Learning Objectives

The core material in this course focuses on the algorithmic side of computational biology: modeling biological problems computationally, formulating algorithms to solve our problems, and considering the efficiency and effectiveness of these algorithms. To balance this theory with some practice, we will give you the chance to explore a practical challenge of your own choosing by running a computational analysis on a real biological dataset and then interpreting the results.

There are two main possible directions for the project. First, you may implement an algorithm solving a biological problem and then use this algorithm to analyze a biological dataset. Second, you may analyze your dataset only using existing software. In either case, the primary expectation for the project will be based on the analysis that you carry out.

For example, if you were to write your own genome assembler for real biological data (a very challenging task to undertake), then it might suffice to apply your code to a bacterial read dataset and compare your resulting assembly and its statistics to those of an existing program. But if you were only to use existing software to assemble a genome, then we would expect you to take the project into additional directions to have a complete analysis. Also, note that while we cover lots of topics taken from genomics in the course, you are not necessarily required to complete a genomics project; as long as your topic features some computational problem that is solved to help us understand or use biology, it will work.

In addition to allowing you to put what you have learned into practice and explore your own interests in computational biology, this project will also allow you to practice working as a team for a semester-long project. Communication, organization, and interpersonal skills are vital to workplace collaboration, and as you will see in this class, almost all progress in computational biology comes from collaborative work.

Project Components

We expect the following deliverables as part of a successful project. See "Important Dates" for due dates. All deliverables can be submitted on Canvas.

Deliverable 1: Project Proposal

You should write a project proposal of at most one page. This proposal should:

- clearly state the problem, question, hypothesis, or goal your project will aim to solve, answer, or achieve;
- explain why this goal is interesting;
- discuss briefly your approach to achieving this goal and why you think this is feasible;
- if any are needed, identify the data or resources you will use for the project; and
- estimate the amount of time it will take you to complete the project.

It is reasonable if some or all of these change during the course of the project, but the proposal should make an attempt to plan out the project. This will help us help you shape the project to be successful.

Deliverable 2: In-Person Check-in #1

We will require a short (approximately 10-minute) in-person check-in to briefly describe project progress and ask questions related to progress.

Deliverable 3: Project Progress Report

By the progress report due date, you should provide a 1-page description of the progress you have made so far on your project, describing what you have done, what you plan to do in the remaining time, and identifying any problems you are encountering.

Deliverable 4: In-Person Check-in #2

We require a second short in-person check-in to briefly describe progress, explain what steps have been taken to resolve the problems in the written progress report, and ask questions related to finishing the project.

Deliverable 5: Project Presentation

You should deliver a short presentation to your peers in one of the last two weeks of the course, explaining your work and demonstrating your results. The format and length of the presentation will be decided mid semester.

Deliverable 6: Final Writeup

You should write a paper (minimum of 5 pages) with a summary of the project. This writeup should start from first principles, explaining the technical background of the biological problem, the computational problems formulated from this biological problem, and the algorithm(s) needed to solve the problem. It should then include a discussion of the results of applying the computational approach to a real dataset. The writeup should be clearly written, self-contained, and contain citations to relevant previous work. Any algorithms that you implement or software that you run should be fully explained on a high level (i.e., do not paste your code into the writeup).

For students enrolled in CSCI 558: additional individual component

In order to earn 500-level credit for this course, we will ask graduate students to extend the project in some way and submit an additional 1-2-page writeup describing the results. This could be by applying the technique to an additional dataset, testing another tool, etc. 558 students will also submit a proposal for their project extension when the project progress report is due.

Grading

Overall, your project will be graded according to the following percentages:

- Final writeup (50%). Your final writeup will be graded based on both the quality of your scientific work and the written quality of the writeup.
- Presentation (30%). Your presentation will also be judged based on scientific merit and your ability to present this scientific work intelligently and clearly to an audience of your peers who are unfamiliar with this work.
- Other deliverables (20%). To ensure you are completing the project in a timely fashion, we expect your group to complete all other deliverables. Each of the five other deliverables will be worth 4% of your total score.

Note that components are graded on both communication and scientific quality; this is by design and is to help you appreciate that communicating one's findings is as important a skill as conducting good scientific research.

For 558 students, your additional component will be graded as 75% final writeup, 25% progress report.

Important Dates

- Sunday, September 29 at 9pm: project proposal due
- Week of October 7th: in-person check-in #1
- Sunday, November 3rd at 9pm: written progress report due (and grad extension proposal due)
- Week of November 18: in-person check-in #2
- November 26 and December 3: project presentations
- December 11 at 9pm: project writeup due

Project Guidelines

Some guidelines for each component of the project to keep in mind when completing the project are as follows. This is not a rubric (i.e., we will not grade your project only according to these guidelines), but things that will help you ensure that you are on track.

Writeup: Written Quality

- Is there a clearly written introduction that explains the scientific problem for a lay audience?
- Are all figures clearly explained via captions and referenced appropriately from the main text?
- Is the writeup structured logically with a clear flow from the beginning of the article to the end?
- Does the writing generally follow English rules of grammar and syntax?

Writeup: Scientific Quality

- Does the writeup have the requisite length?
- Are any computational problems addressed in the project very clearly formulated?
- Are the key algorithms used explained on a high level for a wide audience without resorting to copying code into the document?
- Is there a thoughtful results section that is interpreted in the context of the scientific problem introduced and whose results are explained without resorting to appealing to a figure or dataset?

Presentation

- Does the presentation end on time?
- Does the presentation provide an understanding of the background to the question being addressed and its significance?
- Does the presentation clearly describe the key results of the project including conclusions and outcomes?
- Does the presentation follow a clear and logical sequence?
- Do the speakers convey enthusiasm for their project?
- Do the speakers capture and maintain their audience's attention?
- Do the speakers avoid scientific jargon, explain terminology and provide adequate background information to illustrate points?
- Do the speakers spend adequate time on each element of their presentation, or did they elaborate for too long on one aspect or was the presentation rushed?
- Do the slides enhance the presentation? Are they clear, legible, and concise?

Project ideas

The project is deliberately open-ended; you may choose to complete a project on any aspect of computational biology that you find interesting. However, we are providing some example topics below. You may be interested in one of these, or you may like to pick your own. It is acceptable if more than one group completes the same project.

Please note that some of the projects below may be too large for a single semester final project. You may just use the idea as a jumping-off point and narrow the scope of your project.

Furthermore, it is perfectly reasonable to begin one's project by examining existing research. In fact, in some cases, replication of an existing paper may be challenging enough to constitute a project.

- Compare the quality of the output of a family of different software programs for genome assembly on different types of organisms and read sets.
- Build a protein folding simulation that predicts the structure of a protein from its sequence of amino acids, perhaps including motif information from DNA as well.

- Use genotyping data to identify the population structure of a species (e.g., humans) and identify admixture in individuals.
- Classify a large family of cellular or medical images using deep learning with Tensorflow or PyTorch; in the case of medical images, extend this approach to provide an automated diagnosis of a patient based on their image(s).
- Build your own genome assembler that incorporates data from long reads or from "H-C" data finding chromatin contacts at different points of the genome.
- Replicate a metagenome assembler that can assemble multiple genomes from a single sample containing many species.
- Write an algorithm that will design primers for testing the presence of an arbitrary virus within an individual.
- Apply RNA sequencing to differentiate cells taken from the same tissue in different organisms, or from different tissues in the same organism.
- Build and analyze a gene co-expression graph from RNA-sequencing data.
- Build a transcription factor network connecting transcription factors to the genes they regular, and infer what biological conclusions can be drawn from the properties of the resulting graph.
- Apply game theory to understand evolutionary dynamics.
- Construct a family of evolutionary trees (perhaps using multiple tree construction algorithms) for a variety of multiple alignments on the same collection of taxa to obtain a collection of "gene trees". Then, design an approach that reconciles these differing trees into a single "species tree" for the collection.
- Extend the evolutionary tree model to handle recombination/horizontal gene transfer (e.g., in influenza viruses).
- Find patterns across human microbiome data in five different tissues for 300 different humans as part of the Human Microbiome Project. Or, identify which bacteria (or lack thereof) may be implicated in certain diseases.
- Analyze The Cancer Genome Atlas expression data to find which expression patterns are associated with differing cancer types.
- Gather patient data for the spread of a virus and compare it against theoretical models.

More Resources

If you come across any resources that might be helpful for other completing the project, please let me know and I will add them to this list.

Writing and presenting

UM has a writing center where you can get help with the written and presentation components of the project if needed.

Bioinformatics software

If you choose a project on a topic that we have already covered in the course, you may find the following list of software resources useful.

- De novo genome assemblers: https://en.wikipedia.org/wiki/De_novo_sequence_assemblers
- Sequence alignment software (pairwise/multiple alignments, profile HMMs, read mapping, metagenomics aligners, etc.): https://en.wikipedia.org/wiki/List_of_sequence_alignment_software
- Gene prediction software: https://en.wikipedia.org/wiki/List_of_gene_prediction_software
- Phylogenetics software: https://en.wikipedia.org/wiki/List_of_phylogenetics_software
- RNA Sequencing Software: https://en.wikipedia.org/wiki/List_of_RNA-Seq_bioinformatics_tools

Bioinformatics research

You could check out the following venues for recently published research in bioinformatics. Papers should be available through conference websites or on preprint servers such as bioRxiv or arXiv.

- RECOMB
- Workshop on Algorithms in Bioinformatics

Potential Pitfalls

Before beginning this project, it is helpful to be aware of some possible pitfalls that may make the project much more difficult than it needs to be.

With respect to group work in general, it is very helpful in any team to ensure that you meet frequently; I would suggest meeting at a minimum of once a week on a set schedule, with a clear set of aims each week. It is also helpful to have clearly defined roles. I would suggest the following three roles as a starting point with respect to meetings: meeting organizer, meeting recorder, and deadline enforcer. You may wish to define your own roles along the way, as well as rotate these roles. Feel free to let me know if I can provide any input!

Finally, project groups tend to underestimate the amount of time required to complete a successful project; this is why we have three deliverables along the way to ensure that you are on track. With respect to this particular project, it is critical to make sure that any source of data that you want to use exists. Authors of papers are often not obligated to publish the data that they used to reach a conclusion, and a few repositories of “public” data require too long to obtain approvals to obtain (e.g., full human genomes). Fortunately, we live in an era of very plentiful open data, but you should make sure that your particular project has a publicly available dataset. If you are using existing software, then it is vital that you are able to download and run this software. This sounds obvious, but not all academic software is made alike, and a great deal of this software is very difficult — if not downright impossible — to install and use. Once you identify a resource that you would like to use, please make sure that you are able to install it and run it on an example dataset.

Finally, please do not think that you need to do something entirely novel for this project. A fantastic starting point for a project is often to take what someone else has done, grab their data, and see if you can replicate it. This process may bring up enough issues to count as a complete course project! Or if the process goes smoothly, you can extend what they did.