02-251: Great Ideas in Computational Biology

Carl Kingsford and Phillip Compeau

Spring 2019

1. Course Information

1.1 Vital information

		Lecture	**Recitation**
Course	Time	TR 10:30-11:50 AM	F 12:30-1:20 PM
	Location	GHC 4303	GHC 4215
		Prof. Carl Kingsford	**Prof. Phillip Compeau**
	E-mail	carlk@cs.cmu.edu	pcompeau@cs.cmu.edu
Instructors	Website	kingsfordlab.cbd.cmu.edu	compeau.cbd.cmu.edu
	Office	GHC 7725	GHC 7403
	Office Hour	W 1:30-3:00 PM	W 10:30 AM-12:00 PM
		Hongyu Zheng	**Wendy Yang**
	E-mail	hongyuz1@andrew.cmu.edu	muyuy@andrew.cmu.edu
TAs	Office Hour	TBA	TBA
	Location	TBA	TBA

1.2 Course description

This 12-unit course provides an introduction to many of the great ideas that have formed the foundation for the transformation of the life sciences into a fully-fledged computational discipline. This gateway course is intended as a first exposure to computational biology for first-year undergraduates in the School of Computer Science, although it is open to other quantitatively and computationally capable students who are interested in exploring the field. By completing this course, students will encounter a handful of fundamental algorithmic approaches deriving straight from very widely cited primary literature, much of which has been published in recent years. The course also introduces basic concepts in statistics, mathematics, and machine learning necessary to understand these approaches. Many of the ideas central to modern computational biology have resulted in widely used software that is applied to analyze (often very large) biological datasets; an important feature of the course is that students will be exposed to this software in the context of compelling biological problems.

1.3 Course philosophy

Biology's revolution in the 19th Century was led by Darwin and Mendel, who respectively discovered that species are evolving as a function of natural selection of mutating traits, and that inheritance of inherited traits can be broken down into heritable units. Biology's revolution in the 20th Century was molecular. See, for instance, the discovery of how heredity is encoded in DNA, and the beautiful symphonies in which information encoded in DNA's four-character alphabet is translated into proteins that a cell uses to function, as well as the mechanisms controlling how this translation can be turned on and off (called "gene expression").

Today, biology is in the midst of a new revolution, one in which we can only address the most important unresolved biological questions using machines. How else can we examine 3 GB genome files from hundreds of thousands of individuals? How else can we analyze gene expression data for billions of samples of different human tissues to discover what makes cells sick and why we age? How else can we categorize millions of high-resolution medical images to improve on human diagnosis? How else can we run thousands of experiments in a laboratory in a single day and analyze the output?

Computational biology therefore is not an exotic offshoot of the life sciences. It is not a tool taken down from a high shelf occasionally in the lab. It is the *only* way to solve an ever growing collection of unanswered biological questions.

We must also assert that computational biology is an established computational discipline; its approaches are not merely borrowed from computer science. Rather, innovative computational approaches have been developed within computational biology, and many of them are so fundamental that they then became applied outside of biology. (You will see that the Smith-Waterman algorithm for "local" comparison of two strings is such an example.)

Finally, a great idea is rare. We have left out many "good" ideas in computational biology in constructing this course in favor of showing you the fundamental beauty of the field. The great ideas we present in many cases derive from research results that are at most 50 years old, and some are still hot from the fire.

1.4 Pre-requisites

The most common question we anticipate is, "How much biology do I need to know?" The reason we anticipate this question is that high school biology education is, in most places, boring and outdated. Although taking an introductory biology course concurrently cannot hurt, it is not required for this course, which is first and foremost a computational course.

Because of the course's heavily quantitative and computational nature, we suggest the following prerequisite courses. We also suggest that students consider taking 15-122 (Principles of Imperative Computation) concurrently to guarantee strength in programming.

- 02-201 (Programming for Scientists) or 15-112 (Fundamentals of Programming and Computer Science)
- 15-151 (Mathematical Foundations for Computer Science) or 21-127 (Concepts of Mathematics)

1.5 Course Details

Course Homepage `http://www.cs.cmu.edu/~02251`

Canvas Homepage. Canvas will be used for attendance and as a central repository for grades. You should be automatically enrolled at `https://canvas.cmu.edu/courses/8153`.

Submitting Assignments. We will use Gradescope for theory assignments. Please visit `https://gradescope.com` and use the following entry code to see our course: 94G82E. For automatically graded programming assignments, links will be provided throughout the course.

Discussion Forum. An online forum is provided on Piazza as an area for discussion and questions. The forum will be moderated by the course staff who will respond to questions, but students are encouraged to help each other via discussion. However, assignment specifics should not be discussed — any hints will be provided by the teaching staff. You can find the class on Piazza at `https://piazza.com/class/jpbntwh8tix1hu`.

Programming Expectations. Early in the course, we will provide test datasets; later, we will assume that you will write your own tests. Programming assignments in this class are based on the model of giving you a randomized dataset and asking you to return the result of running your algorithm on this dataset.

Accordingly, there is no official language for the course; you can solve programming assignments using the language of your choosing. We expect you to produce clean, readable, and well-documented code.

2. Curriculum

2.1 Tentative course topics

We hope to cover great ideas taken from the following topics (note: very much subject to change).

Genome assembly The genome of a cell is the sum total of its DNA. In humans, this DNA is made up of three billion nucleotides divided onto 23 chromosome pairs; each nucleotide is either adenine, cytosine, guanine, or thymine. A genome can therefore be seen as a three billion letter long string from a four-symbol alphabet. The nucleotides are submicroscopic, so how can we read a genome? Cutting-edge technologies can identify fragments of DNA called "reads"; by generating reads from a biological sample containing many fragmented copies of a genome, our goal is to *assemble* the fragments into a contiguous genome. We will see how graph theory is critical to solving this problem.

Sequence alignment Two similar genes may have diverged by mutations that either change, insert, or delete a nucleotide. This leads us to a scoring function to determine a "minimum cost" transformation of one gene into another by these mutations. We will learn how to apply dynamic programming to solve this problem and its many variants, before considering the case of multiple strings.

Read mapping Once we have assembled the genome of one organism (or a collection of organisms) from a species, we can use this genome as a "reference" when determining how another organism from the same species differs. We generate DNA sequencing reads from this organism, and "map" them against the reference genome. We will simplify this problem as an instance of matching a family of many patterns against a longer text, and see how computational biology produced the state of the art in this area after many subsequent improvements.

Hidden Markov models for sequence alignment A protein is a chain of amino acids generated (most commonly) from a collection of 20 amino acids. When we identify a new protein, we want to compare it to existing families in a protein database. We will borrow the concept of a Hidden Markov model from machine learning to identify a protein's best "alignment" against a family of proteins.

Metagenomics There are microbes living all around us in the environment and on our bodies. A typical human has more bacterial cells than human cells. Metagenomics is the study of how these communities of microbes function together and affect health and the environment. Modern DNA sequencing technologies allow us to sequence all the DNA present in an environmental sample. From this sequencing we try to discover what microbes are present, and how they are correlated with features of the environment that was sequenced.

Phylogenetics The problem of constructing a "Tree of Life" dates to Darwin, who had the insight that species would branch out as they diverged. The construction of evolutionary trees, or "phylogenetics", is still an accurate area of research. For example, if you take 02-261 (Quantitative Cell and Molecular Biology Laboratory), you will sample river water from Pittsburgh's three rivers and isolate DNA. Sequencing every occurrence of the same gene in your samples is an example of "metagenomics", in which we are interested in the DNA present within an environmental sample (another such environment constitutes the bacteria inhabiting your intestines). One of many questions we can ask of our river water sample, "What is the evolutionary tree constructed from the microbes within it?" We will learn a variety of phylogenetics algorithms and apply them to our sample.

Network biology Within a cell, molecules interact in order to carry out biological functions. One can view these interactions as forming a "social network" of the components of the cell. A number of analyses have been developed to use this social network to infer how the cell works. In particular, we will see how these kinds of analyses are used to predict what proteins do and also to understand how the cell has evolved to be robust to mutations.

RNA sequencing The sequencing of the first human genome in 2000 was hailed by the *New York Times* in the grandest possible terms with the headline "Genetic Code of Human Life Is Cracked by Scientists". The reason why this (biologically incorrect)[1] headline is so overstated is that a genome only provides us with one piece of information about the identity of a cell. For example, every cell in your body has essentially the same genome, and yet your cells are very different because the expression of genes varies. Only in the last ten years have we learned how to accurately and cheaply identify the rates at which genes are expressed by measuring the levels of their RNA transcripts in a single cell. This produces a host of questions – what are the expression profiles of cells in different human tissues? How do they differ between organisms? How does this differ across species sharing the same genes? How is gene expression implicated in disease? We will see how clustering and dimensionality-reduction approaches can be used to analyze the enormous amounts of RNA data produced from millions of cells.

Variant detection and population genetics If you swab your cheek and send a sample to a company like 23AndMe or Ancestry, how are they able to determine your genetic makeup by breaking it into ethnicities, or predict your predisposition for certain diseases or allergies?

[1]The genetic code defines how triplets of DNA nucleotides are eventually translated into amino acids to make protein.

We will explore "genotyping", in which we differentiate individuals by using a subset of the total genome. One of the most popular ways to genotype is to identify "single-nucleotide variants" that are present in a large number of individuals. There are millions of these variants, though; how can we decide which ones are worthy for disease and population studies? We will discuss statistical frameworks for detecting appropriate genome variants and then use genotyping data to group humans into populations.

Machine learning and automated science Much of computational biology relies on high-throughput experiments to generate lots of data. Even with robots and high-capacity devices, however, the space of possible measurements is too large to test everything. For example, suppose you want to test whether each pair of human proteins interact. Assuming each gene produces only 1 protein (a significant underestimate), you would have to test about 242,000,000 pairs of proteins, which is not possible. Instead, the idea of "active learning" can be used to use machine learning to help select which experiments to conduct. In this way, we automate some of the scientist's job, and are able to learn about a system more efficiently because (hopefully) we are only doing the experiments that tell us something new. We will see several applications of active learning, applied to drug discovery and other problems.

Biological image analysis Much as in other areas of biology, the scale at which cellular (and medical) images are being produced is now making human analysis of these images obsolete. How an we train a computer to "see" important features in the images and analyze them automatically on our behalf? Is it possible for a computer to beat a human biologist/doctor at some of this analysis? (Spoiler warning: yes.). "Deep learning" algorithms built on "neural networks" are designed to mimic the human brain, specifically the way in which neurons communicate. These algorithms have revolutionized many fields of study, and we will turn them back upon biology to analyze biological images.

Protein structure prediction and proteomics In the 1950s, it cost on average about ten million (inflation-adjusted) US dollars to bring a drug to market. In a phenomenon known as "Eroom's law" (Moore's law spelled backwards), the number of drugs produced per dollar of research has fallen exponentially since that time; today, the cost of a drug is on the order of a billion dollars, and a pharmaceutical company worth $50 billion may actually only produce a few medicines. There are many reasons for this phenomenon, such as that low-hanging fruit were discovered early, and that most drugs that work in mice will fail when given to human patients. Another issue is that when attempting to find a protein serving as a drug candidate, it would be ideal if we could predict the three-dimensional shape of the protein (and therefore hypothesize on its function) from the sequence of RNA nucleotides that translate into the protein. We will see that this is a much more complicated problem than it seems, a glimpse into the study of proteins, or proteomics.

Mini-lectures In fall 2018, Dr. Kingsford organized a Workshop on the Future of Algorithms in Biology at CMU, or FAB 2018 for short (`http://fab2018.cbd.cmu.edu`). Think of it as a "great ideas" conference on the frontiers of computational biology. Speakers were encouraged to deliver talks that would be accessible to a wide audience, and the resulting presentations were excellent. We plan to borrow the "mini-lecture" concept from 15-112 and provide a menu of FAB 2018 talks (and maybe other talks) for you to choose from. You will watch and summarize these talks in lieu of attending class on the two days before midterms; it will give you some extra time to study, too!

2.2 Tentative course schedule

As with course topics, the lecture-by-lecture below is tentative and subject to change.

COURSE SCHEDULE			
Date	**Lecture Topic**	**Date**	**Lecture Topic**
1/15	Genome assembly (Part 1)	1/17	Genome assembly (Part 2)
1/22	Sequence alignment (Part 1)	1/24	Sequence alignment (Part 2)
1/29	Read mapping (Part 1)	1/31	Read mapping (Part 2)
2/5	Hidden Markov models for sequence alignment	2/7	Advanced sequence alignment
2/12	Mini-lecture day 1	2/14	Midterm 1
2/19	Network biology (Part 1)	2/21	Network biology (Part 2)
2/26	RNA sequencing (Part 1)	2/28	RNA sequencing (Part 2)
3/5	Variant detection	3/7	Population genetics
	SPRING BREAK		
3/19	Metagenomics	3/21	Phylogenetics (Part 1)
3/26	Phylogenetics (Part 2)	3/28	Mini-lecture 2
4/2	Midterm 2	4/4	Machine learning and automated science (Part 1)
4/9	Machine learning and automated science (Part 2)		SPRING CARNIVAL
4/16	Cellular image analysis (Part 1)	4/18	Cellular image analysis (Part 2)
4/23	Protein structure prediction and proteomics (Part 1)	4/25	Protein structure prediction and proteomics (Part 2)
4/30	Project presentations	5/2	Project presentations

3. Recitations

Recitations will focus on a blend of the following:

- reinforcement and further discussion of key algorithms presented in the course;
- additional interesting computational biology tidbits on the current subject that did not fit into the main lecture;
- discussion questions building knowledge of critical concepts;

- new, related topics;
- application of popular biological software implementing ideas in the course and applied to real datasets.

Recitation attendance will not be graded, but attendance is strongly suggested.

4. Coursework

Coursework will consist of the following components. **No late assignments will be accepted.**

Homework assignments. (30% of grade) Homework assignments will comprise two parts.

1. Automatically graded programming assignments (20% of grade) will ask you to implement many of the algorithms forming the great ideas for the course. Programming assignments must be completed on your own (unless noted otherwise) and turned in to the autograder by a given deadline.
2. Theory questions (10% of grade) will be provided before each lecture to encourage you to review the material and arrive to class prepared.

Examinations. (40% of grade) The midterms and final exam will test your knowledge of the material from the class. The midterms will be held in class and the final will be held during the university's scheduled time. The midterm dates are:

- Midterm 1 (10% of grade): Thursday, February 14 (in class)
- Midterm 2 (10% of grade): Tuesday, April 2 (in class)
- Final (20% of grade): Time and location TBD (will be posted when set by university)

The midterms will not be cumulative: midterm 2 will cover material encountered after midterm 1. That having been said, later material in the class builds upon the earlier material, so it is important to know the earlier material.

The final will be comprehensive, i.e., it will cover all the material from the class.

Project. (20% of grade) We want this course to empower you to find your own great ideas in computational biology. Accordingly, you will complete a project analyzing a biological data set. We will provide more details about the project as the course progresses. The final week of the course will feature in-class presentations at the end of the course. You will be graded on this presentation as well as a write-up describing your work.

Attendance and participation (10% of grade) Attendance will be taken, and we will have occasional in-class exercises that serve to reinforce the concepts we have covered. These exercises will not be graded, but participation will be expected in order to receive a complete grade for that day.

You are allowed three "dropped" attendance grades without penalty. These can be used for any purpose.

5. Collaboration Policy and Academic Integrity

All class work should be done independently unless explicitly indicated on the assignment handout. You may *discuss* homework problems and programming assignments with classmates, but must write your solution by yourself. If you do discuss assignments with other classmates, you must supply their names at the top of your homework / source code. No excuses will be accepted for copying others' work (from the current or past semesters), and violations will be dealt with harshly. (Getting a bad grade is much preferable to cheating.) In addition to manual inspection, we use an automatic system for detecting programming assignments that are significantly similar.

The university's policy on academic integrity can be found at the following link: `http://www.cmu.edu/academic-integrity/`. In part, it reads, "Unauthorized assistance refers to the use of sources of support that have not been specifically authorized in this policy statement or by the course instructor(s) in the completion of academic work to be graded. Such sources of support may include but are not limited to advice or help provided by another individual, published or unpublished written sources, and electronic sources." You should be familiar with the policy in its entirety. **The default penalty for any academic integrity violation is failure of the course.**

In particular: use of a previous semesters answer keys or online solutions for graded work is absolutely forbidden. Any use of such material will be dealt with as an academic integrity violation.

6. Other policies

Classroom etiquette: To minimize disruptions and in consideration of your classmates, we ask that you please arrive on time and do not leave early. If you must do either, please do so quietly. **The use of phones or other electronic devices during class is forbidden and will result in a zero discussion grade for the day (counts as missed class).**

Excused absences: Students claiming an excused absence for an in-class exam must supply documentation (such as a doctor's note) justifying the absence. Absences for religious observances must be submitted by email to the instructor during the first two weeks of the semester. Note that job or internship interviews are not a justification for an excused absence.

Other: The following policies of 15-110 also apply to this class. This text is mostly quoted from the 15-110 website (with some modifications):

- **I must be out of town for university related event (e.g. member of a team). What should I do about my assignments?**

  If you have an official excuse we will make special arrangements for you to submit the assignment, please contact the instructors.

- **I am out of town attending a family/important event. How can I submit my assignments due for the week?**

  The programming assignment must be submitted online before the due date. The written assignment must uploaded to Gradescope before the due date.

- **I missed the in-class exam because I fell sick. What should I do?** You must immediately seek medical treatment and receive an official medical excuse. You must also contact the instructors prior to the exam or as soon as possible. If you can produce documentation we can make arrangements to give you a makeup test. Otherwise, we will be unable to make any exceptions.

- **I am failing the course. Is there any extra work I can do to get a passing grade?** Unfortunately, we cannot make exceptions. The best way to avoid this situation is to talk to one of the instructors as soon as possible to find out what you need to do. Do not wait until the last few weeks of classes to discuss your performance.

7. Accommodations for Students with Disabilities

If you have a disability and have an accommodations letter from the Disability Resources office, we encourage you to discuss your accommodations and needs with us as early in the semester as possible. We will work with you to ensure that accommodations are provided as appropriate. If you suspect that you may have a disability and would benefit from accommodations but are not yet registered with the Office of Disability Resources, we encourage you to contact them at `access@andrew.cmu.edu`.

8. Provost's Statement on Student Well-Being

Take care of yourself. Do your best to maintain a healthy lifestyle this semester by eating well, exercising, avoiding drugs and alcohol, getting enough sleep and taking some time to relax. This will help you achieve your goals and cope with stress.

All of us benefit from support during times of struggle. You are not alone. There are many helpful resources available on campus and an important part of the college experience is learning how to ask for help. Asking for support sooner rather than later is often helpful.

If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety or depression, we strongly encourage you to seek support. Counseling and Psychological Services (CaPS) is here to help: call 412-268-2922 and visit their website at `http://www.cmu.edu/counseling/`. Consider reaching out to a friend, faculty or family member you trust for help getting connected to the support that can help.

If you or someone you know is feeling suicidal or in danger of self-harm, call someone immediately, day or night:

CaPS: 412-268-2922

Re:solve Crisis Network: 888-796-8226

If the situation is life threatening, call the police:

On campus: CMU Police: 412-268-2323

Off campus: 911

If you have questions about this or your coursework, please let us know.