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: Francesca Covell

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 organisation and code are generally neat (aside from a few extraneous files) and logical.
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 work! 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. Very good job with the coding overall. Good attention to detail, almost error-free and not too many warnings!
6. Your python code is generally appropriately modular, which is good to see and a practise which will take you far in this language! 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. Although your file organisation is generally good, some of your script inputs/outputs (e.g. exampleTIF.png, exampleTIF.tif) have been saved to your Week1/code/ subdirectory, rather than placed into the relevant data or results subdirectory. Similarly, you appear to have some near-empty extra non-script files (test.txt, test2.txt, test3.txt) sitting in Week1/code. This is messy, and could make it hard for you and/or users of your code to find things within your project structure, particularly when it comes to more complex projects.

2. `LV1.py` threw an IndentationError because the docstring for the function `dCR_dt` was not indented. It's important to be aware that in Python, docstring indentation can cause errors just like the indentation of any other expression!

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. Commenting could be improved – you occasionally (e.g. `csvtospace.sh`) have put block comments in your code explaining steps that the code needs to carry out. This is a reasonable idea but such comments should be split up and distributed over the lines of code that actually execute these steps, rather than being placed all together in a block separate from the code itself. Similarly (e.g. `UnixPrac1.txt`) you have occasionally left reminders to yourself for things to change. This can be helpful while working on code but finalised/submitted code should either have the changes implemented or such comments removed, to let readers/users focus on comments that explain the code logic/workflow.

2 Overall Assessment

Overall a great job, with almost no errors and only minor improvements to suggest. Just be sure to keep files in the relevant directories and be a little more conscientious in your commenting. Otherwise, great work!

Provisional Mark: 72%

Signed: Alexander Kier Christensen & Samraat Pawar

March 23, 2022