

University of Montana

ScholarWorks at University of Montana

University of Montana Course Syllabi

Open Educational Resources (OER)

Fall 9-1-2021

CSCI 558.01: Introduction to Bioinformatics

Travis J. Wheeler

University of Montana, Missoula, travis.wheeler@umontana.edu

Robert Smith

University of Montana, Missoula, robert.smith@umontana.edu

Follow this and additional works at: <https://scholarworks.umt.edu/syllabi>

Let us know how access to this document benefits you.

Recommended Citation

Wheeler, Travis J. and Smith, Robert, "CSCI 558.01: Introduction to Bioinformatics" (2021). *University of Montana Course Syllabi*. 12179.

<https://scholarworks.umt.edu/syllabi/12179>

This Syllabus is brought to you for free and open access by the Open Educational Resources (OER) at ScholarWorks at University of Montana. It has been accepted for inclusion in University of Montana Course Syllabi by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.

CSCI 451/558

Bioinformatics / Computational Biology

Instructors:

Dr. Rob Smith
robert.smith@mso.umt.edu
Please email for appointment.

SS 413
Office hours: MF 3:30-4pm, W 2-3pm.

TA

N/A

Class

Tu/Th SS 362 12:30-1:50PM

Learning outcomes:

This course introduces students to the field of bioinformatics. Students will practice applied problem-solving techniques for recurring classes of coding challenges in bioinformatics, with specific examples pulled from the bioinformatics subdomain of computational mass spectrometry. Class discussions will include an overview of career paths in the field, with practical and honest discussion of challenges in the way bioinformatics research is conducted today.

During this class, you will:

- Gain an appreciation for the need for start-to-finish solutions in bioinformatics (e.g., avoiding the “reducing the world to an algorithm” pitfall).
 - Become familiar with the challenges of taking data from raw formats to actionable outcomes:
 - Develop techniques for accessing and extracting information from large structured datasets.
 - Developing custom data structures for storing and accessing raw data.
 - Develop custom visualizations of data.
 - Finding and integrating existing algorithms for bioinformatics data.
 - Building custom algorithms for bioinformatics data processing.

An important note on programming experience:

The university catalog entry for this course assumes that you have sufficient coding experience that you can perform sophisticated coding tasks without the need for additional learning. If you are not at this level, but would like to use this course as an opportunity to self-study, you are welcome. Please note, however, that I will not be providing assistance with debugging code. At this stage, you are expected to be able to handle that on your own. You are highly encouraged

to work together on all programming projects (see below), so feel free to engage with a classmate or hire a tutor if you need programming help.

You are allowed to use any programming language(s) of your choice for this course, so long as they are capable of fulfilling the requirements of the project.

Tentative schedule:

| WEEK | Lecture | Suggested Project Progress |
|-----------------|---|-----------------------------------|
| Aug 30 – Sep 3 | What is bioinformatics? | |
| Sep 6 – Sep 10 | A framework for bioinformatics research: The data / metrics / existing methods scaffold | |
| Sep 13 – Sep 17 | A framework for bioinformatics research: The data / metrics / existing methods scaffold | Project 1 |
| Sep 20 – Sep 24 | Challenges in current approaches to bioinformatics research | Project 2 |
| Sep 27 – Oct 1 | Computational mass spectrometry: data and problems. | Project 3 |
| Oct 4 – Oct 8 | The Smith Lab research arc | Project 4 |
| Oct 11 – Oct 15 | Computational mass spectrometry: peak picking | Project 4 |
| Oct 18 – Oct 22 | Computational mass spectrometry: mass calculation | Project 4 |
| Oct 25 – Oct 29 | Computational mass spectrometry: molecular formula calculation | Project 5 |
| Nov 1 – Nov 5 | Innovation 1 | Project 5 |
| Nov 8 – Nov 12 | Innovation 2 | Project 5 |
| Nov 15 – Nov 19 | Computational mass spectrometry: proteins and peptides | Project 5/6 |

| | | |
|-----------------|---|-------------|
| Nov 22 – Nov 26 | Computational mass spectrometry: peptide identification from fragmentation mass spectra | Project 5/6 |
| Nov 29 – Dec 3 | Catch up | Project 5/6 |
| Dec 6 – Dec 10 | Project presentation | |
| | | |

Grades:

Lecture attendance (50%)

Projects: Each project is equally weighted for a total of 40% of your grade.

Bioinformatics mini research project: 10%.

Text Book:

None. If you aren't a strong coder, I would suggest: <https://www.codecademy.com/learn/learn-ruby> ASAP.

Cheating:

For the purposes of this course, cheating is defined as passing off the work of others as your own. Any form of collusion or dishonesty, as interpreted by instructors, will be prosecuted to the full extent allowable by University standards and may result in an automatic failing grade in the course. Group work is allowed up to two people per group for the projects, but not the research project.

Incompletes and Late Drops:

The university empowers instructors with discretion to approve incompletes or late drops (dropping the course after 45 days). I will not approve either as a means of avoiding a low grade or as a means of protesting course policies. Valid reasons include family emergencies, work complications, or registration issues, but I reserve the right of approval on a case-by-case basis.

Disabilities:

Eligible students with disabilities will receive appropriate accommodations in this course when requested in a timely way. Please contact us after class or in one of our offices. Please be prepared to provide a letter from your DSS Coordinator. For more information, visit the Disability Services website at <http://life.umt.edu/dss/>.

NOTE ON REQUESTS FOR REMOTE DELIVERY: This class will be offered face to face only. Dual modality instruction (face to face and remote) exceeds the 'reasonable accommodations'

specified by the Americans with Disabilities Act because working with the technology to accommodate remote learning during class seriously degrades the experience of face to face students. As a result, no accommodations will be made for remote learners. If a student has a documented need for a quiet examination space, they must use the facilities provided by DSS. No alternatives for test taking will be permitted.

Changes to Syllabi:

The instructors reserve the right to modify syllabi and assignments as needed based on faculty, student, and/or environmental circumstances. If changes are made to the syllabus, amended copies will be dated and made available to the class.

Moodle:

We will only be using moodle to upload this syllabus and the files you need to do your projects. Grades will not be on moodle. You can keep track of your grade using the formula given in this syllabus.

Grading Scale:

A 93-100; A- 90-92; B+ 87-89; B 83-86; B- 80-82; C+ 77-79; C 73-76; C- 70-72; D+ 67-69; D 63-66; D- 60-62; F < 60.

Bioinformatics mini research project

You must complete this assignment individually (no group work). Choose one of the following, and submit via email AS A .PDF with "bioinformatics mini research project" as the subject line. Option 1: Choose a bioinformatics problem and write a mini literature review. Your review should consist of an introduction, one paragraph for each of three published attempts at solving the problem, and a conclusion. Answer the following questions in the essay: What is the problem? Why is it a problem? What have others done to solve it? Your paragraphs on existing methods should clearly describe to a non-expert scientist (me) the mechanism, strengths, and weaknesses of the method. NOTE: If you are currently conducting research in bioinformatics for another class or your degree, you are allowed to use it for this assignment.

Option 2: Choose a bioinformatics problem and write an algorithm that solves it. Your solution must be original, and need to demonstrate why it is effective. NOTE: In the real world, you would need to quantitatively compare your algorithm with existing algorithms, but for the sake of making this manageable, you simply need to show it works, not that it is better than other approaches. Submit a report as a pdf consisting of a description of your approach and evidence that it works. Attach a zip file of your sourcecode or a link to a public repository, if you have published it online. NOTE: If you are currently conducting research in bioinformatics for another class or your degree, you are allowed to use it for this assignment.

Project Instructions

1. *Students are allowed to work in groups of up to two people. Both students should contribute equally to the work and to the presentation.*
2. *All projects will be graded in an in-class presentation as indicated on the calendar. You will receive 100 points for meeting the criteria of each project, 0 for failing to meet them, and extra points if you exceeded the standard, in the instructor's opinion.*

A note on data: Finding and converting data is a non-trivial part of bioinformatics work. You are expected to find your own mass spectrometry data for these projects. There are repositories like PRIDE that are available. You will likely need to convert from proprietary binary data formats, such as .raw, to open formats, such as .mzML using programs like msconvert (free). .mzML is a text-based file format, and so can be directly manipulated / inspected. It is highly recommended you work through this in partner groups to make it easier.

Project 1

Extract chromatograms from mzML. Show performance on a representative set of ten chromatograms. (Note: you are free to find your own mzML data, or you can use

Project 2

Decode compressed chromatograms from mzML. For this, you will have to find mzML files that contain compressed chromatograms and figure out how to write code to decompress them.

Project 3

Write code to plot your chromatogram. For this, you will have to find some suitable programming language that can create plots. If you are a grad student, you should save yourself time by ensuring the framework you use can handle interactive plots for Project 6.

Project 4

Write an algorithm to automatically choose peaks. Use your plot code to show your algorithm's performance on a representative set of input data.

Project 5

Create a user interface to wrap opening and processing an mzML file with chromatograms, indicate errors and crash gracefully, and display your chromatograms. You must support pan and zoom.

Project 6 (Graduate students only)

Create user controls and the necessary backend code to modify automatic peaks according to the user's preference. Strive to achieve the minimal number of clicks for the desired result.