Miniproject Workplan

Objective:

The general question you will address is: What mathematical model best fits an empirical dataset?

The Project

* It should employ all the biological computing tools you have learned so far: shell (bash) scripting, git, LaTeX, R, and Python. Using these tools, you will build a workflow that starts with the data and ends with a written report (in LaTeX). How you choose the different tools (e.g., Python vs R) is your choice. This is part of the assessment.
* At least two different models (hypotheses) must be fitted to the data. The models should be fitted and selected using an appropriate method. Specifically, irrespective of the problem/dataset you choose (see below), use Nonlinear Least Squares (NLLS) to fit ≥2 alternative models to data, followed by model selection using AIC and BIC (read the Johnson and Omland 2005 paper). You may choose additional means for model comparison/selection beyond these.\*
* The project should be fully reproducible. Write a script that "glues" the workflow together and runs it, from data processing to model fitting to plotting (e.g., in R) to compilation of the LaTeX written report (More detailed instructions on report below). Look back at the TheMulQuaBio to see how you would run the different components. For example, we have covered how to run R and compile LATEX using the subprocess module in Python. The assessor should be able to run just this script to get everything to work without errors. Use Python or to write this main script. If using bash, call it run\_MiniProject.sh and if using Python, called it run\_MiniProject.py.

Marking Criteria

* *Equal weightage will be given to the code+workflow and writeup components — each component marked to a max of 100 pts and then rescaled to a single mark / 100 using equal weightage*
* The assessor will be looking for the following while assessing your submission:
* A well-organized project where code, results, data, etc., are easy to locate, inspect, and use. In the project's README also include: Any dependencies or special packages the user/marker should be aware of, what each package you used is for, version of each language used
* A project that runs smoothly, without any errors once the appropriate script (i.e., run\_MiniProject.py or run\_MiniProject.sh) is called.
* A report that contains all the components indicated above in "The Report" subsection, with some original thought and synthesis in the **Introduction** and **Discussion** sections.
* Quality of the presentation of the graphics and tables in your report, as well as any plots showing model fits to the data.
* The marking criteria you may refer to is the [summative marking criteria](https://nbviewer.jupyter.org/github/mhasoba/TheMulQuaBio/blob/master/notebooks/MARKING_CRITERIA.pdf).

Population Growth

The Question

How well do different mathematical models, e.g., based upon population growth (mechanistic) theory vs. phenomenological ones, fit to functional responses data across species?

Fluctuations in the abundance (density) of single populations may play a crucial role in ecosystem dynamics and emergent functional characteristics, such as rates of carbon fixation or disease transmission. A population grows exponentially while its abundance is low and resources are not limiting (the Malthusian principle). This growth then slows and eventually stops as resources become limiting. There may also be a time lag before the population growth really takes off at the start. We will focus on microbial (specifically, bacterial) growth rates. Bacterial growth in batch culture follows a distinct set of phases; lag phase, exponential phase and stationary phase. During the lag phase a suite of transcriptional machinery is activated, including genes involved in nutrient uptake and metabolic changes, as bacteria prepare for growth. During the exponential growth phase, bacteria divide at a constant rate, the population doubling with each generation. When the carrying capacity of the media is reached, growth slows and the number of cells in the culture stabilises, beginning the stationary phase. Traditionally, microbial growth rates were measured by plotting cell numbers or culture density against time on a semi-log graph and fitting a straight line through the exponential growth phase – the slope of the line gives the maximum growth rate (rmax). Models have since been developed which we can use to describe the whole sigmoidal bacterial growth curve.

The Data

The dataset is called LogisticGrowthData.csv. It contains measurements of change in biomass or number of cells of microbes over time. These data were collected through lab experiments across the world. The field names are defined in a file called LogisticGrowthMetaData.csv, also in the data directory. The two main fields of interest are PopBio (abundance), and Time. Single population growth rate curves can be identified by as unique temperature-species-medium-citation-replicate combinations (concatenate them to get a new string variable that identifies unique growth curves).

Initial Investigation - Tuesday

* Read relevant papers
* Explore the data in R and get a preliminary version of the plotting script without the fitted models overlaid worked out. That will also give you a feel for the data.
* Explore the two models – be able to plot them. Write them as functions in your python script, because that's where you will use them (step 2 above) (you can use matplotlib for quick and dirty plotting and then suppress those code lines later).
* Figure out, using a minimal example (say, with one, "nice-looking" thermal performance, functional response, or population growth curve/dataset) to see how the python lmfit module works. We can help you work out the minimal example, including the usage of try to catch errors in case the fitting doesn't converge.

First script (R) - Wednesday

* Import data and prepare for NLLS fitting
* It should create unique IDs so that you can identify unique datasets
* It may filter out datasets with less than x data points (where x is the minimum number of data points needed to fit the models
* It should deal with missing, and other problematic data values
* Script should add columns containing starting values of the model parameters for the NLLS fitting (how will you get these?)
* Save the modified data to a new CSV file

Second script (R) - Thursday

* Open mew modified dataset and do NLLS fitting
* Use the ‘try’ construct because not all runs will converge
* The model data curves you are able to fit the better – does this mean apply the models to as many graphs as possible or as many models as possible?
* Calculate AIC, BIC and R2 and other statistical measures of fit
* Export results to a csv that the plotting R script (new item) can read

Third Script (R) - Friday

* Import results from previous step and plot every curve with the models overlaid – all plots should be saved in a single sub-directory (use ggplot)

Write Report – Reading Week

* The report should be written in LaTeX using the article document class, in 11pt (any font will do, within reason!).
* be double-spaced, with continuous line numbers.
* have a title, author name with affiliation and wordcount (next point) on a separate title page.
* have an introduction with objectives of the study, and appropriate additional sections such as methods, data, results, discussion, etc.
* should contain in the Methods a sub-section called "Computing tools" which states briefly how each of the three scripting language (bash, R, Python) and what packages within them were used and a justification of why.
* must contain ≤ 3500 words excluding the contents of the title page, references, and Figure or Table captions+legends; there should be a word count at the beginning of the document (typically using the texcount package).
* have references properly cited in text and formatted in a list using bibtex.
* For the writeup, you probably should read the general (not word count, formatting etc.) dissertation writing guidelines given in the Silwood Masters Student Guidebook.

Fourth Script – Reading Week

* LaTex source code that generates report

Fifth Script (Python) – Reading Week

* Run\_MiniProject.py, runs the whole project, right down to the compilation of the LaTex document