

MATH578A: Computational Molecular Biology (Spring 2019)

Modern molecular biology involves unique forms of data. These data are notable for their size: many computational challenges in molecular biology cannot be solved simply by employing more hardware. These data also continually demand novel abstractions, both for data structuring and characterizing the associated algorithmic problems. The goals of this course are:

- Introduce the classical problems and fundamental algorithms in computational molecular biology.
- Establish a robust algorithmic toolkit enabling students to efficiently solving computational challenges encountered during data analysis.
- Teach students to recognize which techniques are appropriate for solving particular computational problems, to recognize when obvious approaches will fail, and when approximations should be considered.
- Show students how algorithmic theory meets practical performance through implementation and experimentation on real molecular biology data.
- Provide students with the foundation for conducting research designing novel algorithms for emerging computational problems in molecular biology.

Text books: “*Algorithms on Strings, Trees and Sequences*” by Dan Gusfield. Each lecture is also accompanied by at least one original research article that will be posted on Blackboard.

Evaluation: Evaluated is based on one midterm worth 30%, a final exam (held the last day of class) worth 40%, and two assignments, each worth 15%. Assignments include both theoretical and practical exercises involving coding.

Week	Date	Topic
1	01/08/19	Course overview and review of asymptotic analysis
	01/10/19	Exact string matching (1): the Z-algorithm
2	01/15/19	Exact string matching (2): the Boyer-Moore algorithm
	01/17/19	Numerical encodings for biological sequences
3	01/22/19	The Rabin-Karp algorithm
	01/24/19	Approximate string matching: exclusion methods
4	01/29/19	Sorting strings: radix sort and bucket sort
	01/31/19	Suffix trees (1): definitions and applications
5	02/05/19	Suffix trees (2): McCreight’s construction algorithm
	02/07/19	Lowest-common ancestor queries in trees
6	02/12/19	Suffix arrays (1): basic construction, efficient searching
	02/14/19	Suffix arrays (2): the Skew algorithm
7	02/19/19	The Burrows-Wheeler transform
	02/21/19	Data compression; Ziv-Lempel
8	02/26/19	Pairwise alignment (1): similarity, distance, indels, the Needleman-Wunsch algorithm
	02/28/19	Midterm
9	03/12/19	Pairwise alignment (2): the Smith-Waterman algorithm, affine and general gap scoring
	03/14/19	Pairwise alignment (3): Hirschberg’s algorithm, bounded distance alignment
10	03/19/19	Multiple alignment (1): sum-of-pairs scoring, general recurrences
	03/21/19	Multiple alignment (2): center-star approximation, progressive alignment
11	03/26/19	Genome rearrangements and sorting by reversals
	03/28/19	Genome rearrangements and sorting by reversals
12	04/02/19	Whole-genome alignment: graph representations and algorithms
	04/04/19	Sequence assembly (1): shortest superstrings, overlap-layout-consensus, greedy algorithms
13	04/09/19	Sequence assembly (2): de Bruijn graphs, Eulerian paths
	04/11/19	Bloom filters and compact de Bruijn graphs
14	04/16/19	Phylogenetic trees (1): topology inference from distance data
	04/18/19	Phylogenetic trees (2): maximum parsimony
15	04/23/19	Genomic annotations: interval trees and static data structures
	04/25/19	RNA folding: Nussinov’s algorithm