairings are disjoint or nested (i.e. have no pseudo-knots). How many possible pairing configuration of this form exist?

2. (20 pts) **Distance vs. Similarity**. In class we argued that we can replace sequence similarity by distance and obtain qualitatively similar results for pairwise *global sequence alignment*. Let $\delta(\cdot,\cdot)$ be the distance function between pairs of letters (including a gap), and let $D(S,T)$ denote the cost of the optimal global alignment (minimum distance) between two *sequences* S and T. Show that if $\delta(\cdot,\cdot)$ is a distance function than $D(\cdot,\cdot)$ is a distance function as well (e.g. that it satisfies the triangle inequality).

3. **Multiple Sequence Alignment**

   The multiple alignment approximation algorithm shown in class aligns all sequences to a well-chosen "center" sequence. Finding this "center" sequence dominates the running time of the algorithm. In this problem we will investigate how essential the choice of the "center" sequence is. ($D(M)$ denotes the SP score of a multiple alignment M, and as in class we will assume the cost matrix δ is a metric)

   (a) (10pt) Show that choosing a different sequence and aligning all sequences to it may yield an infinitely high approximation ratio. Do this by showing that for every ratio r, there exists a set of sequences, a cost matrix, and a "bad" sequence b, such that the SP score of the multiple alignment M_b found using b is worse then the optimal alignment M_{opt} by at least r: $D(M_b)/D(M_{opt}) \geq r$. (**Bonus:** what's the worst ratio you can get with k sequences ?)

(b) (15pt) Show that despite this, for any set of sequences, the mean ratio over all "center sequence" choices is at most 2: $E_b\left(D(M_b)/D(M_{opt})\right) \leq 2$.

(c) (15pt) Use this to devise an $O(kn^2)$ randomized algorithm (instead of the $O(k^2n^2)$ deterministic algorithm shown in class) that finds a multiple sequence alignment M_{rand}, such that $P\left(D(M_{rand})/D(M_{opt}) \leq 3\right) \geq 1/2$ for all input sets of sequences.

(d) (15pt) **Bonus:** For any constant $r > 2$ and $p < 1$, describe an $O(kn^2)$ randomized algorithm that finds a multiple sequence alignment $M_{r,p}$ such that $P\left(D(M_{r,p})/D(M_{opt}) \leq r\right) \geq p$.

4. **Suffix trees**

(a) (10pt) Build a series of strings for which the sum of lengths of the labels along all branches of the corresponding suffix trees are longer (by a non-constant factor) from the string length.

(b) (10pt) **Bonus:** Let us denote by m the length of a sequence, and by $T(m)$ the total length of the labels in its suffix tree. What is $T(m)$ for your construction? Clearly, for any sequence of strings, $T(m) = O(m^2)$. Prove or disprove that for any sequence of strings, $T(m) = o(m^2)$.

(c) (10pt) A palindrome is string W satisfying $W = W^R$. If W is a substring of S and W is a palindrome, we say it is a *maximal palindrome* if for every substring $U \neq W$ of S, if W is a substring of U, then U is not a palindrome.
Suppose S is of length m. Show that the number of maximal palindromes of S is $O(m)$, and give a linear time algorithm for finding all of them.

5. **And the golden star question is:**

The game of Chump! is a two players game, played on an chocolate bar of size M-by-N, with a *poisonous* leftmost/topmost square. Players take turns picking chocolate squares. In her (or his) turn, a player must pick one of the remaining squares, and eat it along with all the squares that are "below it and to its right". Using matrix-notation, the poisonous square is denoted by entry $(1,1)$, and the initial "state" of the brand new bar consists of the whole bar $\{(i,j) | 1 \leq i \leq M, 1 \leq j \leq N\}$. Picking the square (i_0, j_0) means that one has to eat all the remaining squares (i,j) for which *both* $i \geq i_0$ and $j \geq j_0$ hold. The player that eats the poisonous (topmost/leftmost) square dies in excruciating pains, and consequently loses the game. Picking $(1,1)$ to be your move kills you (in pains...), so a player who is non-suicidal will not play that move (unless she/he is forced to).

For example, if $M = 5$ and $N = 3$, then the initial game-position is

$$\begin{array}{ccc} X & X & X \\ X & X & X \\ X & X & X \\ X & X & X \\ X & X & X \end{array} .$$

The first player may choose to play $(5,3)$, in which case the chocolate bar (game) state becomes

$$\begin{array}{ccc} X & X & X \\ X & X & X \\ X & X & X \\ X & X & X \\ X & X & \end{array} ,$$

or she may choose to play $(2,2)$, which shrinks the chocolate bar to

$$\begin{array}{ccc} X & X & X \\ X & X & X \\ X & & \\ X & & \\ X & & \end{array} ,$$

and so on.

A *strategy* for a player can be thought of as a table (possibly huge) which tells the player which move to make in each given position (state of the game). A well known result from game theory states that in every two person game like the one we deal with (*i.e.* finite and deterministic), either the first player or the second one has a *winning* *strategy.* This theorem applies, for example, to the game of chess (but we are still a long way from knowing which player has a winning strategy, let alone *describing* such winning strategy).

(a) Suppose $N > 1$ and our chocolate bar is a square of size N-by-N. Describe (in words) a simple winning strategy for the first player in a game of N-by-N Chump!

(b*) Suppose $M, N > 1$ and our bar is a rectangle of size M-by-N. Prove the existence of winning strategy for the first player in an a game of M-by-N Chump!

The existence proof we know of is very simple, short, and extremely elegant. Unfortunately, the proof gives *no clue* what that winning strategy might be. Finding such winning strategy for the general case ($M \neq N$) will make a very impressive Ph.D. thesis, though not necessarily in computational biology...