CSCI1820 - Algorithmic Foundations of Computational Biology
Spring 2017

Professor Sorin Istrail
Department of Computer Science, Brown University

February 2, 2017

• Tues./Thurs. 2:30–4:50pm, CIT 241 (SWIG)

• Course website:
  http://cs.brown.edu/courses/csci1820

• Questions? Email to
  cs1820tas@lists.brown.edu

• TAs
  – Pran Chanthrakumar, Head undergraduate TA
  – Marko Fejzo, undergraduate TA
  – Alice Chu, undergraduate TA
  – Jonathan Chang, undergraduate TA
1 Course Description

The aim of this course is to provide computer science and mathematical sciences foundations, i.e., algorithmic foundations, for Computational Biology. The course is organized into five chapters:

1. Sequence Alignment Algorithms and the BLAST Algorithm
2. Genome Assembly Algorithms and Haplotype Assembly Algorithms
4. The Regulatory Genome and Gene Regulatory Networks Algorithms
6. A (tentative) additional chapter: Protein Folding Algorithms (An Introduction)

Each chapter is devoted to a class of fundamental computational problems of genomics, that is related to the analysis of DNA, RNA, protein sequences and protein structures and their molecular biology function. Our journey in each chapter is driven by a set of most beautiful algorithms presented together with their theoretical algorithmic foundations, in a comprehensive, almost complete analytical detail. "Beautiful" algorithm here refers to an algorithm that is rigorous, practical and with elegant simplicity that makes it also easy to implement. These algorithms are among those presenting the state of the art of the theory and practice of solving the computational problems presented in the corresponding chapter. The Algorithmic Foundations section in each chapter presents in detail the biological problems discussed and theoretical computer science and statistical results that led to the invention of the algorithms that resolve the modeled biological problems. The algorithms are presented together with their underlying data structures, mathematical analysis of their performance, and at times, the exciting story of the researchers quest for algorithm optimality (speed). The overall work in the class will help in providing an algorithmically advanced journey through today’s most indispensable software genomics tools of a bioinformatician and computational biologist.

**Prerequisites.** One of: CS16, CS18 or CS19. Recommended: CS22, or some other course that introduces concepts from discrete math and probability. Although CS 181 is not a required course for CS182, it provides a very good preparation, and it covers important topics not covered in CS182, such as combinatorial pattern matching algorithms and phylogenetic trees algorithms. Course overrides are available at the instructor’s discretion.
2 Course Format

Meeting times and place: CS182 lectures are during:

**Tuesday and Thursday, 2:30-3:50pm, in the SWIG, CIT 241.**

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

**Homeworks** will be given weekly. Overall, there will be five 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.

**Grading**

- Class participation 5%
- Homework 65%
- Midterm exam 15% (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 performance in the class, especially in solving the extra credit problems of the homework, as judged by the teaching team – the TAs and the professor.

**Readings:** There will be required 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.

3 Collaboration Policy and Grading

CS182 has a Collaboration Policy based on Brown’s Academic Code of Conduct, but is specific to CS182. 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 violation of our policy.

**Late homework policy:** You will receive 5 late days for use throughout the course. All 5 may be used on one project as well. Please make the TAs aware that you are using late days, otherwise you will be penalized.
Like many courses in the Computer Science department, CS182 relies heavily on the role of the Undergraduate Teaching Assistants. In addition to holding TA hours, the CS182 Undergraduate TAs also grade all student work under the supervision of the Head TA and the professor; the final grades are determined by the professor.

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.

4 Lecture Topics

• Chapter 1: **Sequence Alignment Algorithms and the BLAST Algorithm**

Algorithms

– Needleman-Wunsch Pairwise Sequence Global Alignment Algorithm (global alignment)
– Smith-Waterman Pairwise Sequence Local Alignment Algorithm (local alignment)
– The Hirshberg Algorithm for Dynamic Programming alignment in linear space (local and global alignment algorithms in linear space)
– Multiple Alignment Algorithms heuristics

Statistical Theory

– The Karlin-Altschul statistical theory of local alignment (un-gapped local alignment)
– Substitution Matrices and Information Theory - The Margaret Dayhoff Matrices (Margaret Dayhoff is known as the ”mother and father of Bioinformatics”)
– Michael Waterman, founding father of Bioinformatics, pioneer of alignment algorithms and of statistical theory of alignment

• Chapter 2: **Genome Assembly Algorithms and Haplotype Assembly Algorithms**

Algorithms

– De Bruijn Assembly Algorithms (de Bruijn graphs and Eulerian paths)
– Idury-Waterman Assembly Algorithm (Poisson statistics of DNA k-mers)
– EULER Algorithm of Pevzner-Tang-Waterman
– Celera Genomics Assembly Algorithm and the software engineering pipeline (an overview)
– Hamilton Smith, Nobel Prize for Restriction Enzymes, pioneer of genome sequencing at Celera Genomics

Statistical Theory

– Introduction to Genome Assembly – The Sequence of the Human Genome
– Statistical Theory of Genome Assembly: The Lander-Waterman Formulas
  * Three fundamental problems and the derivation of their solution formulas
    1. Problem 1: What is the mean portion of the genome covered by the contigs of the assembly?
    2. Problem 2: What is the mean number of contigs of the assembly?
    3. Problem 3: What is the mean contig size of the assembly?

• Chapter 3: Gene Finding Algorithms based on Hidden Markov Models Algorithms

Algorithms and Statistical Theory

• An overview of the DNA Structure of a Gene
• Hidden Markov Models for Gene Finding: Three Fundamental Problems and their Algorithmic Solutions
  1. Problem 1: THE MODEL EVALUATION PROBLEM (computing the probability)
  2. Problem 2: THE DECODING/REVEALING ”THE HIDDEN PATH” PROBLEM (”Best explanation”/Viterbi maximum likelihood)
  3. Problem 3: THE LEARNING PROBLEM

• Chapter 4: The Regulatory Genome and Gene Regulatory Networks

Algorithms and Statistical Theory

– Motifs and cis-regulatory sequence structure
– Position Weight Matrices Algorithms and their statistical theory
– The Regulatory Genome as an information processing system (the ”Genomic Computer”)
– Logic functions of the genomic cis-regulatory code
– Modeling Evolution: The Parable of the Two Watchmakers: Hora and Tempus
– Eric Davidson, pioneer of regulatory genomics and gene regulatory systems
• Chapter 5: Data Structures and Algorithms for Sets of Sequences: Suffix Tree Algorithms and Burrows-Wheeler Transform Algorithms

Algorithms

1. Linear time Suffix Tree Algorithm
2. Burrows-Wheeler Transform Algorithm and the Positional BW Transform Algorithm
3. Alberto Apostolico, pioneer of Combinatorial Pattern Matching Algorithms in Bioinformatics

• Chapter 6 (tentative): Protein Folding Algorithms – An Introduction
The Computational Protein Folding Problem is one of the grand challenge problems in computational biology, biotechnology, biophysics, biochemistry, mathematics, statistics, and computer science. We will discuss some puzzle problems related to lattice protein folding problems that witness the exceedingly difficult computational problems of this area.

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