



## Item Navigation

# Syllabus

## Course Title

Algorithms for DNA Sequencing

## Course Instructor(s)

Ben Langmead and Jacob Pritt

## Course Description

We will learn the concepts, algorithms and data structures that are foundational to the analysis of DNA sequencing data. We will use what we learn, together with our programming skills, to implement these and similar ideas. We will work with real data: real genome sequences and real sequencing data.

To succeed in this course, it is helpful to have some undergraduate-level background in computer science, though we have worked hard to make the course accessible even to students with minimal or no computer science background. You definitely *do not* have to be a trained biologist. You don't have to be an expert on genomics and genomes. You don't have to know how a DNA sequencer works. You just have to be curious about these things, and to be open to learning a little about genomics and sequencing as you go.

You will need Python programming skills. All of the code that we'll show you in class, in the lecture notes and in the practical sessions, will be written in Python. We will sometimes ask you to modify or adapt our examples written in Python. So if you're not so confident in your Python skills, you might want to get some more practice before you enroll, or you might go ahead and enroll, but if you do you should set aside some extra time to learn and practice as you go along.

## Course Content

**Module 1: DNA sequencing, strings and matching:** Why study this? DNA sequencers and how they work. How DNA can be represented as a string. Using Python to parse and manipulate real genome sequences and real DNA sequencing data. Naive exact matching.

**Module 2: Preprocessing, indexing and approximate matching:** Improving on naive exact matching with Boyer-Moore. Preprocessing and indexing. Indexing: grouping and ordering.  $k$ -mers and  $k$ -mer indexes. Approximate matching and the pigeonhole principle.

**Module 3: Edit distance, assembly, overlaps:** Hamming and edit distance. Algorithms for computing edit distance. Dynamic programming. Global and local alignment. De novo assembly. Overlaps and overlap graphs.

**Module 4: Algorithms for assembly:** Shortest common superstring and the greedy version. How repetitive DNA makes assembly difficult. De Bruijn graphs and Eulerian walks. How real assemblers work. The future of assembly. Wrap up.

## Weekly quizzes

There are four weekly quizzes. You may begin submitting them as soon as the course opens. Quiz 1 is due at the end of the first week, Quiz 2 is due at the end of the second week, Quiz 3 is due at the end of the third week, and Quiz 4 is due at the end of the fourth week.