BIOF 309 - Introduction to Python

Chris Coletta (lead instructor) - christopher.coletta@gmail.com
Matt Shirley - mdshw5@gmail.com
Anatoly Dryga - anatoly.dryga@nih.gov
Ben Busby - ben.busby@gmail.com

Spring 2014

- Class forum: https://groups.google.com/d/forum/faes-biof309-spring2014
- Group email address: faes-biof309-spring2014@googlegroups.com

1 Course Description

This course is intended to teach research professionals without a background in programming to write programs to gain insight into data. In addition to covering tools and syntax that are specific to Python, the class will cover elementary concepts that are ubiquitous in modern software engineering, including object-oriented programming, regular expressions, reading from and writing to text files, recursion, use of the debugger, etc. The end of the course will focus on potential applications of the Python language to bioinformatics, including sequence analysis, machine learning, and data visualization.

2 Logistics

The class will meet sixteen times during the semester at 5:30pm every Thursday from January 30, 2014 until May 15, 2014. Our classroom will be in Building 10 on the main NIH campus in Bethesda, room B1C211. Many lectures will be recorded ahead of time as a screencast which students can watch over the Internet at their leisure. It is hoped that this will free up the meeting time to serve as a discussion section or a time to do class exercises, as well as allow the student to learn at his or her own pace. It is the responsibility of the student to watch the lecture before coming to class in addition to submitting homework assignments. *Students are encouraged to use the Google groups email list (link above) to ask for help or pose a question, rather than email the instructor directly.*

3 Required Materials

Each student is required to have a computer running Python 2.7. Instruction will be given using Python 2.7, while syntactic differences between 2.x and 3.x will be highlighted. When installing Python, students are recommended to use the free Anaconda Scientific Python Distribution, which bundles core Python with many essential & useful 3rd party modules. Choice of operating system is left to the student's discretion.

3.1 Optional Textbooks

There is no required textbook for this course, but students may find these textbooks useful:

- Downey, Allen B. Think Python. Available free online
- Model, Mitchell L. Bioinformatics Programming Using Python. 2009, O'Reilly Media.
- McKinney, Wes. Python for Data Analysis. 2nd ed. 2013, O'Reilly Media.
- Jones, Martin. Python for Biologists. 1st ed. 2013, CreateSpace Independent Publishing Platform.

4 Grading

4.1 Homework

Homework will count for 80% of the final grade. Homework should be submitted via email attachment to the instructor who assigned it before the beginning of class on the day it is due. *N.B.: ANY FILES SUBMITTED MUST ADHERE TO THE FOLLOWING NAMING CONVENTION: lastname_firstinitial_hw#.py, i.e., no spaces, name appears first and assignment number appears second in the filename.* Homework must be able to be executed using a Python 2.7 interpreter – submitted programs written in Python 3.x is not acceptable. Grading will be based on the following rubric:

Program runs, contains useful comments, meaningful variable names, follows "Pythonic" coding conventions : A
Program runs, produces correct result: B
Program runs, produces something close to the correct result: C
Program runs, does not produce correct result: D
Program does not run: F

4.2 Final Project

The final project will count for 20% of the final grade. Each student will write a program to solve a problem relevant to their work, studies or interests. The

student will have wide latitude in establishing the parameters of their project, with the expectation that the program will do one or more of the following: organize, manipulate, analyze, visualize, interpret some data set, or perform a calculation. Each student be responsible for determining and writing up the requirements for the program (what will it do? who will use it?), and making a five minute presentation about their project to the class at the end of the semester.

5 Schedule

Week 1 - January 30, 2014 - Introduction and Installation

Housekeeping issues; How to get help; Discussion of programming language ecosystem and where Python fits in; What makes Python distinctive; The Python data analysis stack: core Python + essential 3rd party modules; Setup of Environment; Using Python interactively via IPython notebook; Running a program. Homework: Email me the magic number.

Week 2 - February 6, 2014 - Python Primatives

Exceptions. Named Values(Variables); Core Python types; Conversion between types; Math expressions; Matrix operations using NumPy; Strings, with escape characters.

Week 3 - February 13, 2014 - Logic, Lists, and Loops

Boolean expressions & operators; conditional flow statements; iterable types; strings as iterables; loops.

Week 4 - February 20, 2014 - Functions & Debugging

Namespaces; Global vs. local scope; Functions: definitions, arguments, return statement, decorators; Coding Style; Debugging: how to invoke, commands.

Week 5 - February 27, 2014 - Slice, Dice, Combine & Sort

File IO; Parsing strings, List comprehensions; Sorting using a lambda function; string interpolation/formatting; Case study: Gene list with Z-scores.

Week 6 - March 6, 2014 - Pattern Matching (Regular Expressions)

Why they are needed; Example Usage; Python RegExp workflow; Metacharacters; Character Classes; Quantification; Grouping; Back-referencing; Lazy vs. Greedy

Week 7 - March 13, 2014 - Classes and Object Oriented Programming, Part I

Definition; Philosophical underpinnings; Syntax of defining classes; Methods; Instance variables vs. class variables.

Week 8 - March 20, 2014 - Classes and Object Oriented Programming, Part II

`__init__(self)` constructor; @classmethod decorator; Magic methods, a.k.a. object hooks; Inheritance; Abstract base classes.

Week 9 - March 27, 2014 - Recursion & Tree Traversal

Call stack; recursive functions; NCBI taxonomy tree traversal; Systems biology tree traversal; Case Study: Guess my girlfriend's name.

Week 10 - April 3, 2014 - Machine Learning

Cross-validation; Training error vs. testing error; Classification: Support Vector Machines; Regression: Elastic Net; Dimensionality Reduction: PCA; Clustering: KMeans.

Week 11 - April 10, 2014 - Data Visualization

matplotlib; vincent; vega; D3.js; HTML5 Canvas element

Week 12 - April 17, 2014 - Sequence Alignment using Biopython

Lecture by Matt Shirley.

Week 13 - April 24, 2014 - Querying aligned short read sequences using pysam

Lecture by Matt Shirley.

Week 14 - May 1, 2014 - Manipulating GWAS data using R

Guest lecture by Jun Ding.

Week 15 - May 8, 2014 - Final Project Presentations

Week 16 - May 15, 2014 - Final Project Presentations