## COURSE ANNOUNCEMENT FOR WINTER 2014

# BIOINF II/CSE 282/BENG 202: An Introduction to Bioinformatics Algorithms

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

**Teaching Assistant: Kai Zhang (kaz007@eng.ucsd.edu)**

**Time: 3:30-4:50 Mon/Wed, Place: CSE (EBU3B) 2154**

**Office hours: PP: (Thursday 3-5), KZ (Tuesday, 12-2)**

**Prerequisites:** The course assumes some prior background in biology, some algorithmic culture (an equivalent of CSE 101), and some programming skills.

**Flipped online class.** In January 2013, California Governor Jerry Brown announced the initaitive to spur online education at UC. Following this initiave, UC established various programs to promote this initiative and to support faculty offering online courses to UC students at both undergrdauate and graduate levels. In particular, UC Innovative Learning Technology Initiative (ILTI) provides professors with substantial support to transform their classes into online offerings available across various UC campuses.

This class is among the first *flipped online* classes at UCSD (many more online classes are being produced and will be offered in Spring 2014). Most lectures in this class will be made available online rather than presented in the classroom. The concept of the flip online class assumes that the students watch lectures at home. This allows the professor and students for more interactions during the class time when students may ask questions about the video lectures and to engage into discussions covering various aspects of the course. 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 in the class.

**Automated homework testing.** This class provides an automated homework testing environment called Stepic, which is inspired by the Rosalind project (www.rosalind.info) aimed at learning bioinformatics through programming. If you are new to programming, please try Rosalind to prepare yourself for homeworks that include programming.

Nearly all HWs in the class will represent programming assignments. Like in real life, there will be no partial credit for programming assignments - you either solve the problem by the deadline (full credit) or not (zero credit). You can use the programming language of your choice to solve HWs. You will have to submit the code that you developed (that passed the automated test) to a code depository before the deadline.

**Textbooks:** You will have access to some chapters of th upcoming textbook:

P.E.C. Compeau and P.A. Pevzner. Bioinformatics Algorithms. 2014

Some lectures in the class are covered in

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: TBA** http://kzhang.org/teach/cse282.

**Grading:** *A midterm (15% of the score) on Wednesday, February 5, a final (pass or fail) on Wednesday, March 19, a class project (35% of the score), and homeworks (50% of the score). If you fail the final, the score of your HWs will be reduced by a factor of 2.*

*Homeworks are assumed to be the result of individual work. If for whatever reason (e.g. a medical or family emergency) you cannot finish the homework on the due date, you should report it at least a day before the due date. Every HW problem is 1 point.*

*Homework Deadlines:*

- Wednesday, Jan 15: Rearrangements.
- Wednesday, Jan 22: Replication origin
- Wednesday, Jan 29: Motifs
- Friday, Feb 7: Alignment (part 1)
- Wednesday, Feb 12: Alignment (part 2)
- Wednesday, Feb 19: Assembly
- Wednesday, Feb 26: Pattern Matching (part 1)
- Wednesday, March 5: Pattern Matching (part 2)

**Reviewing online teaching materials.** Students can review the online teaching materials at their convinience but should be prepared to answer in-class questions about the materials by the following deadlines (to be ready for the Q&A in-class sessions). By default, each student should either ask an in-class question or to answer a question posed by the instructor. If a student prefers a former option, please submit your question by midnight before the class. All questions except for the questions on how to solve the upcoming HWs will be answered in class.

*Deadlines for reviewing online materials:*

- Monday, Jan 13: Rearrangements.
- Wednesday, Jan 15: Replication origin
- Wednesday, Jan 22: Motifs
- Monday, Jan 27: Alignment (part 1)
- Monday, Feb 3: Alignment (part 2)
- Monday, Feb 10: Assembly
- Monday, Feb 17: Pattern Matching (part 1)
- Monday, February 24: Pattern Matching (part 2)

**Logistics of the Class Project:** To work on the class projects, the students are encouraged to form groups (no more than 4 students in each group). We recommend that each group has students with both biological and computational backgrounds. Every group should work on the project without comunicating with other groups.

**Description of the Class Project:** Every bioinformatician faces three major challenges:

i. Select a biological problem to work on

ii. Transform a biological problem into a computational one

iii. Solve a computational problem

This year, the class project encompasses the development of a computational *problem formulation* and algorithmic/software solutions for one of the Massive Online Open Problems presented at the stepic website. The goal of the project is to prepare (and possibly submit) a research paper describing a solution to a MOOR project. 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.

Some students may find it difficult to distinguish between well-defined and poorly-defined problems. A couple of 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), (ii) being **specific** and **precise** when describing the problem's objectives, constraints, inputs, and outputs, and (iii) the problem formulation should be **short** and **elegant** (if, after introducing all definitions, it exceeds a paragraph, you are probably on a wrong track).

The projects may require significant efforts to transform the described biological problems into computational ones. You are encouraged to read the relevant biological 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, writing paper, etc. It is important that you start working on the project as soon as possible and 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.

**Class Project Schedule:**

- Wednesday, Jan 15. 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 by multiple groups, 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. Start working on transforming it into well-defined computational problem(s).

- Monday, Jan 27. Send an E.mail with at most 1-page long computational problem formulation(s) and work plan for your project. Your project will be assigned to another group 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 and work plan) on Wednesday, Jan 29. You will have a chance to update your problem formulation and submit the revision on Friday, Jan.31.

Meet with the instructors on Thursday, Feb 6 to discuss the problem formulation you proposed. Both the problem formulations and the reviews (written by the assigned reviewers) will be graded.

- Sunday, February 9. 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. Prepare a list of a few milestones with deadlines for achieving each milestone.
- Friday, February 22. Send the summary of the progress and the preliminary results. The instructors will evaluate the progress and will send back the critical comments.
- Friday, February 28. The deadline for a 5-page long paper. 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 12. 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
- sensible benchmarking design
- clear description of results in the report
- insightful discussion of further directions
- complete bibliographic review