Syllabus for MMG 1004H: Basic Computational Biology: A Practical Course in Programming for Biologists

Course Instructors: Philip M. Kim and Gary D. Bader (Gary Bader only in 2023, Philip is on sabbatical)

Course Time and Dates: Winter Semester (6 weeks); Mondays and Wednesdays; dates TBD

Course Location: in-person

Prerequisite:

This course is intended and required for first year MoGen students (MSc or direct entry PhD) who do not have advanced computational biology training. Graduate students from other departments will be considered on a case-by-case basis if there is capacity. CBMG students will enrol in Foundational Computational Biology I (MMG1344; 0.25 FTE) instead of MMG1004.

Total Student Contact Hours (0.25 FCE): 15 hours (mixed lecture and lab) plus 10 hours of recorded lectures, plus TA office hours

Course Objectives and Learning Outcomes:

This course is designed to teach experimental biologists the basics and hands-on knowledge of bioinformatics programming. In today's world, most graduate students in the Molecular Genetics will encounter situations where they have to make use of computational tools and deal with larger amounts of data. The main objective of this class is to give students the power of automation via Bioinformatics programming. The class teaches by example and gets students comfortable with doing basic programming in R and adapting existing programs to their needs, as well as interface with standard bioinformatics software. We will also give a brief intro to the more general language of Python as well as software development tools like GitHub.

Course Description and Format:

The class is a standard MoGen module, covering six weeks. The course will be delivered in person or online as required. Both the in-person and online formats will involve viewing recorded lectures, with the exception of Lecture 1 (which will be in-person or online). The course will proceed according to the following schedule Monday/Wednesday schedule:

Each lecture is to be viewed on Mondays and Wednesday before the lecture.
On Mondays and Wednesdays we will all gather in the "Plenary session" for general discussion and Q&A.

In each week one homework assignment will be handed out; we would like to emphasize that they will be fairly labour intensive. Completing these assignments is an integral component to the class and a large portion of the grade will be based on it. The previous week's homework assignments solution will be discussed in a TA session.

Finally, each TA will hold weekly office hours.

Delivery of the course:

Week 1: Goal getting comfortable with basic tools

Lecture 1

- Introduction to programming.
- The Unix shell (command-line) environment.
- Getting comfortable with the shell, basic shell commands, some examples

Lecture 2

- Introduction to R, notebooks, RStudio
- R Statements, Basic Syntax and Variables

Week 2: Basic R Lecture 3

- Numbers, strings, vectors, matrices

Lecture 4

- Lists, arrays, data frames

Week 3: Making more complex scripts Lecture 5

- Flow control, files

Lecture 6

- Loops, vectorization

Week 4: Functions, external programs and recipes Lecture 7

- Functions, regular expressions

Lecture 8

- Interfacing with external programs - Using programming recipes

Week 5: Packages and Bioconductor Lecture 9

- R packages, bioconductor

Lecture 10

- Plotting in R

Week 6: Intro to Python and Github Lecture 11

- Intro to Python

Lecture 12

- Coding and data best practices
- Intro to version control with Git and GitHub

Method of Evaluation:

Grading: 60% assignments (6 assignments), 40% final project.