Syllabus for 290: Introduction to Programming with Biological Data
Instructor: James Umbanhowar Teaching Assistant Professor of Biology
Email: jumbanho@unc.edu
Class meeting: MWF 9:05-9:50 Stone Center 200
Office hours: M 10-12 W 12-2
Office hour location: Mitchell 121

Description

All subdisciplines of biology deal with data. As the amount of data available increases, automated methods of reading, manipulating and displaying data are necessary. Unfortunately, data is not always packaged in a fashion that allows biologists to use pre-existing programs to meet their needs. Programming is, therefore, a key skill for biologists of all types. In this course, you will learn the basics of practical computer programming oriented towards dealing with biological data. The focus will be learning useful techniques rather than theoretical approaches that would be the focus of a computer science course. We will use the programming language R which is the language of choice for many practicing biologists, data scientists, and statisticians.

Course objectives

Core learning goals for this course are:

- Write and debug programs in R
- Become accustomed to general programming concepts that will allow you learn other languages
- Apply programming skills to address biological problems using data
- Generate automated data processing for a standard data structure

Prerequisites

There are no formal prerequisites for this course.

Text

There is no text for this course. Readings will be posted online.
A good reference is Beginning R An Introduction to Statistical Programming by Wiley and Pace.
For more advanced details in using R I recommend Advanced R, Second Edition by Hadley Wickham

Grading

Course grade will be determined on weekly assignments and a final exam.
Weekly assignments: 70%
Final exam: 30%

Grading scale:

Percentage	Grade
93.3	A
90	A-
86.7	B+
83.3	B
80	B-
76.7	C+
73.3	C
70	C-
60	D
<60	F

Assignments

Assignments will be due on Sunday at midnight. Assignments will involve interpreting, writing and debugging code. Late assignments will be deducted 5% per day.
You will be asked to grade your assignments in this course based on a key and a rubric I provide. This process will provide you an opportunity to provide feedback on the concepts in class that may still be unclear to you.
After you have completed grading your assignments, I will read the graded assignments to confirm your grading.

Course Policies

- Class attendance is not required, but is highly recommended. In class exercises are key to developing mastery of the concepts of this class. If you do miss class, you will probably want to come to office hours to catch up.
- Late assignments will be marked down 5% per day unless previous permission has been granted.
- I recognize that life events can interfere with your ability to attend class and complete assignements. I will give one no excuse assignment drop that will exclude that particular assignment from grading.
- If you have a life event that affects your ability to do your class work over an extended periods of time, please contact me as early as possible and we can design a plan for you to progress in the class.

Honor Code

Academic integrity is a core value of the University and the greater academy of ideas. The student led Honor System is the institution that is responsible for monitoring this code. As such this class adheres to that code and any violations of the code will be reported. You can find more information about the honor code here: `http://catalog.unc.edu/policies-procedures/honor-code/`. As in all classes at Carolina, your are expected to adhere to this code.

Course content

In this course we will cover the following concepts of computer programming with R:

- Interacting with the R interpreter with expressions
- Variables and variable types
- Data structures: vectors, matrices and data frames
- Interacting with strings
- Regular expressions
- Using and writing functions
- Control structures and boolean variables
- Loops with for and while
- File input/output
- Graphics
- Creating your own data types
- Debugging code
- Making code run faster ("Optimizing")
- Interacting with databases
- Using version control