CISC471 - Homework 3

Due Friday, February 26, 2021 at 5:00PM EST

Submission

Please submit the programming and theory as multiple files in OnQ. The programming component should be submitted as a Python3 program with the main file `main.py` and any other python files you wish. The theory component should be submitted as a single PDF file. These should be done in groups of either one or two people. In your PDF file, you *must* have each student name and student number present at the beginning of the file.

Failure to follow these instructions will result in a mark of zero. Late assignments are not accepted and will result in a mark of zero. Only the best 4 of 6 assignments will be used for your final grade.
E.g. Directory structure:

- OnQ Submission Directory:
  - hw3.pdf
  - main.py
  - ...

1 Programming - 6 points

Write a program in Python and verify that it works on the sample data (using the on-line Rosalind platform). For each problem, add three unit tests using the Rosalind sample data, and some of your own. There must be at least one positive and one negative unit test.

Problems

1. Implement Greedy Motif Search `http://rosalind.info/problems/ba2d/` in greedy.py with unit tests in greedy_unit.py. **/ 2**
2. Implement Randomized Motif Search `http://rosalind.info/problems/ba2f/` in randomized.py with unit tests in randomized_unit.py. **/ 2**
3. Implement Gibbs Sampler `http://rosalind.info/problems/ba2g/` in gibbs.py with unit tests in gibbs_unit.py. **/ 2**

2 Research - 7 points

In Section 1, you programmed 3 algorithms that find motifs. In this section you will prepare a report comparing the three methods for different parameters.

The report must be a LaTeX-typeset PDF document with the template ACM conference proceedings template located here. It's easiest to make an Overleaf account and use the template at the link.

Your report must have:

1. An introduction and motivation Section describing the motif finding problem in your own words. / **1**
2. A Section for each of the three algorithms, describing: / **3**
   (a) Why you think your algorithm is correct (whether you program worked on the sample data or not).
   (b) Provide an estimate of the time and space complexity of your algorithm.
3. A Section for comparing the three algorithms for
   (a) Time complexity for different parameters (e.g. a graph)
   (b) Space complexity for different parameters (e.g. a graph)
   (c) Another comparison of your choosing

   in which you will describe your process and reasoning for choosing these parameters. You should modify your Python programs to generate real outputs for different data. / **1**
4. A Section for Discussion of your results. / **1**
5. A Section for Conclusion for your overall report. Make a convincing argument for *why* you would want to use different algorithms under different scenarios. / **1**

2.1 Constraints and Grading

Each section is worth one point. Your report grade will be based on grammar and spelling, presentation, clarity, and adherence to the instructions. Marks will be deducted for a lack of these items. A handy link for checking clarity is `http://hemingwayapp.com/`.

2.1.1 Constraints

1. Your report must not exceed three pages.
2. Your report graphs must be done in LaTeX (i.e. not images). Use packages like pgfplots, tikz, etc...
3. Provide error bars (or a different way to visualize error) where appropriate, e.g. if randomized do multiple trials per data point (state how many) to use the mean/stdev.