Computational Genomics
© Benny Chor Eytan Ruppin, and Tomer Shlomi
School of Computer Science
Tel Aviv University
December 5, 2005

Computational Genomics: Assignment No. 2
due on December 22nd, 2005

General Guidelines:
This assignment is part of your final grade in the course. It should be done *independently* and individually, without any help from others. Duplicated or copied works will be given zero grade. Using articles, books, or web sites is perfectly acceptable as long as you include the reference in your relevant answer. You should bring a *hard copy* of the solution and hand it physically to Benny during the break in or before the lecture. Please make a serious effort to have a complete solution in **5 pages or less** (recall that a page, or amud in Hebrew, is one sided). And finally, as usual, the first to notify me of a real bug (not just a typo) in this assignment will get a 5 pts bonus.

1. (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).

2. **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.

3. **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) 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? **Bonus:** 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.

4. (10 pts) Show that the vertex cover problem on undirected graphs (VC) is polynomial time reducible to the vertex cover problem on undirected, *triangle free* graphs (Δ-VC).

5. (10 pts) Consider the reduction showed in class from Δ-VC to BIG maximum parsimony. Let u and v be two "reduction strings" (each encodes an edge in the original graph). Suppose they are both connected to w, an internal node in a labeled tree T. The node w is labeled by some string from $\{0,1\}^n$. Show that if u and v do not share a 1 in the same position, than either $d(u,w) \geq 2$ or $d(v,w) \geq 2$, where d is the number of changes.

6. (10 points) Give a small example of a (non-additive) distance matrix where NJ generates a tree with some negative edge length. The distance matrix should have positive entries (except on the diagonal), and be symmetric. However, it is not required to satisfy the triangle inequality.