6.802 6.874 20.390 20.490 HST.506
Final Project
Spring 2019

Important Dates:

Request to complete 6.874 with a team project: April 2th, 11:59PM
Proposals due: April 11th, 11:59PM
Proposal discussions: Week of April 13th – April 20th (there will be a web sign-up for times)
Project report due: May 9th, 11:59PM
Certain projects will be asked to present to the class May 14th and 16th during normal lecture times.

Process

The final project in this subject will consist of the following components: proposal and proposal discussion, research project, and project report.

A project team may consist up to three registered students who have completed all of the assignments in the subject before the final project, except that 6.874 students will complete a project on their own. 6.874 students may request permission to work as a team by emailing 6.874staff@mit.edu by April 2th by providing a one paragraph description of their project, their team members, and a reason that their project requires team effort.

If you work in a team, you must:

- Make clear before you start what the division of labor will be.
- Make clear in the written report what the division of labor actually was (it's fine if it deviates from the proposal, but it must be specific and accurate).
- Be sure that all participants understand all of the work.
- Projects done by n people will be expected to have n times as much technical depth and content as those done by a single person. For joint projects, the written work may be done jointly.
- Be sure to cite all papers and web sites consulted during the course of your project, as well as to acknowledge others who helped you.

The proposal (one per team) will be a written document, 1–2 pages long, outlining the work to be done. It should include a plan with at least 4 intermediate milestones, and indicate your internal deadlines for each of those steps. If there are multiple participants, the division of responsibility should be made clear. In addition, please include your assessment of what the "risks" to the project are: that is, what things do you think might turn out to be more difficult than planned, and what thoughts do you have about how to mitigate the risks?

If you are going to do an empirical study, be sure that you think about what method to use as a "baseline". It might be running a simple off-the-shelf algorithm or comparing to what happens if you predict the most common class. Remember that almost anything will turn out to be harder and more time-consuming than you expect. Try to arrange your project so your intermediate milestones can serve as alternative finishing points, in case you don't get to the end. It will be much better to turn in a polished version of a small-scale project than to find yourself at the end of the term with a three-quarters implemented system of great depth and scope.

A proposal interview (one per team) of about 10 minutes will be scheduled for all teams. This is your chance to get feedback on your proposal, with some ideas about how to structure your experiments, etc.

The project report (one per team) will be a written document of about 4n pages in double column conference format, where n is the number of people in your group, including whatever graphs and tables that are necessary to make your point. The report is the means by which you communicate the process and results of your project, so it should be clear, coherent, and well written. Do not dump out large quantities of data or code or uninterpreted charts. Emulate the expositional style of a technical conference paper. But, you do not need a detailed related work section. But, be sure to cite and very quickly explain any technical work you referenced in formulating and carrying out your project. Previous work should be referenced in your original proposal, so you do not need to duplicate that in your final report.

The main goals of your report are to make clear what your findings are, why you think they came out the way they did, and why that might be important and to be precise enough to allow someone to replicate your experiments (or verify your proofs).

Projects

You have 4 weeks to do this project, and we expect it to take about 6 person-hours per week; so that's 25 person-hours for a single-person project and 50 person-hours for a two-person project, etc. You'll have to make a plan and stick to it, to avoid getting behind and doing a bad rush job at the end.

Here are some ideas for types of projects:

Comparing different methods for a problem This is probably the best option if you don't have a concrete idea of something different to do. This would involve performing a careful comparison of various extant methods and models for a problem in computational biology or healthcare. You would need to characterize and understand the data characteristics/situations for which the different techniques work well (or don't). These two papers are a nice example of the kind of project you should be aiming for (https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-016-2729-8, https://www.ncbi.nlm.nih.gov/pubmed/20017957).

Apply a technique to new problems: Take one or more of the methods that we have talked about in class, or that we are about to cover, and apply them to a problem. Compare their performance and elucidate why they perform differently, if they do. Do they do a good job on the problem? This is most interesting if you can apply it to some other research question or problem you know about. A big issue here is being sure that you can get the data you need.

Propose new method or variation of existing methods: We have talked about many techniques for solving different problems in computational biology, yet many of them have the room for improvement and innovation. Take one or more of the problems that we learnt in the class and try make modifications to the methods or come up with novel methods that could potentially solve the problem better (e.g. better performance, better computational efficiency, better interpretability, better generalizability, etc.). Given that this is a more challenging direction, we will appreciate the novelty and depth of the proposed method even if the results are not perfect. Comparison to baseline methods is also important but it does not need to extensive as the first type of project.

You don't necessarily have to implement all (or even any) of the algorithms you use. There are several toolkits available with many learning algorithms already implemented in them. However, if you don't do any implementation yourself, we would expect something much deeper in the way of problem formulation or modeling.

If you decide to implement a numerical algorithm, keep in mind that there may be numerical problems such as you've experienced in the homework: for instance, problems may be ill-conditioned or products of probabilities may go to zero (necessitating the use of logs for intermediate values).

Collaboration

Make completely clear in your paper which software you wrote and which software you used but did not write.

Project Ideas

Please starting thinking of ideas now, and avail yourself of office hours to refine your ideas with the help of a TA. Here are some project areas for your consideration.

1. Compare different approaches to predicting the effects of eQTLs using the CAGI 2016 data.
2. Evaluate different methods of predicting the DNase-seq/ATAC-seq measured accessibility of the genome.
3. Evaluate different experimental design methods for the TF k-mer binding data.
4. Produce a method to predict functional genomic variants.
5. Check out the DREAM challenges (http://dreamchallenges.org) for further ideas for projects on computational biology.

Do not hesitate to contact the teaching staff if we can be of help.