Biology 381 - Computational Biology

Course Syllabus

Spring 2022

Meetings

Lecture: Tuesday, Thursday 1:15 - 2:30 pm, 129 Marsh Life Science **Lab:** Wednesday, 1:10 pm - 3:40 pm, 129 Marsh Life Science

Instructors

Lecture: Nick Gotelli
209 Marsh Life Science
ngotelli@uvm.edu
Video conferences by appointment

Lab: Lauren Ash
211 Marsh Life Science
Lauren.V.Ash@uvm.edu
Video conferences by appointment

Lab: Emily Beasley 211 Marsh Life Science
Emily.Beasley@uvm.edu Video conferences by appointment

**Video-conference web link: https://app.inspace.chat/space/3D1AE70150195F8B7890ADE9027FFC3E982F8A3F

Pre-Requisites

- A personal laptop computer that you bring to all lecture and lab meetings
  - Mac or Windows, with battery power to last 1.5 hours
  - all required software packages installed and working
  - wireless access and an active UVM mail account
- Graduate standing as a PhD or MS student at UVM
- Undergraduates by invitation only after consultation with the instructor
- Knowledge of basic statistics (p-values, means, variances, hypothesis testing)
- No prior knowledge of computer programming is required

Course Content

This course is designed to teach you three things:

1. How to use modern (and classic) computational tools to make your analysis, writing, and presentations more efficient and attractive, and to make your analyses transparent and repeatable. Such tools include plain-text editors, markdown, github, regular expressions, and shell commands.
2. Foundational methods in computer programming in R (data structures, functions, control structures, input and output) that will allow you to construct computer models and to easily learn other computer languages (Python, C++).
3. How to use probability distributions, simulate data, recognize, use, and analyze 4 archetypal experimental designs for biologists, apply them to real and simulated data, and create publication quality graphs with the `ggplot2()` package in R.

Lecture Activities

> For 2021 lecture instruction during the pandemic, I have recorded a complete set of video screen casts (available here). Ideally, you should watch these during the scheduled course times to keep up with the lectures and be able to keep up with the homeworks.

Once we finish with the preliminaries, the format for the lecture each day is simple. I will open Rstudio from my computer, project the screen so you can see it, and begin coding. I will comment and explain everything as I go along. You will copy what I am typing on the screen into your own computer and the run the code along with me. This is the fastest way for you to learn how to code.

Lab Activities

> For 2021 lab instruction during the pandemic, Lauren will host the labs live on Teams and teach it in the usual way. Attendance is mandatory, although there may be some days when you do not need to stay for the entire time. It is your responsibility to keep up with all of the assignments, and regular post your completed homeworks to your github webpage (which you will create during the second week of lab).

Each week, there will be a programming assignment to go along with the lecture. Your primary in the laboratory section will be to do this assignment. Using the lab time this way forces you to set aside time each week to get learn the language and get the coding done, minimizes your time spent coding outside of class, and gives you the benefit of being able to ask for help from Lauren and from your fellow students.

Lauren will begin each class with a brief overview of the assignment, and possibly a few suggestions or tips for how to code the exercise.

Student Responsibilities

- Attend nearly all of the lectures and lab meetings
- Do the coding exercises in class when they are assigned
- Regularly update your course webpage with completed and partially completed assignments
- Study the code of your classmates on their webpages to see how they have solved the same problems you are working on
- Annotate your code so it is useful to you and others
- Surf the internet to find solutions to programming problems
- Share code and ideas, and help others who are struggling
- Use the course work to ideally complete some concrete work that will go towards your thesis

Grading

The undergraduate grading paradigm of taking examinations in a limited time frame and doing all of your work exclusively by yourself is antithetical to the programming world. Unless you are at NASA trying to guide an emergency landing for Apollo 13, you will not be writing computer code under a strict time deadline. And, in contrast to the strictures against plagarism in most of science and academia, we will regularly use computer code that others have written and put it to our own use. When you do so, try to include some brief annotation so that you know where the code came from and can get back to the source if you need to. Copying and repeating what others have said or written is how we learn any language, including computer languages. The only thing you should not do is to copy an entire program wholesale without documentation and without any understanding of what it is doing.

Although there will be weekly homeworks, there is no specific due date on these assignments. As soon as you finish them, you will upload them to your individual website. This website is your public “portfolio” that shows your progressive work through the semester. Course grades will be based on an evaluation of your overall portfolio (quality of coding and annotation), attendance, and participation.