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: An Nguyen

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 clean and logically structured.
3. Your Git repo size when I checked week 7 was about 7 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 the weekly readmes, were all present and correct. They were generally comprehensive, and even included requirements and version numbers. Excellent! 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. Your coding is overall excellent. Minimal and minor errors only. Well done! In particular, your Python is generally nicely modularised, which is good Pythonic practise, and well done for remembering all the docstrings.
6. 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. Your commenting is generally very good - informative and non-excessive. However there were a few scripts in which you appear to have forgotten to comment almost entirely (e.g. `TAutoCorr.R` and `Vectorize1.R`). It is generally recommended that you comment "as you

go" when writing code, rather than trying to remember to come back after a script is complete to add comments. This approach has multiple benefits – for example you won't end up with any completely uncommented scripts, and you won't have had time to forget what a particular line(s) of code were doing!

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

1. You had a project-wide .gitignore, which is good, but you can fine tune the exclusions further, for example with exclusions specific to certain weeks. You will likely find this useful: `https://www.gitignore.io`.

2 Overall Assessment

Overall an excellent job. Nearly error-free code, generally solid commenting (with some minor exceptions noted above) and documentation. You have developed a solid coding foundation that should stand you in excellent stead from here onwards.

Provisional Mark: 80%

Signed: Alexander Kier Christensen & Samraat Pawar

March 23, 2022