EE 372: Data Science for High-Throughput Sequencing

Course Overview

Course Description

Extraordinary advances in sequencing technology in the past decade have revolutionized biology and medicine. Many high-throughput sequencing based assays have been designed to make various biological measurements of interest. This course explores the various computational and data science problems that arises from processing, managing and performing predictive analytics on high throughput sequencing data. Specific problems we will study include genome assembly, haplotype phasing, RNA-Seq assembly, RNA-Seq quantification, single cell RNA-seq analysis, multi-omics analysis, and genome compression. We attack these problems through a combination of tools from information theory, combinatorial algorithms, machine learning and signal processing. Through this course, the student will also get familiar with various software tools developed for the analysis of real sequencing data. The target audience for the course include

1. students specializing in information theory/algorithms/signal processing/machine learning who want to learn of applications in biology and get exposure to real data
2. students specializing in computational biology, who want to strengthen their knowledge of basic information theory/signal processing/machine learning

Communication

Course news and assignments will be posted at `ee372.stanford.edu`, which redirects to the course's GitHub website. Each assignment and set of lecture notes will have its own page, and students are encouraged to ask and answer questions by leaving or replying to comments on these pages.

Prerequisites

- Undergraduate level probability
- Some programming experience. We will be using Python.
- Some undergraduate background in algorithms would be beneficial
- No prior background in biology will be assumed

Lecture Times

Monday, Wednesday 3:00 PM - 4:20 PM at McCullough 115
Lab hour: Friday (exact time and location TBA)

Course Staff

Instructor: David Tse (`dntse@stanford.edu`)
Teaching assistants: Govinda Kamath (`gkamath@stanford.edu`) , Jesse Zhang (`jessez@stanford.edu`)
Office hours: 4:20pm-5:05pm MW for instructor, 2:00pm-3:00pm M for teaching assistants

Course Grading

The grading for the course will be broken down as follows:

- Attendance 10%
- Scribing 10%
- Assignments 40%
- Project 40%

Attendance

Students are encouraged to participate in class either during lecture or by leaving comments on material posted at the course website.

Scribing

Each student will be responsible for scribing a lecture. To ensure that the notes will be available for students currently in the course, **scribed notes are due within 72 hours after lecture** (no late submissions accepted). A Google Doc will be used for reserving lectures for scribing.

Assignments

There will be 4 assignments. The assignments will involve a theory component and a programming component. The programming component is aimed at exposing students to the messiness involved in real data and various tools used in practice. The programming assignments will include

- experiments demonstrating biases of different types in various types of data,
- implementing simple algorithms for assembly, alignment, and quantification,
- using popular software packages to perform simple experiments.

All programming assignments will require require only laptop-level computing. The main language used to code will be Python. UNIX/LINUX/OS X may be needed for some of the software packages. We recommend that windows users use the Corn cluster.

Projects

Projects can be theoretic or practical in nature (ideally a mix of the two). Additional details and a list of possible projects will be put up shortly. Students can also come up with project topics that they are interested in (in consultation with the teaching staff).