

02-604: Fundamentals of Bioinformatics

Phillip Compeau
Fall 2020

1. Course Information

1.1 Vital information

Course	Time Location	Thursday 9:20-11:20 AM Zoom	
Instructor	E-mail Website Office Hour Location	Phillip Compeau, Ph.D. pcompeau@cs.cmu.edu http://compeau.cbd.cmu.edu Wednesday 3-4 PM Zoom	
Teaching Assistants	E-mail Office Hour Location	Filipp Gusev fgusev@andrew.cmu.edu By appointment Zoom	Mustafa Guler mguler@andrew.cmu.edu By appointment Zoom

1.2 Course description

How do we find potentially harmful mutations in your genome? How can we reconstruct the evolutionary Tree of Life? How do we compare related genes from different species? These are just three of the major questions in modern biology that can only be answered using computational approaches. This 12-unit course will explore a variety of fundamental topics in computational biology. It will delve into computational ideas used in biology as well as let students apply existing resources that are used in practice every day by bioinformatics professionals. The course also offers an opportunity for students who possess an introductory programming background to become more experienced coders in a biological setting.

1.3 Pre-requisites

There are no formal pre-requisites, but the course assumes an introductory background in biology (such as that provided by 03-709), some algorithmic thinking, and a strong foundation in programming (such as that provided by 02-601).

1.4 Course homepage

Canvas will be used to measure participation, for submitting some assignments, and as a central repository for grades. You should be automatically enrolled at <https://canvas.cmu.edu/courses/16919>.

1.5 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, but students are encouraged to help each other via discussion. However, assignment specifics should not be discussed. You can find the class on Piazza at piazza.com/cmu/fall2020/02604.

1.6 Course textbook

Bioinformatics Algorithms: An Active Learning Approach, by Phillip Compeau & Pavel Pevzner, 3rd Edition (<https://bioinformaticsalgorithms.org>).

We will be using an interactive version of this textbook that contains all of the programming assignments for the course on an educational platform called Stepik (<https://stepik.org>) that will contain an online textbook for the material in the course. The textbook has programming assignments built-in that are language-neutral, as they 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. We suggest that you write your own test datasets to test your code before running it on the larger dataset given.

2. Course philosophy

This course was a finalist for a Teaching Innovation Award at Carnegie Mellon, and it was a subject of an invited education article by the instructor (<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006764>). It is also a bit different from many other courses, and so it is important to pause for a moment to explain the course philosophy.

The central motivation for how this course operates is based on the “2 Sigma Problem”, a paradigm of educational psychology introduced by Benjamin Bloom in 1984. This remarkable principle, which has been verified by student performance, states that students achieve a two standard deviation improvement in learning performance when they are tutored individually compared to a standard lecture in a 30-student classroom — let alone a 100-student classroom!) The reasoning behind the 2 Sigma Problem is that 30 students learn different material in different ways and at different paces, but the traditional lecture is completely blind to this conundrum. In Bloom’s words,

Teachers are frequently unaware of the fact that they are providing more favorable conditions of learning for some students than they are for other students.

How, then, should a course be run? The instructor of any 30-student class unfortunately does not have time to tutor every student in the course individually. Instead, we must strive for extreme class discussion and “mastery learning”, in which a subject is understood perfectly before the next topic is covered.

These principles are not enacted by most courses because most courses do not have the time or resources to enact them. This course is different because it is powered by an interactive textbook that was co-developed by the instructor over a span of several years. This text has become a bestseller in the field, having been adopted by over 140 instructors since 2015 and completed by thousands of students in the instructor’s online courses with Pavel Pevzner.

The course will therefore operate on a “flipped” model, in which students read and complete assessments in the interactive text (as well as watch optional lectures) outside of normal class time.

This allows for class time to be used not for transmissive lectures but rather for in-depth guided discussions with students working together in small groups (see below). We expect that every student will participate in every session of the class by either individually asking questions about the course materials or by individually answering the questions posed by the instructor or other students in the class.

It is important to point out that although the “active learning” environment fostered by a flipped classroom has been proven to be more effective than a lecture environment, students sometimes feel that they would be learning more in a lecture (see Deslaruriers et al. 2019, <https://www.pnas.org/content/116/39/19251>). The flipped classroom is challenging, and we discuss questions in class that are probably more advanced than we would be able to cover in a lecture. It is at these moments of challenge that the most learning takes place.

3. Course Schedule

The week-by-week schedule below is tentative and subject to change.

Date	Topic
9/3	Motif Finding in DNA sequences
9/10	Genome assembly
9/17	Dynamic programming and introductory sequence alignment
9/24	Global and local alignment and additional topics in sequence alignment
10/1	Hidden Markov Models (HMMs) and profile HMMs for advanced alignment
10/8	Genome rearrangements and fragile genomes
10/15	Evolutionary trees (part 1): additive phylogeny and UPGMA
10/22	Evolutionary trees (part 2): the neighbor-joining algorithm and small parsimony
10/29	Read mapping (part 1): tries, suffix arrays, and suffix trees
11/5	Read mapping (part 2): the Burrows-Wheeler transform
11/12	Lecture: antibiotics sequencing and computational proteomics
11/19	Lecture: bonus day! (Topics TBA)
T H A N K S G I V I N G	
12/3	Student project presentations
12/10	Student project presentations

4. Course Details

Course Meetings. The course will meet once per week for approximately 120 minutes. Each weekly meeting will consist of some or all of the following items:

- Resolving common questions from the week's reading.
- A mini-lecture on a topic of interest connected to the week's topic.
- Discussion questions probing understanding on the week's topic.
- Challenge questions that extend the knowledge obtained of the topic to a new area.
- Questions encouraging students to start thinking about the next week's topic.

The final three items will typically be completed in randomly assigned groups of 4-5 students with instructor and TA support. Research has shown that this is the ideal group size for problem-solving. We will change the groups a few times throughout the semester to ensure that everyone gets a chance to work with as many people as possible.

Quizzes. To ensure understanding, a weekly comprehension quiz will be provided along with each week's reading, due the night before class.

Writeups. On the night before class, you will submit a write-up detailing your experience with that week's reading.

The write-up should include the following components:

1. A report of every difficulty that you had in understanding the material, documenting the precise location where the misunderstanding occurred. How did you resolve the misunderstanding?
2. A precise formulation of every remaining question that you have regarding understanding the material. A poorly formulated question would be, "I don't understand how Algorithm X works; can you please explain it again?" A better formulated question would isolate the source of the misunderstanding as closely as possible, indicating the specific location in the text that caused the misunderstanding (e.g., "If I input the data in Figure Y into Algorithm X, I notice that in line Z, I obtain a different result to the output in Figure Y"). Questions should not be easily answered in the text.
3. Any remaining questions that you have about the material that you're curious about.

A couple of notes on writeups. First, the best students tend to ask the most questions, and overly short writeups (especially ones that say "I understand everything!") will not be accepted. Second, the purpose of the writeup is as much for you as anything; we encourage you to take notes while you read the material, and use the writeup for logging things you didn't understand or want further clarification on.

Software Challenges. Software challenges will give you the opportunity to learn the basics of commonly used bioinformatics resources (such as BLAST, MEGA, MEME, SPAdes, etc.) These challenges will reinforce the algorithms that we cover in the book and give you an opportunity to see them applied to real data in the software used every day by biologists around the world.

Interactive Text and Programming Assignments. The interactive text for the course is hosted on Stepik; to obtain access, visit Canvas.

You should complete the weekly reading, quiz, and write-up by 8 PM on the night before class (submitting the quiz and writeup via Canvas), making sure you understand the week's work conceptually. The week's programming assignment is due on the Sunday night in the week following the meeting of the class. You will submit your responses for programming assignments at Stepik.

Code submission. You should also submit each week's code to Canvas, which we will use to prevent plagiarism.

(Optional) lecture videos. If you enjoy learning from a lecture, a complete set of lecture videos supplementing the interactive text can be found at the textbook's YouTube channel: <http://youtube.com/user/bioinfallgorithms/playlists>.

Programming Expectations. You are expected to produce clean, readable, and well-documented code.

Programming Languages. 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. (As such, you can solve these assignments using the language of your choosing.). We will provide test datasets for some of the assignments at the beginning of the course, but we want you to start writing your own tests after that.

We are more than happy to answer your questions about the course material and regarding setting up your programs. That having been said, for the sake of fairness, we will not be providing programming support for individual programming languages.

5. Coursework and Grading

Coursework will consist of weekly quizzes, a participation grade, extensive programming assignments, a series of software challenges, a midterm exam, and a final.

Weekly writeup. (5% of your grade) Each week, you should complete a thorough writeup describing your understanding of the week's reading and detailing any questions that you have on the material.

Participation. (5% of your grade) You should attend and fully participate in the flipped class sessions. You are allowed one "dropped" participation grade.

Comprehension quizzes. (10% of your grade) Due on the night before the weekly class meeting.

Software challenges. (10% of your grade) Due Sunday nights.

Programming assignments. (30% of your grade) Weekly programming assignments will be taken from the interactive text used in the course.

You will have a grace period of one week during which incomplete exercises from the assignment may be completed for half credit.

Final exam. (20% of your grade) The final exam will be an open-book, open-note take-home exam. The final exam will be comprehensive (i.e., it will cover all the material from the course).

Final project (20% of your grade) The course will have a project that you will work on throughout the semester. See the course homepage for details.

6. Collaboration Policy and Academic Integrity

All class work should be done independently unless explicitly indicated on the assignment hand-out. 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 semester's 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.

7. Diversity Statement

It is our intent that students from all diverse backgrounds and perspectives be well served by this course, that students' learning needs be addressed both in and out of class, and that the diversity that students bring to this class be viewed as a resource, strength and benefit. It is our intent to present materials and activities that are respectful of diversity: national origin, gender, sexuality, disability, age, socioeconomic status, ethnicity, race, and culture. Your suggestions are encouraged and appreciated. Please let us know ways to improve the effectiveness of the course for you personally or for other students or student groups. In addition, if any required class meetings conflict with your religious events, please let us know in the first two weeks of the course so that we can make arrangements for you.

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 me know.