Computational Genomics
© Benny Chor and Amos Tanay
School of Computer Science
Tel Aviv University
October 27, 2004

Computational Genomics: Assignment No. 1
due on November 10th, 2004

General Guidelines:
This assignment is part of your final grade in the course. It should be done *independently*, either individually or in *pairs*, 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.
Credit: Solve all items for 120 points + one golden star. Sections marked by (*) are harder, but you can accumulate the full 100% grade even without solving them.

1. **Dry Runs** (10 pts) In this problem we will experience two dry runs of the alignment algorithms (one local, one global). For each execution construct two short sequences S and T over the alphabet A,C,G,T, such that S and T are both of length 6, and both contain all four alphabet letters. The scoring function will give +2 for a match, -2 for a gap, and -3 for a mismatch.

   (a) Construct S and T such that their optimal global alignment is not unique, and show the execution of the global alignment algorithm over them: Draw the table, including values and pointers in each entry. In addition, write down two optimal (global) alignments.

   (b) Construct S and T such that their optimal local alignment is unique, and show the execution of the local alignment algorithm over them, as in (A).

2. **Finding Repeats** : In the problem of local alignment between two different sequences A and B, one has to find a pair of subsequences, one in A and the other in B, with maximum similarity. Suppose that we want to find a pair of subsequences *within* a sequence A with maximum similarity (it is also called the "optimal inexact repeat"). Argue why we cannot simply compute the local alignment between A and itself. Give an algorithm to solve the optimal inexact repeat problem for the following cases. Analyze the space complexity and running time of your solution.

   (a) The two subsequences may overlap. (10 pts)

   (b) Overlap between the subsequences is *not* allowed. (10 pts)

(c) The two subsequences must be adjacent in A, that is, $A = uw_lw_rv$ and w_l, w_r have the highest similarity over all such subsequences of A. This type of repeat is called **a tandem repeat** and it occurs very often in biological sequences. (10 pts)

(d) The same as the previous item, but we require that w_l and $(w_r^c)^R$ have the highest edit similarity over all such subsequences of A, where $(A^c)^R$ is the reverse complement of a DNA sequence A. This biological phenomenon is called an *inverted repeat*. (10 pts)

3. (10 pts) A is called a non-contiguous *supersequence* of B if B is a non-contiguous subsequence of A. Give a polynomial algorithm to find a *shortest* non-contiguous supersequence of both B_1 and B_2. Prove and analyze it.

4. (15 pts) Write a short "dry" c++/c/java/perl program implementing Hirshberg algorithm for global alignment in linear space. We do not expect this to run, just implement the following:

   - implement a function called hirshberg(string &s, string &t)
   - assume everything is OK, e.g., no need to check for system errors etc.
   - assume the function sigma(char s, char t) is given and calculate the distance between two characters (including gaps)
   - you should return the alignment in any way you like to (strings, vectors of positions, anything)
   - specify clearly what kind of output you generate

   Submit your printed program, in two columns documented format, make it as short and elegant as you can.

5. (20 pts) Show how Hirschberg's linear space technique can be used for the computation of a local alignment, and for the computation of a global alignment with an affine gap penalty (i.e. $g(k) = \alpha + k \times \beta$).

6. **Gap Costs**

   (a) In class we described a dynamic programming algorithm for pairwise sequence alignment under an affine gap penalty function $GapCost(k) = g + k * h$. Extend this approach to general monotone gap functions, where $GapCost(1) \geq 0$ and $GapCost(k+1) \geq GapCost(k)$. Describe the recurrence relations and the algorithm for computing an optimal alignment with these general gap penalties. Discuss your algorithm time and space complexity. (15 pts)

(b) (*) Can you improve your algorithm from the previous item when gap costs are known to be convex? Convex gap costs are functions with decreasing derivative. (10 pts)

7. **And the golden star question is:**
What is the *only* cure for a cat that was critically wounded by gun shots to his stomach? (Hint: Literary knowledge may be more useful here than biological or medical knowledge).

The facts regarding cats that are listed below are of general interest, though they may not necessarily help you in snitching the golden star. Additional "cat facts" can be found in http://www.hdw-inc.com/historyofcat.htm, while more serious facts can be found in http://home.ncifcrf.gov/ccr/lgd/comparative_genome/catgenome/index_n.asp – the homepage of the cat genome project. In zoological classification, cats belong to the Class : Mammalia (mammals - hair covered animals that suckle their young with breast milk), the Order: Carnivora (they are carnivores - they eat meat) and the Family: Felidae. Within this family there are three further subdivisions called genera (Panthera (cats that roar), Acinonyx (the Cheetah) and Felis (all other small cats)), and each genus contains individual species. A species of cats (or other animals) is a group that normally breeds and produces fertile off spring. (See http://www.all-science-fair-projects.com/science_fair_projects_encyclopedia/Felidae for the cats' and its close relatives' scientific classification.)

Note for *cat owners*: All cats – including domesticated species (Felis catus) – are obligate carnivores and they cannot survive without ingesting nutrients derived from animals. **Cats must never be fed an exclusively vegetarian ration!**