CSC 314, Final Project
Spring 2024

Bioinformatics is the study, development, and utilization of computational methods for storing, retrieving and analyzing biological data. The field of bioinformatics includes both the *development* of databases and tools for carrying out bioinformatics analyses and the *application* of these tools to answer important biological questions. I hope that you will leave this course with an appreciation for both the development and application of these tools.

For your Final Project, you will select an assignment related to either the development of a bioinformatics tool (i.e., a programming project), or the application of the databases and tools discussed during the semester to answer important biological questions. Your choice should be based on your interest and level of comfort with the project. You may work with a partner on the final project.

Final Project: select ONE assignment from either Option A, Option B, or Option C

Option A. *Bioinformatics programming project*

Write a bioinformatics program, in the language of your choice, that does one of the following. You must send me the source code for your program and I must be able to compile and run the program for you to receive credit. All source code used must be your own; libraries or modules may be used only with permission, unless indicated below.

1. *Open reading frame (ORF) finder*. A DNA or RNA sequence is read from a file. The program translates the entire sequence using all six possible reading frames. In addition, all open reading frames identified beginning with a start codon are highlighted (this will highlight all amino acids beginning with a start codon and ending with a stop codon, if found, or the end of the sequence). You may use packages such as *colorama* to highlight the open reading frames, but you may <u>not</u> use Biopython or other available packages for this assignment.

2. *Optimal pairwise alignment.* Write a program that finds the optimal local alignment between two sequences that are specified by the user. Your program should output the optimal local alignment and the optimal alignment score. If multiple alignments are optimal, only one optimal alignment needs to be displayed (though you are encouraged to output all alignments). Your alignment should use a scoring system where matches are worth 5 points, mismatches are worth -1 point, and there is a linear gap penalty of 4. Note: you may assume that the sequences are no more than 100 characters each.

3. *Prokaryotic gene prediction.* Write a 'simple' prokaryotic gene prediction program, following page 12 of the Gene Prediction notes, that identifies genes that have conserved promoter sequences, Shine-Delgarno sequences, and open reading frames coding for at least 100 amino acids. Note: you *may* use Biopython for this assignment. To handle mismatches, you should use the *regex* (https://pypi.org/project/regex/) module, which allows for mismatches in a regular expression. For example, suppose we want to find all codons that contain at least 2 adenines ('A' characters). This can be accomplished using the *regex* module and the regular expression below

   ```
   regex module:     '(AAA){s<=1}'
   ```

   which will match any *AAA* sequence that allows for up to 1 mismatch (the *s* is for *substitution*).
   I can provide sample data for this option if you would like it.

4. *Analysis of exon/intron boundaries*. Use the UCSC Table Browser to retrieve intron and exon sequences for the following genes in FASTA format: TP53, BRCA1, BRCA2, KDM6A. Use Biopython to read in the sequence data and output the number of times each nucleotide is observed at the first, second, third, and fourth position of each intron, and for each of the last 4 nucleotides of each intron. Note that you can count the number of times an item appears in a list using a *Counter*, which returns a dictionary (https://www.guru99.com/python-counter-collections-example.html). Write your results to a file, and then open them in Excel and construct a stacked bar graph showing the relative frequency of each nucleotide at each position.

5. *Viterbi algorithm.* Implement a *Hidden Markov Model* that finds the optimal state of hidden sequences (coins) that generates *heads, tails,* and *heads,* following the model on page 10-11 of the HMM notes. However, probabilities should be on the *log2* scale (import the *math* module, then use the *math.log2* function). Hint: using Python, you can create dictionaries for looking up transition and emission probabilities. For example, looking up '`FF`' in the dictionary would return the log of 0.90, which corresponds to $Pr(F_{i+1} | F_i)$, or the transition probability of selecting a Fair coin for the next coin when the current coin is Fair.

6. You may choose to develop another bioinformatics program, with my approval.

Option B. *Galaxy tutorial*

Complete one Galaxy tutorial from the list below, generate a link of your history, and share your completed results with me. You will also need to write a small report summarizing your results (around ½ of a page).

1. Which coding exon has the highest number of single nucleotide polymorphisms (SNPs) on human chromosome 22? Follow the tutorial to also answer this question for repeats. Then repeat both analyses using chromosome 21 (follow the steps to get this from the UCSC table browser). https://training.galaxyproject.org/training-material/topics/introduction/tutorials/galaxy-intro-101/tutorial.html
2. Biomarker candidate identification: https://training.galaxyproject.org/training-material/topics/proteomics/tutorials/biomarker_selection/tutorial.html#check-for-previous-detection-by-lc-msms-experiments.
3. Other tutorials may be used with permission

Option C. *Bioinformatics analysis*

If this option is selected, you will be given a gene to perform a bioinformatics analysis on. Your analysis may include (but is not limited to) the following tasks (the specific requirements will depend on the assigned gene).

- Generate a visualization of the gene (using UCSC genome browser), identifying the positions of its introns and exons, and identifying the chromosome the gene is on.
- Identify related gene/protein sequences from NCBI's GenBank and protein databases.
- Identify orthologs using NCBI's Homologene database.
- ~~Retrieve gene information using NCBI's Gene database.~~
- Use BLAST to identify similar proteins in other species, and to determine whether the protein has any conserved domains. Based on the domain information, describe the function of the gene.
- ~~Is the gene differentially expressed (I will tell you what experiments (i.e., GEO series) to look at)?~~

- Are mutations in the gene associated with any diseases or conditions?
- Understanding inheritance patterns related to the gene (e.g., Punnett Square questions)
- Understanding domains in the protein

Important Dates

Project selection (Blackboard): **Tuesday, April 30th, 5:00 PM** (10 point penalty if not completed)
Final Project: **Thursday, May 9th by 4:00 PM**

Additional Information
See the accompanying rubric for additional information.