STAT 114: Introduction to Bioinformatics and Statistical Genetics

Spring 2024

Meeting Time: Mon/Wed 3:00pm-4:15pm (Science Ctr Lecture Hall A)
Course Site: https://canvas.harvard.edu/courses/133733

Instructor: James Xenakis jxenakis@g.harvard.edu

Course Description

STAT 114 is an introduction to bioinformatics and statistical genetics. The course will cover basic technology platforms, data analysis problems and algorithms. We will study statistical procedures commonly used in mammalian genetics (e.g., mouse and human). Topics include sequence alignment, differential gene expression analysis, QTL mapping and genome-wide association studies.

Prerequisites

Statistics 110 (Theoretical Probability).

Textbooks

This course will use the text:

- *Computational Biology - Genomes, Networks, and Evolution* (Kellis et al.), which is a freely available book from LibreTexts (https://libretexts.org/).

In addition, there will be recommended supplemental articles and videos (all freely available) for some class sessions. For those who are interested in more traditional textbooks, the following might be helpful, though are not required for this course:

- *Bioinformatics: Sequence and Genome Analysis*. David W. Mount
- *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Richard Durbin, Sean R. Eddy, Anders Krogh, Graeme Mitchison
- *Siegmund and Yakir (2007) The Statistics of Gene Mapping*. Springer

- *Lynch and Walsh (1998) Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Inc. MA.
- *Lange, K. (2002) Mathematical and Statistical Methods for Genetic Analysis, 2nd edition*. Springer, New York.
- *An Introduction to R, Venables and Smith. http://cran.r-project.org/doc/manuals/R-intro.pdf* Free!

Computing

The first portion of this course is focused heavily on using various commandline softwares that are an integral part of an RNA-seq pipeline. We will start by learning:

- How to use the Bash terminal, which is a command-line interface (CLI) in Unix and Unix-like operating systems, including Linux and macOS. It is also available on Windows, primarily through a feature called the Windows Subsystem for Linux (WSL).
- How to use Bash as a scripting language, allowing you to write scripts that automate (and document) tasks (i.e., your pipelines)
- How to use Vim, which is the greatest text editor of all time, and the only one worth knowing. I am composing text in Vim right now. In all seriousness, if you pursue bioinformatics, you will someday find yourself in a situation where Vim is the only editor available on some system you are using, so it is worth learning the basics.
- How to use a high performance computing cluster. Because bioinformatics requires using and manipulating extremely large files and running extremely memory-intensive software, the pipelines we learn will be implemented on a high performance computing cluster, rather than on your own machines (we will implement toy examples on your own computers). This is the first year Harvard will be using HUIT's new Open OnDemand platform for HPC resources, an Amazon cluster that uses SLURM (Simple Linux Utility for Resource Management) for job scheduling, another piece of software that we will learn in the first portion of this class.

Next, we will use the computing skills we have acquired to implement various aspects of an RNA-seq pipeline on the cluster:

- *FASTQC*: to check quality of raw reads from a sequencing machine
- *STAR*: to align the raw reads to a reference genome (e.g., mouse or human)
- *RSeqQC*: to validate the quality of the alignment
- *RSEM*: to estimate transcript abundance
- *Kallisto*: to perform pseudo-alignment to a reference genome and estimate transcript abundance

- *Salmon*: to perform quasi-mapping to a reference genome and estimate transcript abundance

The remainder of the course will heavily use the R language and environment (RStudio and RMarkdown), for example, to perform differential expression analysis of our processed RNA-seq data. For example, we will use the following R packages for this purpose:

- *limma*
- *edgeR*
- *DESeq2*

We will also explore some additional command line softwares when we enter the GWAS unit (e.g, PLINK).

Learning Environment

Lectures

Some lectures will be conducted in a 'semi-flipped' format; new content will be presented for about 45-55 minutes via annotated lecture slides, and the remainder of the class will be an interactive lab, where students will work through handout problems (a mix of conceptual, mathematical and software-based problems) in small groups. Teaching staff will be present to provide support.

Sections

Optional, but strongly recommended TF-led sections will be held weekly. Sections will be devoted to review of concepts, practice problems, and problem set preparation/questions.

Discussion Forum

Discussions through Ed are highly encouraged: https://edstem.org/. You will be able to link to Ed through the course website.

Office Hours

I (and our TF) will have regular weekly office hours (time and location TBD), where you are free to ask questions about the course material or problem sets. More general statistics questions are also encouraged. In addition, you will be able to schedule one-on-one office hour appointments with me (15 minutes per session). Please do not email me to schedule an appointment; instead, you can directly book an appointment using *Calendly*.

Attendance

Class attendance is highly encouraged, as lectures will not be recorded.

Collaboration

You are encouraged to discuss homework with other students, but you must write your final answers yourself in your own worlds. Solutions that are copied or paraphrased from someone else's work (including previous years' solutions if I happen to assign problems from past years) are not acceptable and will be treated as Honor Code violations. All computer output you submit must come from work that you have done yourself. You must list at the top of each problem set the names of all other students with whom you worked. All exams must be entirely individual work.

Course Work

Problem Sets

There will be approximately 5 problem sets, which will be assigned more or less bi-weekly. These assignments will be posted to the course website approximately 7-10 days before they are due. All assigments will be submitted on Gradescope by 11:59pm EST on the day they are due. Solutions will typically be posted to the website two days after the due date, and graded homeworks will be returned before the next problem set is due.

While you should attempt to complete all assigned problems, it is possible that only a subset of the problems will be graded. This subset will be: a) the same for all students, b) selected before the graders view any of the submitted problem sets, and c) not announced in advanced. Please always refer to the posted solutions to check your work on the ungraded problems.

Late problem sets will be accepted up to 48 hours after they are due, but will be penalized 25% of the total points per day. That is, if the problem set is worth 100 points, you would lose 25 points per day, where time is measured discretely - a problem set submitted at 12:00am will be considered one day late. But, to give flexibility, each student will be allowed 3 late days throughout the semester, with at most 1 day (24 hours) applied to any one problem set. Late days cannot be applied to your final project/presentation. Your lowest problem set score will also be dropped from your average.

In the event of serious illness or unexpected extenuating circumstances, I can provide additional flexibility. You will need to provide documentation from University Health Services or your resident dean.

Exams

There will be several (short) in-class quizzes throughout the semester. These will be fairly conceptual rather than technical, and are simply intended to ensure that you are keeping up with the class material. They will take only a portion of the class period, and are not intended to be particularly difficult. There will be no final exam in this class.

Final Project

The bioinformatics landscape changes very quickly, and many skills you learn will quickly become outdated. For this reason, the ability to learn new tools and pipelines is paramount. Fortunately, it has never been easier to learn new skills! For this reason, in lieu of a final exam, we will have a final project in which you will research a topic that we have not covered in the class (it could be a technique that you perform in your lab), and present it to the class, for approximately half and hour. The project will begin with the submission of a project proposal mid-semester, and the final project should include a written written write-up and an accompanying tutorial that will allow another student to follow (and recreate) a comprehensive pipeline.

This is not the only acceptable format for this presentation - some alternative ideas are included at the end of this syllabus.

The review of your tutorial by another student in the class will be part of your grade.

Grading

Your final raw score for the course will be computed using the weights in the table below and an associated letter grade assigned based on the class distribution.

Component	Weight
Problem Sets	40%
Quizzes	15%
Final Project	40%
Tutorial Review	5%
Total	100%

Regrade Requests: All regrade requests must be made within a week of receiving your initial score. A regrade request submission consists of a written explanation detailing which questions are the main focus of the regrade request and explaining why your answer merits additional credit. Please submit any regrade requests directly to Gradescope.

Accessibility

Harvard College is committed to working with all students. To request accommodation for a disability, please contact the Accessible Education Office (AEO) (https://aeo.fas.harvard.edu/). Advance notice and appropriate documentation (in the form of a Faculty Letter from the AEO) are required for accommodations. Please speak with me by the end of second week of class (9/15/23) so that we can respond in a timely manner. All discussions will, of course, remain confidential.

Generative AI

This course encourages students to explore the use of generative artificial intelligence (GAI) tools such as ChatGPT to gain conceptual and theoretical insights, as well as assistance with coding. In

addition, some problems (explicitly labeled) will *require* the use of ChatGPT, as it is an important tool that you should be experimenting with. Any use of GAI must be appropriately acknowledged and cited.

The text in any homework responses must be entirely your own, in our own words, reflecting your own understanding and reasoning. Cutting and pasting text generated by GAI should be considered equivalent to cutting and pasting from a book, or from someone else's solutions - i.e., it will be considered plagiarism, and therefore academic misconduct.

We encourage you to share your experience with these technologies on Ed, especially in relation to how you think it is making you a more effective learner, or instances where it might have led you astray (these technologies can and do "hallucinate"), etc.

Course Schedule

A (tentative) course schedule appears on the following page.

Class Schedule

Date	Unit		Subunit	Class
01/22 (Mon)	1	Intro to UNIX		Logistics, Cluster & Bash
01/24 (Wed)				Shell Scripts & Vim
01/29 (Mon)	2	Molecular biology		A statistician's guide to biology
01/31 (Wed)	3	RNA-seq		Sequencing
02/05 (Mon)				Quantification
02/07 (Wed)	4	Regression		Linear regression
02/12 (Mon)				Generalized linear regression
02/14 (Wed)	5	Differential expression		limma
02/19 (Mon)				No class (President's Day)
02/21 (Wed)				EdgeR
02/26 (Mon)				DESeq2
02/28 (Wed)	6	Further analysis		Clustering
03/04 (Mon)				Dimension reduction
03/06 (Wed)				Classification
03/11 (Mon)				No class (Spring recess)
03/13 (Wed)				No class (Spring recess)
03/18 (Mon)				t-SNE
03/20 (Wed)	7	A deeper dive		EM algorithm
03/25 (Mon)				Hidden Markov models
03/27 (Wed)	8	Epigenetics, DNA Methylation		Epigenetics, DNA Methylation
04/03 (Wed)	9	GWAS		GWAS I
04/08 (Mon)				GWAS II
04/10 (Wed)	10	Mapping in test populations		Mapping in test populations (1)
04/15 (Mon)				Mapping in test populations II
04/17 (Wed)	11	Proteomics		Proteomics (1)
04/22 (Mon)				Proteomics (2)
04/24 (Wed)	11	Final projects		
TBD (Final period)				

Final Project Ideas

Here are some potential ideas that you might consider for your final project/presentation. This is by no means a comprehensive list - it is simply meant to whet you appetite. You all have different skills and interests that I would love to have you share with the class!

1. The pipeline that we study for RNA-seq analysis in this course took us from sample preperation through differential expression analysis. However, the initial steps (RNA extraction, sample and cDNA library preparation, and sequencing) were treated as "black boxes" (being a statistician, I have never performed these steps). That might not be the case for you! You could prepare some videos explaining these steps in detail.

2. We introduced BLAST (Basic Local Alignment Search Tool) in our introduction to alignment algorithms, but another major value of BLAST is its use in searching biological databases. If this is a tool you use regularly, or are interested in learning more about, you could do a presentation on this.

3. In this course, we used the R package *limma* in the context of RNA-seq data analysis. But this package was originally designed to handle microarray data. Although the popularity and prevalence of microarray technology has declined with the advent of next-generation sequencing (NGS), it remains an imporant source of historical data. You could prepare a presentation on microarrays.

4. Seurat and Scanpy are two open-source software tools used for the analysis of single-cell RNA sequencing (scRNA-seq) data. Here, Lior Pachter (one of the authors of TopHat) demonstrates that implementation of seemingly identical functions in the two packages yield discordant answers (https://twitter.com/lpachter/status/1693346165096620050). There are many similar examples you could document and attempt to reconcile.

5. You could do a presentation on some aspect of epigenetics - for example, DNA methylation and its effects, as well as techniques for measuring DNA methylation if we don't end up doing this in class (it is currently in the syllabus, but that is subject to change).

6. An introduction to Mendelian Randomization: essentially, this entails the use of genetics variants as instrumental variables

7. You can do a presentation on some technology that we did not cover in class, for example:

   (a) ChIP-seq (Chromatin Immunoprecipitation sequencing), which is a method for investigating interactions between proteins and DNA within the cell. This technique is particularly useful for studying how transcription factors and other DNA-binding proteins regulate gene expression, and for mapping the locations of histone modifications across the genome. These interactions play a critical role in many biological processes, including gene regulation, cell differentiation, and the response to environmental signals.

   (b) An introduction to single cell RNA sequencing analysis (scRNA-seq)

(c) scATAC-seq (single-cell Assay for Transposase-Accessible Chromatin using sequencing), a technique used in genomics to study chromatin accessibility at the single-cell level. This method provides insights into how the regulation of chromatin structure affects gene expression in individual cells. Understanding chromatin accessibility is crucial for comprehending the mechanisms of gene regulation, cellular differentiation, and the development of various diseases

(d) Hi-C - a technique used to study the three-dimensional architecture of the genome that enables understanding the organization and interaction of chromosomes within the cell nucleus. This is important for understanding gene regulation, genome function, and the cellular processes that underlie health and disease.