**Instructor:** Danielle Presgraves, Ph.D. [**danielle.presgraves@rochester.edu**](mailto:danielle.presgraves@rochester.edu)

**Office Hours:** There are only a few of you so please just email me and we'll find a time to zoom. We will need to set up an initial zoom meeting in the first week of classes and we can discuss how everyone prefers to proceed, ie. weekly group check in, or monthly group check-in? Let's see what works for everyone. This is the first time this course has been offered, so some flexibility will likely be required for everyone, but we'll do our best to fit into everyone's scheduling constraints!

**Teaching Assistants Information:**

**Content:** The wide appeal of the computational biology major is rooted in diverse and rigorous training in technical fields such as developing algorithms, programming, statistics, data analysis and modeling in a biological context. There is a great deal of flexibility in the degree, but that can result in exposure to slightly variable core principles as well as the development of different skills as individual students work through the degree.

The purpose of BIOL 160 is to provide an opportunity to cohesively review and synthesize the fundamental concepts that are taught across the various courses that lead to a computational biology degree. You will learn collaboratively from your peers as you complete assignments as a group. Students will refresh key technical skills such as basic programming, and statistical analysis and visualization, and Data Structures and Machine Learning working on problem sets together. You can individually deposit your code in Github (a tutorial has been developed to help you do this). This course also includes collaboration with the Gwen Greene Career Center to develop a personal website (via 'Wix"), and a CV that targets data science/computational biology careers.

The content of this course will be broken into the major modules outlined below. The fundamental concepts that are featured in modules 1-5 are already addressed in various courses required for the Computational Biology major, but the focus of existing classes is on the 'theoretical' aspects of these important foundations. BIOL 160 will provide more concrete 'hands-on' learning opportunities of these skills and encourage participants to think about the transfer of these techniques to areas outside of biology.

**Learning Management System:** We will be using Blackboard and (probably) box since the problems are intended to be group activities.

**Credit Hours:**  This is a one credit hour course. For every credit hour of a course at the University of Rochester, our accreditation body lists the expectation that students complete at least an additional **2 hours** of out of class time work (outlined for one credit hour here: <https://www.rochester.edu/provost/assets/PDFs/UR-Credit-Hour-Policies-March2016.pdf>). This policy translates to approximately 2 hours out of class time, depending on your particular background, per week for a one credit hour class. If you are spending significantly less or more time than this on this course per week, please come and talk with me. As the first iteration of this course, it is crucial that I know how to modify the wEmpork load of the course to ensure that it is both useful and a reasonable amount!

**Grade Composition:**  This course reviews and synthesizes material learned in previous courses. As such, you cannot earn a letter grade and it is only offered as **S/F**. As long as you engage with the material and contribute to the problem sets with your peers (project communication is important and is emphasized in this course), you are expected to pass the course.

**Modules: Learning Objectives and Material Covered:** *This is the first time this course has been offered and the assigned problems and module material are our best estimate at both what is relevant, and what can be reasonably completed as a group in a one credit course. For the course to be successful, we need your feedback!*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module Number** | **Module Name** | **Existing Courses that originally teach material** | **Content/concepts** | **Learning Objectives** |
| **0** | Introduction to the course  **Week 1** |  | * **Meet to discuss expectations and introduce everyone** | * Everyone connected * Emphasis on communication and material |
| 1 | Programming Languages  **Weeks 2-4** | BIO208, CSC161, CSC171, BIO257L, BIO259L, BIO152, BIO153, | * Introduction to Jupyter – we're going to focus on Python * Variables/breaking down problems/how to think like a programmer * Loops   + Functions   + List comprehensions * OOP   + Versus procedural programming * Recursion * General intro to Libraries * Small introduction to R | **Implement:**   * Break down and solve problems with elements of Python programming   + Use controls statements (if/elif/else) and AND OR NOT   + Apply for and while loops, bundle repeated loops into functions, and modules   + Replace loops with list comprehensions   + Recursion problems   + Regular expressions * Recognize data types (lists, tuples, dictionaries, iterators, list comprehensions) * Apply the appropriate libraries and manipulate data with their features. Explain why these libraries are powerful.   + NumPy     - Arrays     - Subsetting/slicing   + Pandas     - groupby   + BioPython   + SQLite: SELECT, APPEND, SORT, FILTER, JOIN, AGGREGATE   **Debug and refactor:**   * Identify problems in code * Know how to use assertions and review debugging strategies   **Design:**   * Create and contrast a solution to a problem that uses all three types of programming: OOP, procedural and recursion * Use appropriate libraries to solve given problems   **Analyze and interpret:**   * Follow scope of variables: <http://pythontutor.com> is particularly good for this |
| 2 | Data Structures and Algorithms  **Weeks 5-6** |  | * Trees (especially connected to Recursion) * Major algorithms (especially wrt Big O notation): DFS vs BFS, mergesort, quicksort | **Demonstrate:**   * Bit manipulation   **Explain:**   * Importance of Big O notation and be able to compare the runtime of the common algorithms (DFS, BFS, mergesort, quicksort) using O notation * Explain stacks and queues   **Apply:**   * Compare and contrast hashtables, arrays, linked lists * Compare and contrast recursion and traversing binary search trees.   **Data structures and algorithim resource**   * <https://www.geeksforgeeks.org/data-structures/?ref=grb> * <https://www.tutorialspoint.com/python_data_structure/python_data_structure_introduction.htm>   **Design:**   * solve problems using common algorithms * justify the choice of algorithms |
| 3  3 | Data Analysis:  Statistics, Probability, and Mathematics  And  Data Visualization  **Weeks 7-8** | BIO218P, BIO219L, BIO259L  BIO/STT214, BIO253L, BIO259L | * Statistical analysis * Flow chart of statistical tests: ie. t tests, ANOVA, GLM * Relationships between variables (ie. correlation/regression) * Power tests * Bayes' versus frequentist * Data visualization * Flowcharts of variable types and their most appropriate graph * Data visualization libraries | * **Utilize** the appropriatestatistical test and appropriate arguments * R programming:   + Data visualization   + ggplot2   + popular libraries   + \*apply functions   + SWIRL() * Compare and contrast the capabilities of R and Python and the types of problems that best utilize the strengths of each language * **Compare** and **contrast** ggplot2 library from R and plotly library from Python * **Explain** the difference between frequentist and Bayesian approach to solving probability problems. |
| 4 | Machine Learning  **Weeks 9-11** | BIO253L | * Unsupervised versus supervised * Regression versus Classification methods * PCA (linear algebra refresher) * K-clustering, near neighbours * Maybe MCMC (if time) | * **Compare** and **contrast** supervised versus unsupervised learning * **Explain** the principles ofclassification methods (usually supervised) and regression or dimensionality reduction * **Compare** and **contrast**  K nearest neighbor, neural networks, Principle Component Analysis methods * **Apply** k nearest neighbor, neural networks and PCA to the appropriate problems. |
| 5 | Behavioral Questions  **Week 12** | We will collaborate with the experts at the Gwen Greene Career Center. This module will stress "thinking on your feet" and focus on breaking a problem down and giving a 'chalk talk' via voicethread where students will need to explain their solution via short video. | * **GGCC:** * Mock interviews with GGCC * Quinncia – AI platform for prep interviews (prior to interview with alumnus) and resume review * Potential alumni mock interviews | [**https://www.tutorialspoint.com/python/python\_interview\_questions.htm**](https://www.tutorialspoint.com/python/python_interview_questions.htm) |
| 6 | Developing your own personal brand  **Week 13-14** | Developing an effective C.V. and construction of your own ''brand" website. This module will be offered concurrent with the other models so students have multiple iterations to improve their website over the duration of the course. | * Templates for CV (GGCC has this) * Template for website through Wix * Github tutorial | [**https://flowcv.io**](https://flowcv.io) |