CSCI1810 - Computational Molecular Biology
Fall 2016

Professor Sorin Istrail
Department of Computer Science, Brown University

August 18, 2016

- Tues./Thurs. 2:30-3:50, CIT 243 (SWIG)
- Course website:

  `http://cs.brown.edu/courses/csci1810`

- Questions? Email to

  `cs181tas@cs.brown.edu`

- TAs
  - Cyrus Cousins, graduate TA
  - Tyler Devlin, Head undergraduate TA
  - Natalie Tsvetkova, undergraduate TA
  - Hai Tran, undergraduate TA
  - Sudheesha Perera, undergraduate TA

1 Course Description

The aim of this course is to provide an introduction to Computational Molecular Biology. The course is organized into five chapters: Sequence Alignment, Combinatorial Pattern Matching, Phylogenetics Trees, Hidden Markov Models, and Genome Assembly. Each chapter is devoted to a class of basic computational problems related to the analysis of DNA, RNA and protein sequences and their molecular biology function. Our journey in each chapter is driven by a set of most beautiful algorithms. "Beautiful" algorithm here refers to an algorithm that is rigorous, practical and with ellegant simplicity that makes it also easy to implement. These algorithms are the state of the art of the theory and practice of solvng the computational problems presented in the corresponding chapter.

In addition to the beautiful algorithms, each chapter contains a *Foundations* section that presents computer science and statatistical theoretical results that led to the invention of the algorithms with their underlying data structures, the mathematical analysis of their performace, and at times, the exciting story of the researchers quest for algorithm optimality (speed).

Prerequisites. One of: CS16, CS18 or CS19. Recommended: CS22, or some other course that introduces concepts from discrete math and probability. Course overrides are available at the instructor's discretion.

2 Course Format

Meeting times and place:
CS181 lectures are during K hour

Tuesday and Thursday, 2:30-3:50pm, in the SWIG, CIT 243.

You are expected to attend all classes. Class lecture notes will be made available.

Homeworks will be given weekly. Overall, there will be three programming assignments. Biological, life sciences and medical students (bio students) that do not write code will have special homework questions assigned to them to compensate for the programming assignments. We will work together with the bio students that do not write code to still advance their computational skills in alternative ways (e.g., running and tinkering Mathematica code).

Grading

- Class participation 5%
- Homework 60% - the programming assignments will count for 30%
- Midterm exam 20% (in class)
- Final exam 15% (take home)

Grades are determined by the overall performance according to these measures. You are not competing with your classmates. The professor, at the end of the class, awards a *Pastiche Pie slice* award to the student (s) (one or two) with the overall most impressive performace in the class, especially in solving the extra credit problems of the homework, as judged by the teaching team – theTAs and the professor.

Readings: There is no textbook for this class. Lecture notes of the lectures will be provided. The textbook "An Introduction to Bioinformatics Algorithms" by Neil Jones and Pavel Pevzner, MIT Press, 2004 would be a useful book for additional readings for this class. On the class website there are comments about other good textbooks for the area and their specialized focus. part of our work for the class there will be required articles readings, as well as suggested ”seminal readings” that would complement and enhance the lecture content of the class.

Graduate credit can be obtained by work on a final project selected in consultation with the professor.

Typically, the students find that CS181 requires about 10 hours of coursework a week.

3 Collaboration Policy and Grading

CS181 has a Collaboration Policy based on Brown’s Academic Code of Conduct, but is specific to CS181. A copy of this policy is available on the course website. Overall, while students may discuss concepts in the context of the lecture material, any collaboration on any stage of a project or assignments (problem solving, designing, debugging or programming) is a vialation of our policy.

The course staff takes violations of the collaboration policy very seriously and will prosecute them with the standing committee on the academic code as necessary.

Like many courses in the Computer Science department, CS181 relies heavily on the role of the Undergraduate Teaching Assistants and Graduate Teaching Assistants. In addition to holding TA hours, the CS181 Undergraduate and Graduate TAs also grade all student work under the supervision of the Head TA, Graduate TA, and the professor; the final grades are determined by the professor.

4 Lecture Topics

1. Chapter 1: **Sequence Alignment**
   Algorithms:

   - Needleman-Wunsch Algorithm (global alignment)
   - Smith-Waterman Algorithm (local alignment)

   Foundations: Margaret Dayhoff and similarity matrices statistics (intro); dynamic programming; protein structure alignment as gold standard for sequence alignment; k-mers and Poisson statistics; DNA, RNA and protein sequence alignment; gaps in alignment; multiple alignment.

2. Chapter 2: **Combinatorial Pattern Matching**
   Algorithms:

    - Knuth-Morris-Pratt Algorithm (finding a string pattern in a text)
    - Aho-Corasick Algorithm (generalized KMP)
    - BLAST Algorithm (intro)
    - Weiner Algorithm (intro) (Position Trees and Suffix Trees)

    Foundations: Finite-automata and regular expressions; DNA motifs of the regulatory regions of genes; approximate string matching; patterns in DNA, RNA and protein sequences.

3. Chapter 3: **Phylogenetic Trees Algorithms:**

    - UPGMA Algorithm (evolutionary distance matrices with uniform clock)
    - Neighbour-Joining Algorithm (general evolutionary distance matrices)
    - Parsimony Algorithm (minimizing the number of mutations)

    Foundations: Evolutionary models; tree alignment and maximum likelihood and other probabilistic models.

4. Chapter 4: **Hidden Markov Models (HMMs) Algorithms:**

    - Forward Algorithm for PB 1. “Computing the probability” (model scoring)
    - Viterbi Algorithm for PB 2. “Best Explanation” (Viterbi maximum likelihood)

    Foundations: Probabilistic finite automata; finding genes in genomes (intro); PB. 3 ”Learning (intro)”

5. Chapter 5: **Genome Assembly (Intro) Algorithms:**

    - De Brujin Assembly Algorithm (de Brujin graphs and Eulerian paths)
    - Idury-Waterman Assembly Algorithm (intro) (Poisson statistics of DNA k-mers)

    Foundations: Sequencing technologies: Sanger, Illumina; statistical error models; Lander-Waterman statistics (intro).

5 Finally ...

Please come to the professor’s office hours as well as to the TAs office hours.