Computational Genomics
© Benny Chor and Amos Tanay
School of Computer Science
Tel Aviv University
December 8, 2004

## Computational Genomics: Assignment No. 3
## due on December 22th, 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 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 110 points + one golden star.

1. (40 points) Using Suffix Arrays: In this assignment you are asked to use a program that construct a suffix array given a string and to build a phylogeny using matching statistics averages between proteomes.

   - Directory /scratch/CGfall2004 contains 9 proteomes (.caa) files. These files were constructed by concatenation of all known protein sequences in several species. The directory also contains the program

     ```
     make_sa
     ```

     that generate a suffix array given a file. Running

     ```
     make_sa myfile
     ```

     will generate a binary file named myfile.array, containing one unsigned long for each suffix. You may try this on a short string where the outcome can be reviewed manually. This can also be run on your home Linux machine, but Windows will not interact with this code.

   - Write a short program that uses the suffix arrays constructed by the above program to compute the average matching statistics (as discussed in class) between each of the two proteomes. Your program should then use the distances to construct a phylogenetic tree using standard neighbor joining (employing either publicly available programs, something you code, or a careful "dry run"), and output the tree to a file.

   - Submit your program's code, the tree you obtained, and the distance matrix in muhomedir/cg/ex3/.

2. (20 points) Give an example of a (non-additive) distance matrix where NJ generates a tree with some negative edge length.

3. (20 points) In class, the NP hardness of *maximum parsimony* was shown by a reduction from *vertex cover*. An important ingredient in the proof was the observation that if we got a depth one subtree, where the leaves are labelled by "edge strings" such that every two leaves share one "1" position, then all the leaves share a common "1" position. This claim is "almost correct". It fails if the three leaves represent a triangle in the original graph. Your mission, if you choose to take it, is to fix this problem. There are a few paths to take. The simplest is to make sure the graph has no triangles to begin with. For that, you need to show that vertex cover on triangle free graphs is NP-complete, e.g. by a reduction from vertex cover.

   Thanks to Yaron Lev for pointing out this bug (and please, no violence of any sort against him.)

4. (30 points) You are analyzing the results of an ESP (extrasensory perception) experiment. You got Uri Geller and Oren Hakatan sitting on the two sides of a curtain where one (Uri) is writing down a binary sequence and the other tries to guess it. You thought of using mutual information to test the success of the two to read each other's mind. Out of $n$ trials, you got $n_{ij}$ cases where Uri thought of $i$ and Oren guessed $j$. Write down the mutual information as a function of the $n_{ij}$'s. Can you compute the probability of observing such mutual information or higher by chance? Suggest an efficient algorithm to compute this probability.

5. **And the golden star question is:** Where does a French signer, a Russian revolutionary activist and a foreign ministry official meet? (Let's see Google getting you out of this one)