Description
This course will survey current areas where computer science approaches have been applied to genomics research. Chiefly, the course focuses on DNA sequencing data analysis, including sequence alignment, de novo assembly, error correction, and DNA data compression. Subject matter will be partially guided by student interests. Students will present papers orally.

Prerequisites
No specific course requirements, but all participants should have background in genomics or computer science.

Instructor
Ben Langmead, langmea@cs.jhu.edu, www.langmead-lab.org

Teaching Assistant
None

Meetings
Monday, 11:00–11:50 am, Malone 222

Textbooks
No required textbooks. All readings will be academic papers or other materials that all can access.

Attendance
There will be 13 class sessions (possibly 12 due to Professor's travel). Students auditing the course must attend all sessions, barring illness or religious holidays. JHU attendance policies apply.

Online Resources
See Piazza site (http://piazza.com/jhu/spring2015/en600780) for all materials.

Course Objectives
(1) Apply knowledge of data structures, algorithms and analysis of algorithms to problems in Computational Genomics
(2) Attain knowledge needed to read and interpret cutting-edge results in computational genomics
(3) Understand the strengths and limits of current genomics data analysis methods
(4) Prepare to lead new research projects in computational genomics

Course Topics
The specific topics we will cover will be decided in the initial week of class. The backgrounds and interests of the enrolled students will be taken into account.

Following are general areas we may decide to cover with our readings. Generally speaking, any topic that involves applications of ideas from computer science to the analysis of biological data is relevant.

- Methods for correcting sequencing errors
- Methods for compressing sequencing data
- Methods for assembling sequencing reads
- Methods for aligning sequencing reads to genomes
- Non-string representations of reference genomes (e.g. pan-genomes, graphical genomes)
- Min-hashing, minimizers, and locality-sensitive hashing for sequence analysis
- Text indexing for sequence analysis
- Streaming algorithms and sketch data structures for sequence analysis
- Use of computational methods by large consortium projects
- Scalable methods for analyzing large collections of sequencing data

Following is a sampling of papers that were assigned in this class or similar classes in the past:


**Ethics**

The strength of the university depends on academic and personal integrity. In this course, you must be honest and truthful, abiding by the *Computer Science Academic Integrity Policy*:

Cheating is wrong. Cheating hurts our community by undermining academic integrity, creating mistrust, and fostering unfair competition. The university will punish cheaters with failure on an assignment, failure in a course, permanent transcript notation, suspension, and/or expulsion. Offenses may be reported to medical, law or other professional or graduate schools when a cheater applies.

Violations can include cheating on exams, plagiarism, reuse of assignments without permission, improper use of the Internet and electronic devices, unauthorized collaboration, alteration of graded assignments, forgery and falsification, lying, facilitating academic dishonesty, and unfair competition. Ignorance of these rules is not an excuse.

Academic honesty is required in all work you submit to be graded. Except where the instructor specifies group work, you must solve all homework and programming assignments without the help of others. For example, you must not look at anyone else’s solutions (including program code) to your homework problems. However, you may discuss assignment specifications (not solutions) with others to be sure you understand what is required by the assignment.

If your instructor permits using fragments of source code from outside sources, such as your textbook or on-line resources, you must properly cite the source. Not citing it constitutes plagiarism. Similarly, your group projects must list everyone who participated.

Falsifying program output or results is prohibited.

Your instructor is free to override parts of this policy for particular assignments. To protect yourself: (1) Ask the instructor if you are not sure what is permissible. (2) Seek help from the instructor, TA or CAs, as you are always encouraged to do, rather than from other students. (3) Cite any questionable sources of help you may have received.

On every exam, you will sign the following pledge: "I agree to complete this exam without unauthorized assistance from any person, materials or device. [Signed and dated]". Your course instructors will let you know where to find copies of old exams, if they are available.

Report any violations you witness to the instructor.

You can find more information about university misconduct policies on the web at these sites:

- For undergraduates: [http://e-catalog.jhu.edu/undergrad-students/student-life-policies/](http://e-catalog.jhu.edu/undergrad-students/student-life-policies/)
- For graduate students: [http://e-catalog.jhu.edu/grad-students/graduate-specific-policies/](http://e-catalog.jhu.edu/grad-students/graduate-specific-policies/)
Students with Disabilities
Any student with a disability who may need accommodations in this class must obtain an accommodation letter from Student Disability Services, 385 Garland, (410) 516-4720, studentdisabilityservices@jhu.edu.

ABET Outcomes

- An ability to function effectively on teams to accomplish a common goal (d)
- An ability to communicate effectively with a range of audiences (f)
- An ability to use current techniques, skills, and tools necessary for computing practice (i)
- An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices (j)