Data Science and machine learning for the biosciences
– Assessment and feedback.

Assessment
There are two components to the assessment for this unit:

1) A short group project on day 3, including a verbal presentation to the whole cohort and to which all group members will need to contribute (30%). Assessment to take place: Afternoon of Wed 2nd December.

Requirements of presentation:
Time – aim for ~5 mins in total. Please introduce yourselves and tell us what biological problem the code was written to solve, talk us through your code and show us a demo including the output produced. You can address any problems with the code if it isn't working quite as you envisaged. All group members should participate in the presentation.

Feedback. You will receive feedback on your presentation using the following form:

	Excellent	Good	Room for improvement	Comment
Explained the purpose of the code written				
Showed how the code was implemented				
Demonstrated working code and clear output				
Group enthusiasm				

2) From the course handbook: *An individual short project, involving development of simple software for elementary analysis of a large data set from their area of doctoral research (70%) (of which 50% is submission of a functioning code along with a documented log of debugging steps and 50% for a short, written summary of the code and the main outcomes from your analysis).*

What you need to do:
a) Upload the working, annotated code, a README file and sample data to GITHUB your code annotation will count as the minimal record of debugging steps).
b) Write a brief report (maximum 2 pages, 11 point arial font, including any figures and references). You must include the URL of your GitHub page in your report.

IF YOU NEED TO KEEP YOUT GITHUB PRIVATE, MAKE A NOTE OF THIS IN YOUR REPORT AND TAKE A SCREEN SHOT TO APPEND TO YOUR REPORT TO SHOW THE GitHub WAS CREATED.

c) Submit a zip file containing all of the files (a & b) to blackboard.

Please note that you must provide working code and (where necessary) test data for us to run it on in order to pass this component.
An example of a suitable README file is attached below. This needs to provide us with the information necessary to understand what your code is, how to run it and what the output should look like.

Submission deadline: 2pm, Wed 13th January.

Getting help with your work:

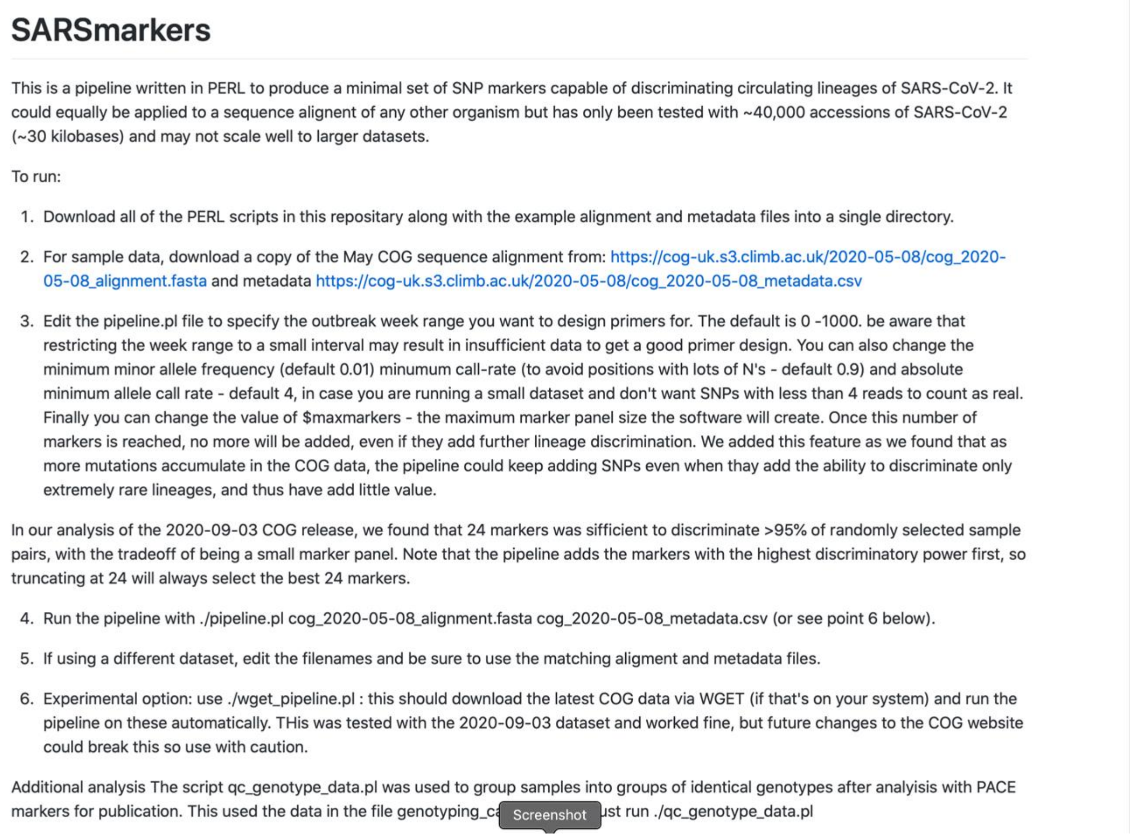
1) Check the documentation for any function they are trying to use
2) Paste any output errors into google so they can (hopefully) find answers there.
3) email ask-rse@bristol.ac.uk if the above fail. Remember to explain succinctly what you are trying to do and what the issue is – paste in code and error messages.

Feedback. You will receive feedback based on the form below:

	Excellent	Good	Room for improvement	Comment
Explained the purpose of the code written				
Code is properly annotated				
Sample data supplied or linked if required.				
Code runs without error				
Code produces output which can be understood based on information in the README file				

Unlike you other units, this unit is a simple pass/fail so you will not receive a grade, but you will need to achieve the 50-59 marking descriptor on the marking scale to pass each assessment (please refer to the 'marking scale' section in the SWBio DTP Taught handbook for further information).

Example of a minimal README file from GitHub. Note that it introduces the purpose of the code, provides links to the data required to run it and gives a clear command line example to show us how to make it work.



SARSmarkers

This is a pipeline written in PERL to produce a minimal set of SNP markers capable of discriminating circulating lineages of SARS-CoV-2. It could equally be applied to a sequence alignent of any other organism but has only been tested with ~40,000 accessions of SARS-CoV-2 (~30 kilobases) and may not scale well to larger datasets.

To run:

1. Download all of the PERL scripts in this repositary along with the example alignment and metadata files into a single directory.
2. For sample data, download a copy of the May COG sequence alignment from: https://cog-uk.s3.climb.ac.uk/2020-05-08/cog_2020-05-08_alignment.fasta and metadata https://cog-uk.s3.climb.ac.uk/2020-05-08/cog_2020-05-08_metadata.csv
3. Edit the pipeline.pl file to specify the outbreak week range you want to design primers for. The default is 0 -1000. be aware that restricting the week range to a small interval may result in insufficient data to get a good primer design. You can also change the minimum minor allele frequency (default 0.01) minumum call-rate (to avoid positions with lots of N's - default 0.9) and absolute minimum allele call rate - default 4, in case you are running a small dataset and don't want SNPs with less than 4 reads to count as real. Finally you can change the value of $maxmarkers - the maximum marker panel size the software will create. Once this number of markers is reached, no more will be added, even if they add further lineage discrimination. We added this feature as we found that as more mutations accumulate in the COG data, the pipeline could keep adding SNPs even when thay add the ability to discriminate only extremely rare lineages, and thus have add little value.

In our analysis of the 2020-09-03 COG release, we found that 24 markers was sifficient to discriminate >95% of randomly selected sample pairs, with the tradeoff of being a small marker panel. Note that the pipeline adds the markers with the highest discriminatory power first, so truncating at 24 will always select the best 24 markers.

4. Run the pipeline with ./pipeline.pl cog_2020-05-08_alignment.fasta cog_2020-05-08_metadata.csv (or see point 6 below).
5. If using a different dataset, edit the filenames and be sure to use the matching aligment and metadata files.
6. Experimental option: use ./wget_pipeline.pl : this should download the latest COG data via WGET (if that's on your system) and run the pipeline on these automatically. THis was tested with the 2020-09-03 dataset and worked fine, but future changes to the COG website could break this so use with caution.

Additional analysis The script qc_genotype_data.pl was used to group samples into groups of identical genotypes after analyisis with PACE markers for publication. This used the data in the file genotyping_ca ust run ./qc_genotype_data.pl