CSC 314, Final Project
Fall 2020

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 of 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 or Option B.

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 that are identified are highlighted (this will highlight all amino acids beginning with a start codon and ending with a stop codon; if a stop codon is not present, then the rest of the sequence is highlighted). 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. You may also choose to use the *regex* (https://pypi.org/project/regex/) module (different than what we used in class). The *regex* module 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}'
   ```

   will match any *AAA* sequence that allows for up to 1 mismatch (the *s* is for *substitution*).

   Using the *re* module (what we used in class), the appropriate regular expression is given by

```
re module:     '(AA.)|(A.A)|(.AA)'
```

which says to match the following:

1. *AA* followed by any character, OR
2. an *A* followed by any character, followed by another *A* OR
3. any character followed by an *AA*

4. *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 0.90, which is $Pr(F_{i+1}|F_i)$, or the transition probability of Fair coin → Fair coin.

5. *BLAST.* Using Biopython, write a script that BLASTs the sequences stored in *mRNA1.fasta* and *mRNA2.fasta* from Lab #11. In each case, for the top two hits of each BLAST, output the organism, accession number, identity, e-value, and alignment of the top hit for each sequence. Biopython's BLAST is describe here: http://biopython.org/DIST/docs/tutorial/Tutorial.html#sec119. Note: when using BioPython's BLAST, you should save a local copy of your results (as described in the above link). You should create two Notebooks: one for carrying out the BLAST and saving the results; and one for parsing the results from a local file.

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

Option B. ***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).

- Draw a graph of the gene (using UCSC genome browser), labeling 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 similar proteins in other species, and 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?
- What secondary structure characteristics are present in the protein?

Important Dates

By 5:00 PM, November 30, sign up for your project on Piazza, in the thread provided. This will be worth 1% of your assignment grade.

Additional Information
See the accompanying rubric for additional information.