MOLB 7900 Practical Computational Biology for Biologists: Python Spring 2019

Organization and Contacts:

Srinivas Ramachandran (Director, Instructor): srinivas.ramachandran@ucdenver.edu
Matthew Taliaferro (Director, Instructor): matthew.taliaferro@ucdenver.edu

Room = XXXXXXX
January 28 to February 15
MWF 9am - 11am
Office hours: Tuesday 9-11 in RC1 South 10114, Thursday 9-11 in RC1 South 9102

Goals and Learning Objectives

The goal of this course is to introduce students to the Python programming language and its application to computational biology. Students are not expected to have any prior experience with Python or other programming languages. The goals of the course are to familiarize students with the basic tenets of Python programming such that, at the completion of the course, they are able to write basic software and scripts that will enable them to derive meaning from the large datasets typical of modern biology. Further, the students will have the basic skills necessary to continue their computational development using tried-and-true crowdsourced knowledge repositories including StackOverflow. In short, this course aims to “teach students to fish” by teaching them the necessary basics of the language and pointing them toward resources for further development rather than “giving students a fish” by handing them pre-designed, cookie-cutter scripts that answer specific biological questions.

The **primary learning objectives** are therefore to work towards being able to:

1. Understand basic datatypes, their properties, limitations, and combinations.
2. Master the use of different containers of these datatypes and understand their relative advantages and disadvantages.
3. Understand and use iterating loops and their associated controls.
4. Incorporate external packages into the workflow of student-designed software.
5. Be able to define and write functions, and understand their scope as well as incorporate Python’s powerful built-in functions where appropriate.
6. Use the spreadsheet-mimicking package pandas to organize and manipulate tabular data.
7. Create a computational workflow that identifies analyzes the genes and sequences bound by a protein in a ChIP-seq experiment.
8. Create a computational workflow that integrates ChIP-seq and RNA-seq data to analyze the biological consequences of chromatin-binding events.

Prerequisites

There are no strict prerequisites for this course. However, a basic knowledge of the Unix command line may be helpful. Additionally, a basic knowledge of the experimental techniques behind genomic experiments including ChIP-seq and RNA-seq may also be helpful.

A personal laptop for a student is also not required, but if one is available, it is highly encouraged that the coursework and materials be performed on the laptop so that the student has continuous access.

It is **REQUIRED** that all students configure their laptops for use with course materials. Depending on the familiarity of the student with installing Anaconda, Python 3.6.2, and Jupyter 1.0.0, they can:

i) Do it themselves OR
ii) Attend a preliminary meeting at XXXXX in XXXXX, where the instructors will be available to help OR
iii) If students cannot attend the above meeting, they can set up an appointment with an instructor **PRIOR TO THE FIRST CLASS** for this configuration to take place.

The student should have Python, Anaconda, and Jupyter working on the computer they intend to use for the course before 1st day of the class - configuring software will not be done once the classes have begun.

Structure Overview

Classroom time will consist of a brief lecture on the theory and principles behind the day's material. This will be followed by interactive exploration of the topics to be covered using Jupyter notebooks (installed using Anaconda - instructions will be provided before class starts). These notebooks allow fully integrated explanation of topics, blocks of code, and embedded graphics. Each classroom session will consist of one Jupyter notebook that will cover that day's material. Near the end of the notebook, there will be a coding exercise that will be completed by the student within the notebook.

Examinations and Grading

Coding exercises at the end of each lesson will be completed by students within the notebook. These notebook files must then be turned in to the instructors before the beginning of the next class period. These exercises will, in sum, account for 50% of the student's grade. The remaining 50% will be comprised of a take-home final exam to be administered after the completion of the final classroom session. This exam will also be completed in Jupyter notebook format and must be turned in to the instructors by 5 PM on February 22. In completing the exercises and final exam, any materials available to the student, online or otherwise, are permitted to be used.