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:

- *All script/code files, errors and other info mentioned below are in the weekly log/feedback files.*
- *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: Ruth Keane

1 Specific feedback

1.1 The Good (what you did well!)

1. Found all the expected weekly directories in your parent directory.
2. Very neat, clean project organization and code.
3. Your Git repo size when I checked week 7 was about 120 MB — an OK size (but could have been smaller), suggesting you did not keep unnecessary binary files under VC, and that you 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 I won't check these possibilities!
4. You had a .gitignore throughout, with meaningful exclusions specific to certain weeks. Good, but you can fine tune the exclusions further. You will likely find this useful: `https://www.gitignore.io`.
5. Nice job with the coding overall. Good attention to detail.
6. You did all extra credit Qs - fantastic.
7. Great job with the shell scripts - you made them more robust, providing a message if necessary inputs were not provided. Indeed, in general, it is a good idea to add some input checks and return a meaningful message with error for utility scripts like these, especially in case somebody else uses it. I had not asked for these explicitly, but was hoping this would be something you would arrive at yourself after gaining some experience with coding and revisiting your code. But most students don't get to this, great work!

8. The Autocorrelation practical code and report were good. You plotted the correlation pattern, and also the histogram of the permuted correlation coefficients - good. The interpretation of the results was good, but you could have gone a bit deeper. Why fit a regression line?

9. Scripts from Weeks 4, 5 & 6 were not part of the assessment, but you kept these weeks organized - good!

1.2 The Bad (errors, missing files, etc)

1. No errors as far I could tell!

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

1. You had an overall Readme file, but it did not contain anything useful (like a description of the overall project structure and usage).

2. The weekly Readmes could have been more succinct. In these Readmes you could have included the language and dependencies requirements. 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.

3. Commenting could be improved – too verbose at times, and inadequate/missing in others. It will get better with coding experience.

4. In many places, especially early weeks (e.g., UNIX), you could have broken the description of certain complex commands or code lines into key components using a comment.

5. The occasional missing Python docstring

6. Also, as much as possible, it is a good idea to write scripts to be self-sufficient / modular. For example, align_seqs.py was nicely done, but it could have been written to be a full blown module that also take external inputs optionally (though I did not ask for it specifically). Compare with the solution.

7. Please do compare as many of your solutions with the ones I have given (e.g., Unix-Prac1.txt, using_os.py) as possible. There are simpler ways to solve some of them, especially the last one, and in general it will be insightful to see how the same code/solution can be written/found. In particular:

   (a) using_os.py: the script could have provided some more meaningful output to screen.

   (b) You did a good job with lc1.py, lc2.py, dictionary.py, and tuple.py, but if you compare with the solutions on the repo, you will notice that you could have make them produce better-formatted output.

2 Overall Assessment

You did an outstanding job overall. I was impressed by your efforts to understand as many details of the programming languages and coding as possible. You clearly like coding!

It was a tough set of weeks, but I believe your hard work in them has given you a great start towards further training, a quantitative masters dissertation, and ultimately a career in quantitative biology!

Provisional Mark: 93

Signed: Samraat Pawar

March 8, 2020