CMEE Masters: Computing Coursework Assessment

Assignment Objectives: To work on a series of computing/programming exercises and problems in a coherent, modular, reproducible workflow under version control.

Note that:

- *The overall assessment will typically have significantly lesser marks than a simple weighted average of each week's points because the overall assessment is based on not just the "Computing Coursework Assessment Criteria", but also the the "Marking Criteria for Exams, Essays and Coursework". Both sets of marking criteria are in the Assessment Appendix of the online TheMulQuaBio notes and git repository.*
- *In your 1:1 post-assessment feedback session, we will discuss where you gained or lost marks, and what you could have improved further. To the extent possible, please come with questions about specific scripts based upon the overall and weekly feedback you have received. This may require you to compare your code with the solution code in many cases.*

Student's Name: Sarah Dobson

1 Specific feedback

1.1 The Good (what you did well!)

1. Found all the core CMEE weekly directories in your parent directory.
2. Your code and structure are logical and tidy.
3. Your Git repo size when I checked week 7 was about 5 MB – nicely compact! This suggests you correctly suppressed unnecessary files from version control, and did not commit excessively. It could also mean that you did not commit enough, and/or somehow along the the way lost parts of your git history – but we don't check these possibilities!
4. You had an overall readme file, as well as one within each week. The Readmes were clear, and comprehensive, even including info like dependencies and language version numbers. Good job! Also check out this resource: `https://github.com/jehna/readme-best-practices`. As you become a seasoned programmer, you will learn to make the readme file descriptions even more informative yet succinct.
5. You have made reasonable efforts to produce modular Python code. This is good - Python is intended to work this way so the more practise you can get at it the better!
6. Overall a good job on the coding. Some slightly careless errors (see below) remain, but they are easily fixed with a little more attention to detail. Well done also for remembering all the docstrings.
7. Your Groupwork practicals were all in order, and your group did well in collaborating on it. More feedback on this in the 1:1 sessions.

1.2 The Bad (errors, missing files, etc)

1. `align_seqs_better.py` and `align_seqs_fasta.py` tried to load data from data/407228326.fasta but this file was not included in your repo.
2. `cfexercises1.py` throws an error due to a missing bracket at the end of line 76 (in the main() function). Line 77 is also missing a second closing bracket.
3. `profileme2.py` throws an error due to the docstring of my_join() being indented by one space more than the rest of the function. This also leads to `timeitme2.py` throwing an error when it imports that function.

1.3 The Ugly (niggling issues like commenting, cosmetics, complexity of code, etc)

1. You had a .gitignore to control which files were under version control, which is good, though you might also have opted to make week-specific exclusions. You will likely find this useful: `https://www.gitignore.io`.

2 Overall Assessment

Overall a very good job. Your code is clean, logical and well documented. Although there were some errors left in your code, they do not correspond to significant lapses in your understanding of Python/R, but rather seem to just be slightly careless syntactical mistakes. This should improve with practise (and with judicious testing!). You should be well placed to improve as a programmer based on this foundation.

Provisional Mark: 71%

Signed: Alexander Kier Christensen & Samraat Pawar

March 23, 2022