Project 2 details

Below are more details about the several projects that were picked for project 2 in the class. Please let me know as soon as possible if the material indicated is unclear or if you'd like more details or more test data.

1. Overlapper
Write a program that performs the pairwise comparisons between the set of sequences provided as input to a shotgun sequence assembler and report which sequences overlap. This project will require you to use a combination of exact matching (e.g. k-mer hashing) and inexact matching (Smith-Waterman) techniques.

Additional details:

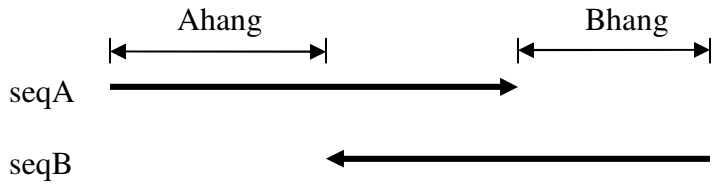
- Input should preferably be in AMOS format (amos.sourceforge.net), though .fasta files are acceptable.
- The broad outline of an overlapper program: (i) build a hash of all k-mers in the set of sequences to identify which sequences could overlap; (ii) run the local aligner from project 1 on each pair of sequences that could overlap in order to identify good alignments between them; (iii) if S-W indicates a proper "dove-tail" alignment, report the layout of the two sequences with respect to each other.

Sample data available at: http://www.cbcb.umd.edu/research/benchmark.shtml
Output format: for each pair of overlapping sequences you will have to report their relative orientation (assume first sequence is always "forward") as well as the amount each sequence "hangs" (extends) outside of the overlap region. If the sequences are oriented as below, the output should be:

seqA seqB I Ahang Bhang

where I(nnie) indicates the second sequence is in the opposite orientation than the first one. N(ormal) would indicate both sequences are in the same orientation.

Positive Ahang values indicate seqA starts before seqB, negative indicate seqB starts before seqA.



Ideally, the output should be represented in AMOS format as well.

6. RNA folding
Write a program that computes the secondary structure of an RNA

molecule and output this structure in a standard structure format. The RNA folding is a relatively straight-forward dynamic programming algorithm (left as an exercise in Pevzner's book).

The algorithm will be presented in class, but let me know if you are doing this project and I will give you a description of this algorithm.

Test data and examples of inputs/outputs are available at:
http://wilab.inha.ac.kr/pseudoviewer2/

7. Phylogenetic tree with NJ
Starting with a distance matrix between a set of sequences, compute and display the phylogenetic tree for these sequences using the neighbor-joining algorithm. The neighbor joining algorithm was described in class.

The output should be in the Newick file format:
http://evolution.genetics.washington.edu/phylip/newicktree.html

Sample data can be obtained from: http://rdp.cme.msu.edu/misc/resources.jsp . The DNA fasta files listed under Universally Conserved Genes are pre-aligned sequences, i.e. they all have the same length and corresponding characters align to each other. These can be used to compute the distance matrix, or you can use the dnadist program from the PHYLIP package (http://evolution.gs.washington.edu/phylip.html).

9. Multiple aligner
Implement a simple multiple alignment program, extending the Smith-Waterman algorithm implemented in project 1 to allow implementing the progressive alignment approach.

You will have to perform all-versus-all comparisons of the sequences using Smith-Waterman, then use either a tree method or the star method to decide on the order in which the sequences will be aligned together. Note that you will have to use global alignment settings for the dynamic programming.

Some sample data can be obtained at:
http://www.ebi.ac.uk/2can/tutorials/protein/clustalw1.html
This page is a tutorial on how to use CLUSTALW, thus using their examples will also allow you to assess the performance of your program compared to CLUSTALW.

10. Gene Finder
Implement a simple gene finder for bacteria, specifically find stop codons, identify in-frame starts, then evaluate the ORFs, based on codon preferences in the organism being analyzed.

Full genome sequences for bacteria, as well as "gold standard" annotations (to compare your predictions against) can be downloaded from http://cmr.tigr.org.

Codon usage tables can be found at http://www.kazusa.or.jp/codon/
Part of the report for this project should be a comparison between your results and the annotations found at the CMR.

15. Protein-protein interactions
Given a network of interactions between proteins (e.g. yeast 2-hybrid) compute several characteristics of this network, e.g. diameter, clustering coefficient, etc. Use these parameters to compare networks determined through different experimental techniques. This project will require you to build a graph data-structure then traverse it to compute several characteristics of the structure.

Data for this project can be downloaded from the Yeast Resource Center:
http://www.yeastrc.org/pdr/pages/download.jsp
Information on the type of analyses you can do on these networks is available in references 6 and 7 from: http://www.nd.edu/~networks/BioNetworks/
Also, a cool tool for playing with the networks is Cytoscape: http://www.cytoscape.org/

18. 2D Gel
A protein 2D gel separates proteins in two dimensions according to isoelectric point on one dimension and mass on the other. Given a genome and all the predicted proteins, generate a picture that simulates the image you would get from a 2D gel electrophoresis experiment.

The input to this program consists of all the proteins in a single bacterium, as downloaded, for example from: http://cmr.tigr.org.
For each protein you will have to compute the molecular weight, specifically add up the molecular weights of the amino-acids forming the protein (http://www.expasy.ch/tools/pscale/Molecularweight.html), then subtract the molecular weight of n-1 water molecules (18). You will also have to compute the isoelectric point - sample code that does this is available at: http://www.biophp.org/minitools/protein_properties/. You will also have to account for the relationship between molecular weight and distance of migration in the gel. For more details see: http://recomb2000.ims.u-tokyo.ac.jp/Posters/pdf/37.pdf.

A sample simulated 2D gel is available at http://rice.tigr.org/tigr-scripts/CMR2/Pseudo2DGel.spl?db_data_id=20129 for *Bacillus anthracis.*

19. GUI
Develop a GUI for one of the tools described above, or for the code you wrote in project 1.

If you choose to create a GUI for project 1, here are additional requirements. The GUI should allow you to load the appropriate files (the two sequences, or sets of sequences that are compared) and to set the parameters (matrix, gap open/extension penalties). Furthermore, the display should allow you to perform the alignment with different sets of parameters in order to compare the effect of the parameters on the alignment. For

example, you could have two result boxes, with two separate parameter choices and you provide a mechanism for viewing the results in both boxes in lock-step if the aligned sequences are too big to be shown on the screen in their entirety.

You could also experiment with different ways of representing the alignments.

20. Viterbi algorithm

For this project you will have to implement the viterbi algorithm to compute the most likely path of states in a given hidden markov model, given a string of observations. The models will be provided in the format used by the HMM implementation present at the following link: http://www.kanungo.us/software/umdhmm-v1.02.tar

Some examples on how this package can be used, in the context of speech processing, are described here:
http://www.umiacs.umd.edu/~resnik/nlstat_tutorial_summer1998/Lab_hmm.html