

CMEE Masters: Computing Coursework Assessment

Note that:

All script/code errors and other info mentioned below are in the weekly log files

In the weekly feedback/assessments, please compare with the solutions whenever needed to see why I might have taken off points for a particular exercise/script or code file. We can then discuss these in your 1:1 post-assessment feedback session.

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

Student's Name: Eva Linehan

Overall Project workflow

Found all the expected weekly directories in your parent directory.

You had a .gitignore throughout, but could have had more exclusions specific to certain weeks – great. You will likely find this useful: <https://www.gitignore.io>.

You had a readme file for the overall project, and then within each week, another one – good. The weekly readmes were clear, including a description of what the overall project structure is – good. However, you should have listed what the language and dependencies requirements are, and no need to list stuff in sandbox and data – after all, the documentation (e.g., docstring) of each script would ideally be explicit about what sorts of data are required. Check out this resource: <https://github.com/jehna/readme-best-practices>. Also, showing the directory tree is handy, but not totally necessary. As you become a experienced programmer, you will learn to make the readme file descriptions more informative yet succinct.

Your Git repo size when I checked week 7 was about 9.39 MB — a good size, suggesting you did not keep 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!

You had a gitignore in sub-directories in multiple weeks; this is not necessary, as all file/directory patterns in sub-directories can be excluded from a single .gitignore in the parent directory of the repository. See <https://git-scm.com/docs/gitignore> and <https://labs.consol.de/development/git/2017/02/22/gitignore.html> : -2.5pts

WEEK 1

Found directories Data, Sandbox, Code

Found 12 code files: ConcatenateTwoFiles.sh, CountLines.sh, variables.sh, tiff2png.sh, csvtospace.sh, CompileLateX.sh, FirstExample.tex, MyExampleScript.sh, FirstBiblio.bib, UnixPrac1.txt, tabtocsv.sh, boilerplate.sh

UnixPrac1.txt was fine. Each solution was described in a comment, great. You also described the key components of the unix command - great! Compare with the solutions, especially the last one.

csvtospace.sh was fine. You could have made it give (with a message) if no input csv file was provided (something I had not asked for, of course). In general, it is a good idea to add some input checks and return a meaningful message with error for utility files like this, especially in case somebody else uses it. Similar comment for other shell scripts.

Points for this week: 100

WEEK 2

Found the Code, Sandbox, Data, Results directories

lc2.py, boilerplate.py, basic_csv.py, cfexercises2.py, sys.argv.py, align_seqs_better.py, dictionary.py, debugme.py, scope.py, cfexercises1.py, tuple.py, basic_io.py, lc1.py, oaks_debugme.py, oaks.py, loops.py, using_name.py, align_seqs.py, align_seqs_fasta.py, control_flow.py, test_control_flow.py

Found no extra files; great!

lc1.py, lc2.py, dictionary.py, tuple.py were all nicely done. They could have produced slightly better-formatted output – Compare with the solution on the repo; -0.5 pt each.

align_seqs.py was nicely done. You could have written it as a self-sufficient script that could also take external inputs optionally (though I did not ask for it specifically). Compare with the solution.

You did align_seqs_fasta.py and align_seqs_better.py correctly — so +5 extra credit pts.

All other scripts were fine – great job.

Points for this week: 100

WEEK 3

Found directories Practicals, Code, Data, Results

Found 34 code files: browse.R, PP_Regress.R, Vectorize2.py, apply1.R, sample.R, run_get_TreeHeight.sh, get_TreeHeight.py, Vectorize.sh, KeyWestResults.tex, boilerplate.R, TreeHeight.R, PP_Lattice.R, sample1.R, next.R, Maps.R, Girko.R, Vectorize1.R, SQLinR.R, break.R, plotLin.R, basic_io.R, Vectorize1.py, try.R, apply2.R, Inclass_plotting_exercise.R, get_TreeHeight.R, TAutoCorr.R, Vectorize2.R, DataWrangTidy.R, preallocate.R, PP_Regress_loc.R, DataWrang.R, MyBars.R, control.R

MyBars.R gave an error: -5pts

Vectorize1.R was fine.

Vectorize2.R was fine, nice job — compare with the solution.

PP_Regress.R: fine. Also have a look at my solution.

TAutoCorr.R was OK, except you resampled with replacement (compare with solution). The report was OK, but you did not report the p-value (which would actually be somewhat wrong because of the resampling problem). Glad you plotted the histogram of the permuted correlation coefficients. You could also have plotted the correlation pattern itself. Some more interpretation of the results would have been nice. -5pts Also, you could have kept the writeup in a completely separate directory

You did the Mapping (good description of the data and potential biases), pp_regress_loc , and Vectorize extra credit questions correctly – +10 pts.

Points for this week: 100 pts

WEEKS 4, 5 & 6

Not assessed.

WEEK 7

Found directories Code, Data, and Results

Found a README

Found 18 code files: TestR.py, LV3.py, profileme2.py, timeitme.py, blackbirds.py, regex.py, TestR.R, profileme.py, LV4.py, fmr.R, using_os.py, LV1.py, DrawFW.py, subprocessstutorial.py, Nets.R, run_fmr.R.py, LV2.py, CallScripts.sh

Found the following extra files: .RData, .Rhistory: 0.5 pt deducted per extra file

MyFirstJupyterNb.ipynb missing: -10 pts

DrawFW.py gave an error: -5pts

The two LV* scripts with profiling were fine. You also did the LV3-4 extra credits: +5pts . But compare with the solutions, especially for lv4.py.

using_os.py is OK, but compare with the solution. The script could have provided some more meaningful output to screen. -2pts

blackbirds.py was fine, including nicely-formatted output.

Points for this week: 87

Overall Assessment

You did an excellent job overall, including practically all extra credit Qs!

Neat, clean project organization and code, and almost no errors. I was impressed by your efforts to understand as many details of the programming languages and coding as possible.

Overall, as this is the first time you have done programming in a heady mix of UNIX, Python, & R with a sprinkling of \LaTeX and git, you did very, very well! In particular, you seem to be well on your way to becoming comfortable with both Python and R.

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: 83

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.

We will discuss where you gained or lost marks, and what you could have improved further in your 1:1 post-assessment feedback session. To the extent possible, please come with questions about specific scripts based upon the feedback you have received. This may require you to compare your code with the solution code in many cases.

Signed: Samraat Pawar

January 18, 2019