MCB112 Biological Data Analysis: Fall 2018

http://mcb112.org

Lectures: Mon/Wed 1:30-2:45pm, Biolabs 1080

Section: Fri 1:30-2:45pm, Biolabs 1080

Instructor: Prof. Sean Eddy seaneddy@fas.harvard.edu

Office hours: Mon 3-4pm, Biolabs 1008

Description

Biology has become a computational science. New technologies are generating larger and more complex data sets, especially in genomics and imaging. This course teaches computational, statistical, and mathematical methods for biological data analysis, using an empirical and experimental framework suited to the complexities of biological data, emphasizing computational control experiments. The course is primarily aimed at biologists learning the fundamentals of data analysis methods, but it is also suitable for computational, mathematical, and statistical scientists learning about biological data.

Aims and learning objectives

MCB112 teaches fundamental principles of biological data analysis by example. The course is built around roughly 12 weekly data analysis problems. These problems typically use synthetic simulated data sets from a fictitious *in silico* creature, the sand mouse *Mus silicum*. Most problems focus on gene expression analysis with RNA-seq, but this is not an RNA-seq course *per se*.

In the course of solving analysis problems, you will learn practical skills in how to write scripts to analyze data, and how to use simulations to do computational positive and negative control experiments. The course is taught in Python, using Python-based data science tools including NumPy, SciPy, Pandas, and Jupyter Notebook. You will learn how to do computational science: how to understand computational methods, how to design computational experiments, how to think critically, how to develop an organized work pattern, and how to communicate results reproducibly.

You will also learn fundamentals of probabilistic inference, statistics, computer science, and applied math – how to think about statistics from first principles, and how to read and understand an algorithm well enough to implement it. The course aims to motivate biologists to learn more mathematics, computer science, and statistics, by showing how these skills are relevant to biological data analysis.

Schedule

Week	**Dates**		**Topic, *problem set***
0.	5 Sept	W	welcome and preview – Jupyter and python
1.	10/12 Sept	M/W	molecular biology of genes, gene expression, and RNA-seq – *the case of the dead sand mouse*
2.	17/19 Sept	M/W	RNA-seq read mapping – doing controls on new programs – kallisto – the unix command line – *the adventure of the ten Arcs*
3.	24/26 Sept	M/W	data exploration and visualization – subsampling data – tidy data – pandas and seaborn – *the adventure of the missing phenotype*
4.	1/3 Oct	M/W	probability, likelihood, and inference – Laplace and Bayes – *a plague of sand mice*
5.	10 Oct	W	P-values and statistical significance – *Student's game night* – (Monday is Columbus Day)
6.	15/17 Oct	M/W	mixture models – K-means – expectation maximization – *a mixture of five*
7.	22/24 Oct	M/W	regression – regression as probabilistic inference – *the cycle of twelve*
8.	29/31 Oct	M/W	inferring hidden variables – multimapped reads and mRNA isoform expression estimation – expectation maximization again – *the return of the ten Arcs*
9.	5/7 Nov	M/W	cluster analysis – non-negative matrix factorization – *the moonlighting genes*
10.	14 Nov	W	differential expression analysis – *the adventure of the lost labels* – (Monday is Veterans Day)
11.	19/21 Nov	M/W	work patterns in computational research – artifacts & batch effect – *no sand mouse work this week; only turkey* – (Thanksgiving weekend)
12.	26/28 Nov	M/W	dimensionality reduction – principal component analysis – *2001, a space problem*
13.	3/5 Dec	M/W	t-SNE – *the Moriarty Brain Atlas*

6-10 Dec : Fini! reading period – no further assignments

11-20 Dec: Finals – no assignments

Prerequisites and background

There are no formal prerequisites. We expect students to come from different backgrounds – a mix of biologists, computer scientists, and applied mathematicians – and to have varying degrees of experience in writing code in Python. The course is designed to bring students up to speed in any area that they haven't seen much of before.

MCB112 is designed to be a course that could come *before* other rigorous coursework in biology, programming, statistics, and applied math, even though we do a mix of biology, programming, stats, and math in the course. Underlying the course's design is a philosophy that a biologist (indeed anyone) is perfectly capable of learning enough math, programming, and statistics to do sophisticated data analyses, but many of us have trouble building up abstract skills without first knowing why we're doing it. MCB112 emphasizes practical data analysis problems, and though at times you may feel like you've been dropped into the deep end of the pool, you will learn by example why math/programming/stats skills convey mutant superpowers for modern biology research. In part, we judge the success of MCB112 by how many of our students go on to take coursework in fields they wouldn't have dreamt of studying before.

However, it would be tough to come into the course with no background at all. We expect you to have course background in either the molecular biology side or the stats/math/programming/CS side. We do molecular biology at the level of LS1; Python programming around the level of CS109 or CS50; statistics around the level of STAT110 and STAT111; multivariate calculus and linear algebra around the level of MA21 or AM21; and a wee taste of data structures and algorithms. The more of these things you've taken already, the easier MCB112 will be.

Policies, expectations, grading

Most of the work is outside of class on your own, working on the weekly data analysis problems.

The Tuesday lecture each week covers fundamental background you need to know for that week's problem. We expect you to start thinking about your approach to the analysis problem after the Tuesday lecture.

The Thursday class time is more interactive and practical. We expect you to come with whatever questions you have from thinking about the problem so far, for discussion and review. We will walk you through approaches and resources you might want to tap. For example, especially in the early weeks of the course when people are

coming up to speed, we will show Python code examples of related problems.[1]

After that, you're working on your own on the week's problem. The instructors and teaching assistants are available for office hours and recitation sections for more individual discussion and questions.

Your solution to each week's problem is due at the start of the next week's Tuesday lecture (1pm). You submit your solution by email as a Jupyter notebook page.

We generally won't accept late work. We may consider rare extenuating circumstances on a case-by-case basis, and generally only if you've discussed the circumstances with us in advance. (Like, if you know you have to miss a week because you have to be out of town for something important, work that out with us beforehand.)

The grade is based entirely on the weekly data analysis problems. There are no exams or finals. Grades are not curved. We expect that everyone in the course will be able to solve every analysis problem proficiently, or at least competently – we will consider our work on the course to be a failure otherwise. Each problem will be graded on a scale of 1-5, where 1=proficient, 2=competent, 3=needs work, 4=insufficient effort, 5=zero effort, in 0.5 increments. Evidence of hard work alone, even with an unsuccessful solution, guarantees at least a 3 – most of the battle in learning data analysis is in investing time and thought in it, even if it comes slowly at first for some. The final letter grade is an unweighted average of the weekly problem grades, with A$\leq$ 1.33, A-$\leq$ 1.67, B+$\leq$ 2.0, B$\leq$ 2.33, etc.

Regular attendance is expected. It will be difficult to do the problem sets otherwise.

Lecture notes are made available online, and lectures are videotaped and available in Canvas.

You can use laptops and mobile devices to take notes in class.

[1] Because we expect some people will have never coded before, we'll aim to bring you up to speed quickly and practically by providing working Python code examples that you can study and adapt.

Materials and access

There is no required textbook for the course. Readings will be available online as PDFs.

You need to have access to a computer (laptop or otherwise) that you can install a Python scientific data analysis environment on.[2] If you do not have one, the course has a limited number of Mac laptops available for lending. You need to have Internet access; among other things, you will be submitting your work each week electronically as a Jupyter notebook page.

There are many on-line resources for learning Python, but for books, we recommend:

[2] We recommend the free Anaconda distribution from Continuum Analytics, `https://www.continuum.io/downloads`. You'll be ahead of the game if you install it ahead of time.

- Mark Lutz, *Learning Python*. O'Reilly and Associates, 2013.
- Wes McKinney, *Python for Data Analysis*. O'Reilly and Associates, 2012.

For an excellent (albeit formal/mathematical, and physics-oriented rather than biology-oriented) introduction to the fundamentals of data analysis, we recommend:

- D.S. Silvia and J. Skilling, *Data Analysis: A Bayesian Tutorial*. Oxford, 2006.

Academic integrity

You must do each week's data analysis project individually, on your own, rather than working collaboratively in groups. Your writing and your code must be your original work.

One goal of the course is to teach you how to understand and do biological data analysis yourself, without relying on interdisciplinary collaborations between people of disparate skills and interests. This is what the weekly data analysis projects are designed to push you to do.

As you're learning, though, you are free to talk with each other, and to consult any resource, and to study code from other sources. This is how we learn anything. It's how you'll learn at every step of your future. We trust you to know the difference between asking a question and copying an answer.

The principle is that although we encourage you to learn in any way that you prefer, each week you must reach the point where you can understand and execute your work independently and originally; this is what we'll be looking for.

We trust you. We expect you to act with honor and integrity. For example, it would not be hard to find previous versions of the course notes and problem sets online, but you should not go looking for them, and we trust you not to. We expect you are taking the course because you want to learn.

Accommodations for students with disabilities

Students needing accommodations because of a disability should present their Faculty Letter from the Accessible Education Office (AEO) and speak with an instructor by the end of the second week of the term (8 September) for us to be able to respond in a timely manner.