Project Ideas

Given below is a list of possible projects for you to work on. Some of the projects are part of ongoing work of some of the students. These projects have been marked with a [*] along with the students' names. Some projects are better defined than others. But they are all very interesting, and the only limitations are the amount of effort you put in and your creativity. If you wish to pick a project outside this list, please contact me as soon as possible. You should have picked something by Friday, October 4. Check with me on your choice to make sure that is no conflict with another group. Each group may have 1-2 members. In exceptional cases or for very structured projects (project 1), I will allow larger groups. Lot of the work is research-oriented and also result-oriented. I want to see some good results by the end of the semester. So start early. You are required to email me an update of your progress at least once every two weeks. Maintain a log file (or journal) containing your activities on this project containing: updates on your reading, progress on implementations and partial testing, ideas for future work, ideas that you may not be able to follow up, bug fixes, known current bugs in your code, organization of your program files and data files, etc.

At the end of the project, you will need to write a report (in pdf format). It must include a short summary of your project. State clearly the following: name & date, goals, hypotheses or assumptions, background with references and URLs, methods used (with references), what was implemented or achieved, summary of results, conclusions, possible future work, and URL describing your work.

Finally, prepare: (1) a handout to distribute to your classmates and (2) a 15-minute PowerPoint presentation of your work, (3) web page describing your project, and (4) a zip-compressed file containing your (commented) source code, data, results, report and your web page to be mailed to me. Your project should be completed and submitted by **November 15**. Your presentations will start from November 19.

Contact me for detailed information on the projects.

Traditional Bioinformatics

1. *Analyzing unknown DNA fragment from unspecified bacterium*: Ms. Einstein isolated a mutant bacterium that was resistant to increased amount of antibiotic X. She narrowed down the search for the gene(s) responsible for this behavior and sequenced a DNA fragment. She has now sought your help. For her sequence, figure out: CpG islands, ORFs, ribosome-binding sites, promoter regions, termination regions, inverted repeats that may serve as regulatory binding sites, restriction sites, genes and the corresponding proteins, hydrophobicity plots (& interpretations), homologous genes and proteins, secondary structures in the protein, protein motifs, at least 10 possible functional annotations, restriction sites, protein structures using some homology modeler, discrepancies in your information, and any other new things you can think of. Actually, you can have 5 different sequences. I suggest a 2 or 3 person team.
2. *Gene Ontology problem*: What can you infer about the function of one gene/protein in an organism using a database of known genes/proteins in another organism. How can you systematically compare the genes/proteins in two genomes?

3. *Analyze whole genomes*: If you are given the entire sequence of a new genome, what can you do with it? What information can you extract and what tools can you utilize?

Pattern Discovery

4. *New HTH motif detection algorithm*: GYM is an existing algorithm for HTH motif detection implemented by my students. There is now a modified algorithm. Implement the new algorithm.
5. *HTH motif discovery in complete genomes*: Run the existing program GYM on complete genomes that have been recently sequenced to locate all possible HTH motifs.
6. *Implement TEIRESIAS*: GYM is based on supervised pattern discovery techniques. TEIRESIAS is its unsupervised counterpart. The first part of TEIRESIAS has been implemented. Implement the second part of this algorithm.

Projects related to Protein Structure

7. *Geometric Hashing*: How to store structures in such a way that they can be retrieved quickly later on? Implement geometric hashing for this problem.
8. [*Milledge & Zheng] *Structure Pattern Discovery*: How to discover common patterns in a collection of protein structures? There are new algorithms for discovering structure patterns. Implement the algorithm by Jonassen *et al.*

Phylogenetic Analysis

9. [*Buendia] *HIV data analysis*: Study evolution of HIV using the large available databases.

Eco-informatics

10. [*Yang and Wang] *Analysis of microbes in soil samples*: Microbial ecologists are interested in knowing what microbes are present in a given soil sample. Sequencing all the microbes present in a sample is too time-consuming and expensive. A quick and dirty method is to use ALH profiles. Every microbe has a specific profile, and the composition of the microbes in the sample gives it a distinctive profile. Implement a method to compare profiles of samples and to classify them.

Problems on Microarrays

11. *Microarray data analysis*: Use SAM, SVMs and SOMs to analyze/classify microarray data. The Stanford Microarray Database is an enormous database of interesting microarray data.
12. [*Wu & Dai] *Pattern discovery in microarray data*: Use pattern discovery techniques to find patterns in temporal microarray data.

Miscellaneous Problems

13. *Phage Improvement problem*: Bacteriophages are viruses that attack bacteria, and it provides nature's own way to fight bacteria. However, bacteria have

mechanisms to destroy the phages by using their own restriction enzymes, which locate the restriction sites on the phage sequence. Can the phage DNA sequence be modified so as to eliminate (or at least minimize) the number of restriction sites? Implement an algorithm due to Skiena *et al.*

14. [* Wei] *Primer Design Problem*: Primers are short nucleotide sequences that are used as initiators for the PCR reaction. Given a set of (aligned) sequences find a pair of "good" primers.
15. [* Cazalis] *Probe Design Problem*: Probes are short nucleotide sequences that are used to check if a clone contains the complement of the probe sequence. If a clone contains the complement of a probe then we say that the probe is compatible with the clone. A probe that is compatible with clone A and not clone B can be used to differentiate between the two clones. Given a set of clones, and a large set of possible probes, and information on which probe is compatible with which clone, find the smallest subset of the probes that differentiates every pair of clones.
16. *Comparing Multiple Sequence Alignment Software*: A paper from 1998 compares several existing multiple alignment software programs on the basis of sensitivity and selectivity. Since that study is already 4-5 years old, repeat that study with your own possible improvements.