

COURSE ANNOUNCEMENT FOR WINTER 2011

BIOINF II/CSE 282: An Introduction to Bioinformatics Algorithms

Instructors: Terry Gaasterland (SIO) and Pavel Pevzner (CSE)

Terry Gaasterland: phone: (858) 822-4600, e.mail: tgaasterland@ucsd.edu
<http://genomes.ucsd.edu/gaasterland.shtml>

Pavel Pevzner: phone: (858) 822-4365, e.mail: ppevzner@cs.ucsd.edu
<http://www-cse.ucsd.edu/users/ppevzner/>

Teaching Assistant:

Anand Patel (patel.anandd+taCSE282@gmail.com)

Boyko Kakaradov (boyko@ucsd.edu)

Time: 5:00-6:20 Mon/Wed, **Place:** EBU3B 2154

Office hours: TG: TBA, PP: (Thursday 3-5), AP: (Tuesday 2-4)

Description: The course assumes some prior background in biology, some algorithmic culture, and some programming skills.

Textbooks: There is no required book for this course. We recommend:

N.C. Jones and P.A. Pevzner. Introduction to Bioinformatics Algorithms. The MIT Press. 2004

For those who have not taken biology we also recommend:

Larry Gonick and Mark Wheelis. The cartoon guide to genetics. Harperperennial Library, 1991

Course Website: <http://cseweb.ucsd.edu/classes/wil1/cse282/>

Grading: *midterm (25% of the score), one course project (40% of the score), and 5 homeworks (35% of the score overall). Homeworks are assumed to be the result of individual work. If for whatever reason you cannot deliver the homework on the due date, you should report it at least a day before the due date. 30% of the maximal number of points is deducted for every day or part of a day that an assignment (or progress report on research project) is handed in too late.*

Class Project: Every bioinformatician faces three major challenges:

- i. Select an interesting biological problem to work on
- ii. Transform a biological problem into a computational one

iii. Solve a computational problem

While the homeworks and the midterm will focus on part (iii), the project will focus on parts (i) and (ii), sometimes the most difficult components of bioinformatics research. This year, the class project encompasses the development of a computational *problem formulation* and initial algorithmic solutions for a biological problem. Transforming a (typically imprecise) biological problem into a well-defined computational problem is a major and often under-appreciated challenge in bioinformatics. Some biologists somehow assume that as soon as the biological problem is formulated, its transformation into a computational problem will be automatically taken care of by bioinformaticians. Moreover, some biologists are not trained to distinguish a “well-defined” (e.g., precisely describing the Input, Output, and the Objective Function) from an “ill-defined” problem.

Every student should work on the project individually since the final report is assumed to be the result of the individual work.

The students are encouraged to choose a class project from (i) the set of the rotation projects announced at the Bioinformatics Program web site, (ii) whatever projects proposed by UCSD faculty, (iii) research papers. Alternatively, a student can design his/her own research project. While the project cannot be based on the ongoing or previous rotation project, we encourage students to use the projects as the preliminary developments for *future* rotation projects in Spring/Summer 2011 (some of the projects in the previous class led to rotations in Spring/Summer and later joining the lab).

The following are the requirements for selecting the projects: (i) the algorithmic problem resulting from this project is new, i.e., was not formulated in previous research papers and was not covered in previous rotations, and (ii) the problem is not trivial, i.e., it represents a graduate rather than undergraduate level problem.

Most of the projects will require significant efforts to transform the described biological problems into computational ones. You are encouraged to talk to the professors who proposed these projects (if any) and to read the relevant literature to better understand the subject area. The students will be guided through various stages of bioinformatics research: formulating the problem, designing the research plan, responding to the criticism of the reviewers, preparing the presentation, etc. It is important that you start working on the project as soon as possible and to file the progress reports reflecting your work on the project according to the following schedule. It is important to complete the project on time (to allow time for presentations) and the schedule below ensures the timely completion. Deviations from this schedule will negatively affect your grade.

Some students may find it difficult to distinguish between well-defined and poorly-defined problems. A couple tips for developing well-defined problems include: (i) writing in a mathematical and/or computer science language (the language commonly used for biology research neither requires nor encourages the precision this project demands) and (ii) being **specific** and **precise** when describing the problem’s objectives, constraints, inputs, and outputs. In addition, please attend a presentation by the TA in the first week of the class to better understand what we mean by well-defined problems.

This year we decided to abandon the traditional “in-class” final exam format and to substitute it with presentations of the class projects.

Class Project Schedule:

- Wednesday, Jan 12. The deadline for selecting the project you plan to work on. By this deadline (or earlier) you will have to send an E.mail to the TA specifying what project you selected. The project will be assigned on the first E.mailed-first served basis, i.e., if your project has been already selected, you will have to choose another project. Try to select the project well before the deadline to make sure that you have a variety of projects to choose from.

Read the recent research papers relevant to the project and make sure that you understand all aspects of the project. Schedule a meeting with a professor who proposed the selected project (if any) to better understand its scope and start working on transforming it into well-defined computational problem(s).

- Monday, Jan 24. Send an E.mail with at most 1-page long computational problem formulation for your project. Your project will be assigned to another student who will write an initial review and will decide whether the problem is well-formulated. The reviewer will provide a feedback (half a page report describing the potential pitfalls of the proposed problem formulation) on Wednesday, Jan 26. You will have a chance to update your problem formulation and submit the revision on Saturday, Jan.29.

Meet with the instructors on Monday, January 31 (in the second part of the day) to discuss the problem formulation you proposed. Both the problem formulations and the reviews (written by the assigned reviewers) will be graded.

- Wednesday, February 2. Send the corrected and extended description of the project specifying the detailed plan of your work in the next 5 weeks. The E.mail should specify your research plan, algorithmic challenges, and software implementation efforts (if any). Prepare a list of a few milestones with deadlines for achieving each milestone.
- Wednesday, February, 23. Send the summary of the progress and the preliminary results. The instructors will evaluate the progress and will send back the critical comments.
- Wednesday, March 2. The deadline for a 5-page long report. You are expected to meet the instructors and TA to make a short powerpoint presentation (based on the 5-page report) describing your project, the remaining challenges you are facing, and the results. The projects will be presented in the class in the last day of the quarter.
- Wednesday, March 9. The in-class presentations of the selected class projects. The assigned reviewers serve as moderators for these presentations. It will be a long class but the pizza will be provided!

Class Project Grading Criteria:

- ability to transform the biological problems into the computational ones.
- ability to formulate well-defined and "solvable" computational problems (there is little utility in well-defined but intractable computational problem)
- ability to review the previous research in the area.
- ability to write a self-contained and concise report.
- ability to propose efficient algorithmic solutions
- sensible implementation decisions (if any)
- sensible benchmarking design (if any)
- clear description of results in the report
- insightful discussion of further directions
- complete bibliographic review