Programming for Genomics IB 501 (4 credits)
Instructor: Julian Catchen <jcatchen@illinois.edu>
Class Schedule: Monday and Wednesday, 1-2:50PM
Office hours: Friday 3pm in 233 Morrill Hall or by appointment

Course Description:
The goal of Programming for Genomics is to teach students to *think algorithmically*. By the end of this course, students should be able to construct a biological hypothesis, and implement code, or deploy an existing code implementation to test that hypothesis. Students will first learn to use the UNIX operating system, including the use of command streaming through pipes, and will learn to program in Python. We will use biological data sets, mostly from high-throughput sequencing genomics projects, as a base for learning UNIX and Python. As the course progresses we will cover major genomics approaches and the algorithms that underlie them, including K-mer analysis, whole genome and transcriptome assembly, databases and SQL, and visualization techniques. Students will use these analysis programs and supplement the analysis with data generated by their own Python code. This course will provide students with a strong introduction to the major bioinformatics analyses approaches as well as the ability to write their own code in Python, within a UNIX environment.

Student Learning Outcomes:
The main learning outcome is teaching students how to *think algorithmically*. That is, starting with a biological hypothesis, how to break down the problem of biological analysis into a series of algorithmic steps, and then how to use code to implement those steps and thereby test the hypothesis. The second learning outcome is to study the major algorithmic approaches to understand the landscape of genomic analysis. Finally, the third outcome is to understand the state of the art in the field of population genetic and comparative genomic analysis by studying the literature in this area.

Prerequisites:
No computational skills are assumed for this course, however familiarity with molecular biology and evolution is expected and necessary.

Textbook:
Downey, Allen. 2015. Think Python: How to Think Like a Computer Scientist. Green Tea Press.

Journal articles:
Readings from primary literature for discussion sections, and tutorials for computer labs, will all be made available online.

Grading:

Quizzes	20%
Presentations	10%
In-class Assignments	15%
Problem Sets	45%
Discussion Participation	10%

Expectations: My primary expectation is that students come into the class ready to learn new skills and develop new perspectives. The course does not assume any previous experience in programming, but different students are able to learn the abstract thinking involved in programming at different rates. So, if it is a brand new skill, you will have to work harder to master it. Attendance is essential, as well as engagement with the assigned readings, to get the most out of the course.

All students should follow University of Illinois "Code of Policies and Regulations Applying to All Students." The Code is available online at: http://www.admin.uiuc.edu/policy/code/index.html

Class Attendance and Late Work:
Attendance in class is required. Late work will be allowed at the discretion of the instructor with a late penalty.

Class Format:
The general format for each class will be a lecture in the first hour followed by time to work on projects. Early in the course we will use lecture/lab time to work together on new skills but that will transition to a more open-lab format as we work on more substantive projects. Groups of students will take turns presenting journal articles the second meeting of each week, and all students will be expected to read and discuss these articles ahead of presentation online.

Computer Requirement
Students are required to have a laptop computer, or at least reliable access to a personal computer, to complete computational assignments. We will do our actual computing on the on-campus IGB BioCluster, but a personal computer will be necessary to access the cluster outside of class. Class will take place in the IGB computer lab, so you have the option to bring your laptop to class, or use the computers in the lab to access the cluster during class time.

In-class assignments:
Early in the course we will work on assignments together in class (and students will continue working on them after class). These assignments will serve as a scaffold for learning UNIX and Python. These assignments are worth 15% of the final grade.

Problem sets:
The main way we will engage with the material, and the bulk of the final grade in the course, is through applied projects. Each problem set will involve a bioinformatics analysis, which will typically include a mixture of running existing software – for example to assemble a genome or identify SNPs – combined with writing your own Python code or using UNIX to answer additional questions about the data. Problem sets make up 45% of the final grade.

Quizzes:
There will be several quizzes during the course in total worth 20% of the final grade. These quizzes will cover key concepts from the lecture material and particularly the assigned reading. Students with a valid reason for missing a quiz will be given an opportunity to take a make-up exam at the discretion of the instructor. Valid reasons include only medical reasons (with a note from McKinley), tragedy in your immediate family, or religious observances and practices.

Group paper presentations:
Each week a group of students will be responsible for presenting papers from the literature relevant to the current topic in the course. The group in charge of the papers will post a summary to the Moodle discussion group several days ahead of class. The group will then present the paper during class as well. Each presentation should be about 30 minutes in length. These presentations are worth 10% of the final grade.

Discussion participation:
Readings from the literature will be provided online in advance of the group presentations. Students are expected to participate in an online Moodle forum prior to each paper presentation so that students get the most out of the in-class presentation. This participation is worth 10% of the final grade.

Academic Integrity
According to the Student Code, "It is the responsibility of each student to refrain from infractions of academic integrity, from conduct that may lead to suspicion of such infractions, and from conduct that aids others in such infractions." Please know that it is my responsibility as an instructor to uphold the academic integrity policy of the University, which can be found here: http://studenttcode.illinois.edu/article1_part4_1-401.html.

Disability Accommodations
To obtain disability-related academic adjustments and/or auxiliary aids, students with disabilities must contact the course instructor and the Disability Resources and Educational Services (DRES) as soon as possible. To contact DRES you may visit 1207 S. Oak St., Champaign, call 333-4603 (V/TTY), or e-mail a message to disability@illinois.edu.

The following schedule and set of topics is subject to change, depending on the pace of class and other factors.

Weekly Schedule

Week	Class	Topic
1	Day 1	Introduction to Genomics; Introduction to UNIX, part 1
	Day 2	Introduction to UNIX, part 2; Papers assigned, Group #1
2	Day 1	Advanced UNIX, part 1
	Day 2	Advanced UNIX, part 2; Paper Presentation Group #1, #2
3	Day 1	Labor Day Holiday, No class
	Day 2	Advanced UNIX, part 3
4	Day 1	Introduction to Python, part 1
	Day 2	Python, part 2; Paper Presentation Group #3, #4
5	Day 1	Python, part 3
	Day 2	Python, part 4; Quiz #1; Paper Presentation Group #3
6	Day 1	Introduction to K-mers: Normalization, Error Correction, Repeat Estimation
	Day 2	K-mers: the BLAST and BLAT algorithms; Paper Presentation Group #4
7	Day 1	Dynamic Programming
	Day 2	Whole Genome Assembly Algorithms, part 1; Paper Presentation Group #5
8	Day 1	Overview of Computer Hardware; Whole Genome Assembly, part 2
	Day 2	Short-read Alignment Algorithms and SAMTools, part 1; Paper Presentation Group #6
9	Day 1	Short-read Alignment Algorithms and SAMTools, part 2
	Day 2	Introduction to Databases and SQL, part 1; Paper Presentation Group #7
10	Day 1	Databases and SQL, part 2
	Day 2	Population Genomic Analysis Algorithms with RAD-seq, part 1; Quiz #2; Paper Presentation Group #8
11	Day 1	Population Genomic Analysis Algorithms with RAD-seq, part 2
	Day 2	Using Hidden Markov Models to identify Genomic Signals; Paper Presentation Group #9
12	Day 1	Transcriptome Assembly, Part 1
	Day 2	Transcriptome Assembly, Part 2; Paper Presentation Group #10
13	Day 1	Ortholog Discovery
	Day 2	GO Term Analysis; Paper Presentation Group #11
14	Day 1	Gene Expression Quantification, part 1
	Day 2	Gene Expression Quantification, part 2; Paper Presentation Group #12
15	Day 1	Data Visualization with Python, part 1
	Day 2	Data Visualization with Python, part 2 Quiz #3; Paper Presentation Group #13