CMEE Coursework Feedback 2023/24

Note:

> - 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: *Hongyuan Guo*

The Good

- I found all the required weekly folders, and your repo was generally well organised.
- Your repo was also of a reasonable size, though you didn't need to push your HPC outputs to github.
- I was impressed with your solution to `csvtospace.sh`, using `sed`. This is a clever use of inbuilt bash utilities!
- Generally speaking I liked your use of the uniqueness property of the set object. This allowed you to create very efficient code.
- Your `align_seqs.py` solution is really very good. Nice and modular, though does contain some hard-coded paths that would not work on my machine.
- Your use of tuple unpacking in `lc2.py` is very good. This is a good way to write readable python code.
- Nice use of mapply in `TreeHeight.R`

The Bad

- You lacked the `FirstBiblio.bib` file from your Week 1 directory. The `FirstExample.tex` file will not compile without it.

- Your Week 3 code directory was inside another subdirectory which stopped the automarking script from picking it up. Be careful to keep your folder structure consistent!
- You had no `.gitignore` files, for the repo or for individual week folders. This meant that files like `__pycache__` and `.DS_Store` ended up in your code folders when they could easily have been ignored.
- Generally you have a tendency to use hard-coded paths rather than relative paths (such as `../data/input.csv`). This makes your code less portable and harder to use for other people (see `basic_io.R`).

The Ugly

(pedantic code style bits and other notes)

- I would have preferred a few more comments. Occasionally it was hard to make my way through your code (such as in `PP_Regress.R`). Generally though this was pretty good.
- You also should include shebang lines at the top of *every* standalone script.
- It is good practice to check to see whether files are present before deleting them, for example `[ -e *.aux ] && rm *.aux`

Mark

Provisional mark: *79*