

# Course Syllabus

## CS 238: Algorithmic Techniques in Computational Biology

### Spring, 2026

**Introduction:** Due to the revolutionary development of genomics, the sheer volume of digital genetic information now available is surpassed only by the potential for biological and medical discovery. The exploration of this information is critically dependent upon the development of advanced computational methods for data analysis. From this dependency, a young field of research, *Computational Molecular Biology* (or simply *Computational Biology*), emerged in recent decades.

The primary reason that molecular biology is of great interest to computer scientists is that genes, proteins, chromosomes, and genomes can all be viewed, at one level, as simply strings of symbols from a finite alphabet (the alphabet  $\{C, G, A, T\}$  of the four nucleotides or the alphabet of 20 amino acids). At this level, they are similar to textual documents with a different alphabet; thus, many of the techniques of “stringology” are applicable to them (indeed, genomic issues were responsible for much of the development of this subfield of Computer Science). The analogy can be carried at least one step further. Just as a document, such as this syllabus, has a structure, so too does a chromosome; a chromosome is a sequence of genes and noncoding regions and a gene is a sequence of exons and introns. A second reason for computer scientists’ interest is that various completed (or ongoing) large-scale sequencing, mapping and profiling projects in life sciences such as the Human Genome Project, ENCODE, modENCODE, HapMap Project, 1000 Genome Project, Human Microbiome Project, 4D Nucleome Program, etc. have produced an enormous amount of digital data and have raised many *intractable* computational questions of an optimization flavor. While many biologists have intuitively realized that a problem such as multiple sequence alignment or phylogenetic inference is intractable due to its combinatorial nature and, therefore, developed heuristic algorithms, few proofs of intractability and even fewer proofs of the quality of the results of the heuristic algorithms have been established for these combinatorial optimization problems.

**Course format:** The course will be focused on the design and analysis of efficient (combinatorial) algorithms for important problems in computational molecular biology. The format of the course will include lectures by the instructor, class discussion, directed reading, and student presentations or projects. The exact format will depend on the size of enrolment and student background. We emphasize mathematics, algorithms, and data structures instead of biological implications and applications, although some relevant biological background and motivations will be discussed. We may also have some guest speakers to talk about their research problems.

**Schedule:** TR 11:00-12:20pm, WCH 143. The lectures are only offered in-person, but recordings of similar lectures from the past can be made available on Canvas/Yuja upon request.

#### Textbooks:

*An Introduction to Bioinformatics Algorithms* (required), Neil C. Jones and Pavel Pevzner, The MIT Press Series on Computational Molecular Biology, 2004, 4th Ed. ISBN 0-262-10106-8. Available at the Bookstore with the same discount as on Amazon.

*Bioinformatics Algorithms: An Active Learning Approach* (optional), Phillip Compeau and Pavel Pevzner, Active Learning Publishers, 2018. ISBN: 978-0-9903746-3-3

**Lecture Notes:** The slides used in class (in PPT or PDF or web formats) as well as some supplementary material can be found on the class homepage <http://www.cs.ucr.edu/~jiang/238-homepage.html>

#### Optional reference books:

*Genome-Scale Algorithm Design*, Veli Makinen, Djamel Belazzougui, Fabio Cunial and Alexandru I. Tomescu. Cambridge University Press, 2015. ISBN 978-1-107-07853-6

*Algorithms for Strings, Trees, and Sequences: Computer Science and Computational Biology*, Dan Gusfield, Cambridge University Press, 1997. ISBN: 0-521-58519-8

*Fundamental Concepts of Bioinformatics*, Dan Krane and Michael Raymer, Benjamin Cummings, 2003. ISBN: 0-8053-4722-4

*Molecular Biology: Not Only for Bioinformaticians*, Wieslawa Widlak, Lecture Notes in Comp Sci 8248, 2013. Downloadable at <https://link.springer.com/book/10.1007%2F978-3-642-45361-8> or via eLearn/Canvas (under the Reference Books module) or reserved books at the Science Library.

*Rosalind*: An online platform for learning bioinformatics through problem solving. See [www.rosalind.info](http://www.rosalind.info).

**Instructor:** Tao Jiang, WCH 336, phone: x22991, email: [jiang@cs.ucr.edu](mailto:jiang@cs.ucr.edu). Office hours: TR 2:00-3:00pm. Zoom sessions can be made available upon request.

**Voluntary reader:** Jingong Huang, WCH 110, email: [jhuan381@ucr.edu](mailto:jhuan381@ucr.edu).

**Prerequisite:** CS218 (Design and Analysis of Algorithms) **or** CS141 (Algorithms and Data Structures) **and** CS/Math 111 (Finite Math), **or** equivalent knowledge. No background in biology is required.

**Topics covered:** introduction to molecular biology (1.5 lectures), physical (restriction) mapping (1.5 lectures), motif finding, regulatory signal recognition, and probabilistic algorithms (2.5 lectures), genome rearrangement by greedy and approximation methods (1.5 lectures), sequence alignment by dynamic programming and divide-and-conquer methods (2 lectures), multiple sequence alignment (1 lecture), gene prediction (1 lecture), fragment assembly by graph methods and shortest common superstrings (1 lecture), string matching and suffix trees (1 lectures), reconstruction of evolutionary trees (2 lectures), comparison of annotated sequences and RNA secondary structures (1 lecture), gene expression analysis and clustering methods (1 or 2 lectures), and selected topics (*e.g.* haplotype inference, comparative genomics). The actual topics may change according to progress.

**Homework assignment:** There will be three assignments to help digest the material learned in lectures. Most of the assignments will involve analytical work (*i.e.*, design and analysis of algorithms), although some may require serious programming effort.

**Term presentation:** Each student is required to give a 30-minute presentation on a topic selected from a list of topics provided by the instructor later in the quarter. Examples of possible topics include follow-up discussion on a topic covered in lectures, survey on something not covered in lectures including new technologies and data analysis challenges, original solution to a technical problem posed in class, nontrivial improvement on a known result, proposal of new methods, practical considerations of theoretical results, *etc.* The presentations will be held in the last three or two weeks of the quarter.

**Reading assignment:** The students will be expected to review, in advance, the material to be covered in class. In addition to the text and reference books, there will be handouts of papers and book chapters from time to time.

**Grading:** Homework 50%, term presentation 40%, and class participation 10%.

**Academic dishonesty:** You are expected to work alone on all your assignments and presentations, although it is common to rehearse your presentations in front of friends and classmates and obtain their feedback. For a detailed campus policy on academic dishonesty, see <http://conduct.ucr.edu/policies/academicintegrity.html> In particular, *copying solutions from the Internet/AI or books or any other sources without explicit citations and explanations is prohibited!*

**Class mailing list:** Please subscribe to the class mailing through the class homepage ASAP and remember to confirm the subscription.