جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Course Syllabus: Algorithms in Bioinformatics - CS 249

Division	Computer, Electrical and Mathematical Sciences & Engineering
Course Number	CS 249
Course Title	Algorithms in Bioinformatics
Academic Semester	Fall
Academic Year	2021
Semester Start Date	08/30/2020
Semester End Date	12/15/2020
Class Schedule (Days & Time)	07:00 AM - 08:30 AM | Wed , 04:45 PM - 06:15 PM | Mon

Instructor(s)				
Name	**Email**	**Phone**	**Office Location**	**Office Hours**
Robert Hoehndorf	robert.hoehndorf@kaust.edu.sa	+966128081643	4222, 3, Ibn Sina (bldg. 3)	On request.

Teaching Assistant(s)	
Name	**Email**
Maxat Kulmanov	maxat.kulmanov@kaust.edu.sa

Course Information	
Comprehensive Course Description	How do we sequence and compare genomes? How do we identify the genetic basis for disease? How do we construct an evolutionary Tree of Life for all species on Earth? When you complete this course, you will learn how to answer many questions in modern biology that have become inseparable from the fundamental

	algorithms used to answer these questions. This course features dozens of algorithms and challenges you to implement the bioinformatics algorithms that you will encounter along the way in dozens of automatically graded coding challenges that can be completed in any programming language. In addition, there will be group discussions and exercises that relate what you learn to recent trends in bioinformatics research. This course will be taught using an interactive textbook in a "flipped classroom" style where you read book chapters and watch videos at home and teaching time is reserved for discussion of problems and ways to improve the algorithms, reviewing your solutions, and determining how you can apply them to new problems in biology.
Course Description from Program Guide	The course will introduce methods, algorithms, and data structures used in bioinformatics. The main focus will be on String algorithms used in sequence alignment and genome assembly, algorithms used in structural bioinformatics, pattern discovery in sequence data, phylogenetics, biological networks and graphs, and knowledge representation in biology. Broad topics will include dynamic programming, linear programming, tree and array structures for String matching, graph structures for genome alignment, and network algorithms to cluster and align networks.
Goals and Objectives	The course will focus on the algorithms used in bioinformatics, inspired by challenges in biology and solved through computational means. At the end of the course, students will know different algorithms to align biological sequences, assemble genomes, construct phylogenetic trees, identify causative variants involved in disease or other phenotypes, determine the structure of proteins, and many more. In addition to the algorithms, students will gain an understanding of the computational challenges involved in analyzing large (genome-scale) biological datasets, analyze the comlexity of bioinformatics algorithms, and exploit domain-specific information to improve algorithms.
Required Knowledge	Students should be good technical thinkers and have a strong introductory knowledge of programming. No biological background is necessary.
Reference Texts	We use the interactive textbook "Bioinformatics Algorithms: An active learning approach" by Phillip Compeau and Pavel Pevzner. For the course, we use the interactive version of the textbook available at https://stepik.org/course/73755.
Method of evaluation	**50.00**% - Homework /Assignments **10.00**% - Attendance and Participation **15.00**% - Midterm exam **25.00**% - Final exam
Nature of the assignments	There are weekly programming assignments that require implementation of an algorithm and submitting the results to the Stepik platform. To pass the

	course, 80% of the programming assignments must be completed successfully. Each week, students will present their solution to the programming assignments in class and we discuss ways to improve the solution as a group. Participation in these discussion is mandatory and constituted 10% of the final grade. Programming assignments will be graded based on their presentation in class, amount of assignments completed (in addition to the 80% which are mandatory), and the quality of the algorithm implemented; algorithm quality is determined by the analysis of the algorithm as presented in class, and will depend on considerations of correctness and complexity.
Course Policies	The course will be taught online-only or in a hybrid model with voluntary presence in the classroom and possible participation online. The majority of teaching time is dedicated to in-class discussions of problems and possible solution. No more than 2 classes may be missed without cause; in case of a missed class, homework assignments must still be completed. 80% of programming assignments must be completed and submitted to Stepik to pass the course. The course lives from being able to participate in discussions during teaching hours; therefore, it is expected that all students complete the programming assignments before class, and late work will not be accepted (no exceptions except due to significant external circumstances, e.g., illness).
Additional Information	The course will use the Stepik platform at https://stepik.org/course/73755. We explain how to sign up in the first class.

Tentative Course Schedule
(Time, topic/emphasis & resources)

Week	Lectures	Topic
1	Mon 08/31/2020 Wed 09/02/2020	Introduction to the course, group discussions, and the interactive textbook.
2	Mon 09/07/2020 Wed 09/09/2020	Finding hidden messages in biological sequences We will start with "DNA replication" as algorithmic warmup and introduce the first methods to find frequent substrings as a way to solve the question of where DNA replication begins. The class will focus on the algorithmic aspects of solving the biological questions, discussion of complexity and different approaches to find patterns in strings (with and without mismatches).

3	Mon 09/14/2020 Wed 09/16/2020	Randomized algorithms for motif search in strings; or: Which DNA patterns play the role of molecular clock? We expand string matching algorithms to randomized algorithms, including Gibbs sampling, in order to identify genes involved in circadian rhythm (molecular clocks).
4	Mon 09/21/2020 Wed 09/23/2020	String reconstruction from small fragments, or: How do we assemble genomes? We investigate graph-based algorithms for reconstructing large strings from small fragments; de Brujin graphs; Eulerian paths and circles. We will also introduce Variation graphs as a recent research topic to represent and query potentially millions of genomes.
5	Mon 09/28/2020 Wed 09/30/2020	String reconstruction for special types of strings, or: How do we find and sequence antibiotics? Many antibiotics are somewhat special in that they rely on non-ribosomal mechanisms; we introduce the problem as extension of string reconstruction and introduce algorithms that can solve this new class of problem.
6	Mon 10/05/2020 Wed 10/07/2020	Dynamic programming for computationally hard problems, or: How do we compare sequences of genes/genomes in different species? Introduction to dynamic programming for the string alignment problem; focus on space and time complexity.
7	Mon 10/12/2020 Wed 10/14/2020	String rearrangement, or: How do we compare whole genomes? Comparing the genomes of different species and give us insights in places that are more "fragile" than others. We will study algorithms that can identify fragile regions by looking are large rearrangements of substrings.
8	Mon 10/19/2020 Wed 10/21/2020	Midterm
9	Mon 10/26/2020 Wed 10/28/2020	Building trees from string distances, or: which animal gave us COVID-19? We use distance between strings (biological sequences) to generate trees that can give us information about the relatedness between the genes (or species). We use this to identify which animal gave us COVID-19.
10	Mon 11/02/2020 Wed 11/04/2020	Clustering algorithms, or: Finding groups of related genes in yeast Clustering is a way to find groups of related entities based on their representations as features. We use this to investigate gene expression data in yeast.

11	Mon 11/09/2020 Wed 11/11/2020	Efficient combinatorial pattern matching, or: How do we find disease-causing mutations in human genomes? Introducing tries, suffix trees, suffix arrays, Burrows-Wheeler Transform, and extensions of these algorithms. We use them to query whole genomes for patterns that can cause diseases.
12	Mon 11/16/2020 Wed 11/18/2020	Hidden Markov Models, or: Why don't we already have a vaccine for COVID-19 or HIV? Hidden Markov Models and learning HMMs; we use these to find CG islands and features (biological functions) that are preserved even when many mutations are accumulated in a sequence over time.
13	Mon 11/23/2020 Wed 11/25/2020	continued
14	Mon 11/30/2020 Wed 12/02/2020	Spectral analysis, or: Was T. rex just a big chicken? We discuss statistical methods to analyze and identify peptides from spectral data.
15	Mon 12/07/2020 Wed 12/09/2020	Final exam
16	Mon 12/14/2020	

Note

The instructor reserves the right to make changes to this syllabus as necessary.