02-604: Fundamentals of Bioinformatics

Phillip Compeau
Spring 2017

1. Course Information

1.1 Contact information

Instructor: Phillip Compeau
E-mail: pcompeau@cs.cmu.edu
Website: http://compeau.cbd.cmu.edu
Office: Gates-Hillman Center (GHC) 7403

TA 1: Myles Mao
E-mail: mmao1@andrew.cmu.edu

TA 2: Tim Majarian
E-mail: tmajaira@andrew.cmu.edu

1.2 Office hours

Note: Because of the structure of the course (see below), all TA and instructor office hours are by appointment.

1.3 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 offers an opportunity for students who possess an introductory programming background to become more experienced coders in a biological setting.

1.4 Course textbook

Bioinformatics Algorithms: An Active Learning Approach, by Phillip Compeau & Pavel Pevzner, 2nd Edition (http://bioinformaticsalgorithms.com).

We will be using an interactive version of this textbook that contains all of the programming assignments for the course (see below for more details).

1.5 Course goals

This course has two primary goals:

1. Hack a large selection of algorithms that are fundamental to an understanding of modern computational biology.

2. Obtain a practical "toolkit" for computational biology by learning how these algorithms are applied in existing software resources.

Along the way, the course offers a setting for students to take the next steps to become much stronger programmers.

1.6 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. 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 been adopted by 49 institutions in the past two years and has been completed by thousands of students in the instructor's online courses.

The course will therefore operate on a "flipped" model, in which students watch lectures and complete assessments in the interactive text outside of normal class time. This allows for class time to be used not for transmissive lectures but rather for discussions between the professor and students 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.

1.7 Pre-requisites

There are no strict pre-requisites, but this course assumes an introductory background and interest in biology, some algorithmic culture (such as that provided by 02-613), and some background in programming (such as that provided by 02-601).

1.8 Course Details

Course Meetings. Because the flipped classroom model frees up so much time, we will not be holding traditional lectures. Instead, we will have weekly 90-minute meetings in which students will be divided across four small groups (held on Tuesdays and Wednesdays). Each weekly meeting will consist of resolving student questions from the reading (provided via Blackboard) in addition to discussion ensuring that students have complete understanding of the material.

The times of discussion meetings are found below.

- Tuesdays 12:00 PM - 1:30 PM (GHC 7717)
- Tuesdays 1:30 PM - 3:00 PM (GHC 7717)
- Wednesdays 3:00 PM - 4:30 PM (Scaife Hall 214)
- Wednesdays 4:45 PM - 6:15 PM (GHC 7717)

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

Writeups On the Monday night before the weekly class discussion, you will be required to submit a short write-up detailing your experience with that week's reading.

The write-up should include the following components:

1. A number between 0 and 10 indicating a self-assessment of your overall understanding of the material, where 10 indicates complete understanding.
2. 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?
3. The third component depends on your understanding score:
   (a) If the score is 10, submit two nontrivial candidate quiz questions taken from the week's material (by "nontrivial", I mean no true-false, "define this term", etc.) This exercise is to help you further process the material (and I may use one of your questions in future quizzes!)
   (b) If the score is 9 or lower, precisely formulate the remaining questions that you have regarding the text, as we will use your submission as motivation for the question and answer discussion. For example, an imprecisely 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 book.

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 the Course Resources page at Piazza.

You should carry out the weekly reading, quiz, and any remaining questions that you have on the night before your discussion group, making sure you understand the week's work conceptually. The week's programming assignment is due on Sunday night. You will submit your responses for programming assignments at Stepik. You should also submit each week's code to Blackboard, which we will use to prevent plagiarism.

You will submit software challenges and weekly write-ups via Blackboard.

(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/bioinfalgorithms/playlists`.

Discussion Forum. An online forum is provided on Piazza (`https://piazza.com/cmu/spring2017/02604/home`) as an area for discussion and questions. The forum will be moderated by the course staff who will respond to questions, but you are encouraged to help each other via discussion. However, assignment specifics should not be discussed — any hints will be provided by the teaching staff.

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

Programming Languages. The 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 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.

2. Coursework

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

Comprehension quizzes. (10% of your grade) To be completed on Monday nights.

Weekly writeup and participation (10% of your grade) Each week, you are required to complete a writeup clarifying your understanding of the week's reading in addition to participating in your assigned discussion group. To receive full credit for the week's participation grade, you must complete both.

You are allowed one "dropped" participation grade that can be used for any purpose.

Programming assignments. (40% of your grade) Weekly programming assignments will be taken from the interactive text used in the course. They are language-independent.

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

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

Midterm. (10% of your grade) The midterm will test your knowledge of the material covered in roughly the first half the class; it will be held in class on **Wednesday, March 8**.

Final exam (20% of your grade) The final exam will be held in location and time TBA. The final will cover all the material from the course.

3. Course Schedule

Below is a **tentative schedule** of the course, indicating the central biological questions that we will cover. Next to each biological question is the main computational idea that we will use to answer this question. Software challenges will be incorporated as parts of relevant homework assignments.

- Week 1: Where in the Genome Does DNA Replication Begin? (*Algorithmic Warmup*)
  - In-class discussion: n/a
  - Homework due: Sunday, January 22
- Week 2: Which DNA Patterns Play the Role of Molecular Clocks? (*Randomized Algorithms*)
  - In-class discussion: January 24-25
  - Homework due: Sunday, January 29
- Week 3: How Do We Assemble Genomes? (*Graph Algorithms*)
  - In-class discussion: January 31-February 1
  - Homework due: Sunday, February 5
- Week 4: How Do We Sequence Antibiotics? (*Greedy Algorithms*)
  - In-class discussion: February 7-8
  - Homework due: Sunday, February 12
- Weeks 5-6: How Do We Compare Biological Sequences? (*Dynamic Programming*)
  - In-class discussion 1: February 14-15
  - Homework 1 due: Sunday, February 19
  - In-class discussion 2: February 21-22
  - Homework 2 due: Sunday, February 26
- Week 7: Are There Fragile Regions in the Human Genome? (*Combinatorial Algorithms*)
  - In-class discussion: February 28 - March 1
  - Homework due: Sunday, March 5
- **Midterm Exam: Wednesday, March 8**
- Week 8: Was *T. rex* Just a Big Chicken? (*Computational Proteomics*)
  - In-class discussion: n/a (because of midterm)
  - Homework due: Sunday, March 19

SPRING BREAK!

- Weeks 9-10: Which Animal Gave Us SARS? (*Evolutionary Tree Construction*)
  - In-class discussion 1: March 21-22
  - Homework 1 due: Sunday, March 26
  - In-class discussion 2: March 28-29
  - Homework 2 due: Sunday, April 2
- Week 11: How Did Yeast Become a Wine-Maker? (*Clustering Algorithms*)
  - In-class discussion: April 4-5
  - Homework due: Sunday, April 9
- Weeks 12-13: How Do We Locate Disease-Causing Mutations? (*Combinatorial Pattern Matching*)
  - In-class discussion 1: April 11-12
  - Homework 1 due: Sunday, April 16
  - In-class discussion 2: April 18-19
  - Homework 2 due: Sunday, April 23
- Weeks 14-15: Why Have Biologists Still Not Developed an HIV Vaccine? (*Hidden Markov Models*)
  - In-class discussion 1: April 25-26
  - Homework 1 due: Sunday, April 30
  - In-class discussion 2: May 2-3
  - Homework 2 due: Sunday, May 7

FINAL EXAM TBA

---

4. Collaboration Policy

You may discuss programming assignments with classmates, and I encourage you to do so. **However, you must not share or show or see the code of your classmates.** You must write your own code entirely. You can post general coding questions (with pseudocode snippets) on the discussion board.

You must write all programming assignments on your own and cannot share code with other students or use code obtained from other students. In addition to manual inspection, we use an automatic system for detecting programming assignments that are significantly similar.

5. Other policies

Classroom etiquette: To minimize disruptions and in consideration of your classmates, I 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 during class is forbidden and will result in a zero discussion grade for the day.

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.

Academic honesty: All class work should be done independently unless explicitly indicated otherwise. 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.)

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.

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

6. 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.