CS4234: Optimization Algorithms

MiniProject Ideas

Mini-Project 3: Computational Biology

There are a large number of fascinating and hard optimization problems in computational biology. We have increasing amounts of genome and other biological data available, and one of the big questions is what information we can derive from this data.

The following site has a variety of data sets available: `http://www.cs.utexas.edu/~phylo/datasets/`. You can find even more raw data at `http://www.ncbi.nlm.nih.gov/`. These data sets may be very difficult to work with, as they involve real data; you might find that in this case you want to work with synthetic data (i.e., data that you manufactor yourself), or a data set that has already been "cleaned up."

One of the most common questions is trying to organize species based on their genetic data: can we construct a "tree of life" that captures the relationships between species? A possible mini-project would be to explore these data sets, attempting to implement existing algorithms for constructing these "phylogenetic trees." For example:

- How do you measure the distance between two genome sequences? Find an efficient (and simple) algorithm for deciding distance between two species. How similar are two species? Can you differentiate two genes from the same species from two genes from different species?
- Imagine the goal is to build a tree with all the current species at the leaves, with the internal nodes representing possible (hypothetical) common ancestors. Assume, for now, that a genome is just a binary string and the distance between two species is the Hamming Distance (i.e., the number of mutations that need to occur to change from one to another).

  The goal is to build a tree—i.e., come up with a binary string for each internal node in the tree—that minimizes the cost of the tree. In many ways, finding this type of tree is really a Steiner Tree problem: find a set of binary strings (i.e., internal nodes in the tree) to minimize the sum of the distance along the edges of the tree. On the other hand, it is also very similar to a dynamic programming problem, finding the best way to transform one string into another. How would you build such a tree?
- What if you are given two sets of genes and told that one evolved into the other. Assume that evolution follows certain rules, i.e., is more likely to transform certain sequences than others. Choose one of the common models and see if you can fit it to existing data.

(Notice that each of these on its own may be quite difficult, depending on to what extent you are able to use real data, and to what extent you design your algorithms/programs to scale well and run efficiently. There have been entire PhD theses written on these problems!)

To learn more about this area, you might look at the following website: `http://www.comp.nus.edu.sg/~ksung/algo_in_bioinfo/`. This includes information on algorithms, as well as some programming projects and datasets.[1]

[1]Note: if you have taken a computational biology class before, your mini-project must be different from any previously completed assignments.