COURSE ANNOUNCEMENT FOR WINTER 2022

BENG181/CSE 181/BIMM 181
Molecular Sequence Analysis
https://canvas.ucsd.edu/courses/32442/

Instructor: Pavel Pevzner

- phone: (858) 822-4365
- e.mail: ppevzner@ucsd.edu
- web site: https://bioalgorithms.ucsd.edu/

Teaching Assistants:

- Vikram Sirupurapu (email : vsirupur@eng.ucsd.edu , Ph : +1 858 900 6992)
- Ishaan Gupta (i3gupta@ucsd.edu, Ph : 810 529 6351)

Course Details:

Item	Days	Time	Location	Zoom link
Class	Monday and Wednesday	5:00 – 6:20 PM PT	CENTR 113 or online	https://ucsd.zoom.us/j/94143930247
Discussion Session	Monday	4:00 – 4:50 PM PT	CSB 001 or online	https://ucsd.zoom.us/j/99665771725

Please see a weekly schedule at the end of this syllabus.

Piazza:

1. You should be automatically added to Piazza (Course ID = BIMM_CSE_BENG_181_WI22_A00). If you are not added, please contact the TAs via email.

2. Piazza is only used for logistics, announcements and general course related doubts. Piazza shall not be used to discuss weekly homework problems (which have to be completed individually).

Office hours:

Item	Days	Time	Location*	Zoom link

Pavel Pevzner	Tuesday	3:00 – 5:00 PM PT	Online only	https://ucsd.zoom.us/j/96574970505
Vikram Sirupurapu	Friday	5:00 – 6:20 PM PT	CSE B270A or online	https://ucsd.zoom.us/j/97858581844
Ishaan Gupta	Thursday	3:00 – 4:30 PM PT	CSE B270A or online	https://ucsd.zoom.us/j/95257293825

TAs can hold office hours under special circumstances by appointment online.

Prerequisites: The course assumes some prior background in biology, some algorithmic culture (CSE 101 course on algorithms as a strictly enforced prerequisite), and some programming skills.

Flipped online class. The *Innovative Learning Technology Initiative* (ILTI) at University of California encourages professors to transform their classes into online offerings available across various UC campuses. Dr. Pevzner is funded by the ILTI and NIH to develop new online approaches to bioinformatics education at UCSD. This class was the first fully flipped class at UCSD. Since 2014, well before the COVID-19 pandemic, all learning materials in this class are available online (rather than presented in the classroom) so that students can start learning *before* the class starts. For reference, watch the TedX video "The Era of Online Learning" where professor Niema Moshiri (who was a student in this class just seven years ago) discusses the advantages of flipped courses over the traditional classroom format.

This class closely follows the textbook *Bioinformatics Algorithms: an Active Learning Approach* that has been adopted by 160+ instructors from 40+ countries.

Automated homework testing. Computer science legend Donald Knuth once said "*I don't understand things unless I try to program them.*" We share his "*Learning through programming*" approach - all HWs in the class represent coding challenges that are described in the textbook.

This class provides an automated homework testing environment inspired by the *Rosalind project* aimed at learning bioinformatics through programming. 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 any programming language of your choice to solve the HWs.

Since Rosalind is not supported anymore, we will move from the Rosalind platform to the more advanced Stepik platform. You will have to submit all your solutions of the HW programming challenges at Stepik. Stepik currently accepts just the solutions of the HW programming challenges, not the code files used to compute these solutions: the code files that resulted in successful solutions on Stepik should be uploaded to the corresponding assignments in Canvas

***prior to the HW deadline for follow-up plagiarism checking* (*no exceptions*).** Do not upload the code if your solution was not successful on Stepik.

No hints on how to solve HWs will be provided before the HW deadline. The textbook and extensive FAQs are designed in such a way that they contain all information (including explicit or implicit hints) needed for you to succeed in HWs. However, if you fail to solve one of the HW problems, you will have an opportunity to submit a question that describes a learning breakdown that you experienced and explains why this breakdown makes it difficult to solve a specific problem. If this description presents a well-formulated summary of your learning breakdown (see below) you will get a credit (from 50% to 100% of a point) even if you failed to solve a specific HW problem! This unusual credit is well-deserved since asking a good question about a problem is sometimes as difficult as solving it! You can submit a description of a learning breakdown for at most a single unsolved problem per each week's HW.

HW Problem Register. To help with solving HW problems, students in this class will have access to the online book *Problem Register for "Bioinformatics Algorithms: An Active Learning Approach"* written by Parker Côté and Ryan Eveloff. Parker (now at Broad Institute) and Ryan (now a graduate student at UCSD) took this class in Winter 2021 and have done a great job to unify the descriptions and test cases for all HW problems and help future students during the time they work on HWs.

Do not attempt to code before you understand how the algorithm works! It is important that you understand the ideas behind each algorithm that you implement in this course. We do not want you to blindly code a "line-by-line" implementation of pseudocode to pass the automatic grader without understanding how the algorithm behind this pseudocode works. That is why we ask you **not to submit** the code unless you can write down an explanation of how your solution works in your own words without opening the textbook, e.g., write pseudocode and outline the key idea of your solution without copying pseudocode from the textbook.

We will waive the Final Exam for students who follow this approach. To test whether you follow this approach, most assessments in this class (including, quizzes, the midterm and the final) will be designed to test how well you understand your OWN previously submitted HW programs (see below). The decision on whether to waive the Final Exam will also take into account how well students answer questions raised by the professor during class meetings.

Midterm and Final. The midterm and final exams will consist of newly designed programming challenges that represent modifications of problems that have been previously given as HWs in the class. Therefore, to solve a novel problem A* that originated from a HW problem A, you merely need to slightly modify the original code for A (that you have already submitted). If you successfully solved problem A on its corresponding HW, you **are required** to modify your submitted code for A instead of implementing A* from scratch and mark all new lines where you made changes as compared to the code for A. Thus, you will need to have the code from all your previous HWs available when you start the midterm/final. If you did not solve problem A on its corresponding HW, you can solve problem A* from scratch during the midterm/final.

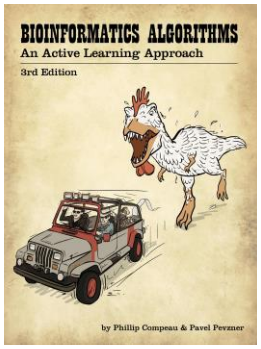
Non-graded quizzes. Quizzes will either be given in the format described above, or will ask you to describe one of your previous HW solutions in your own words, or will ask you to solve simple problems that test your knowledge of some basic concepts. Quizzes will not be graded but instead used for deciding which students are not eligible for waiving the final exam. Most quizzes will be given on Mondays (if time allows) but some quizzes may be issued on Wednesdays, depending on when we have time left after the communication sessions or invited lectures. We will not have quizzes during some (busy) weeks.

Avoid anti-correlation between HW and midterm/final scores! These quizzes and exams should represent a simple task for students who followed the suggestions in the previous subsection, "**Do not attempt to code before you understand how the algorithm works!**" However, they will be difficult tasks for students who submitted HWs without deep understanding of the algorithms behind them or (worse!) for students whose HWs do not represent an independent effort. All midterm, final, and quiz solutions will be checked for plagiarism.

If it turns out that your high HW score "anti-correlates" with your low midterm/final score and the results of the quizzes, it likely means that you have not taken the above subsection seriously or worse, the HWs you submitted do not represent independent work. The midterm and final will be designed in such a way that you *will have to* modify the code in one of your previously submitted HWs to solve each of the problems. Therefore, the best way to prepare for the midterm/final and to pass this class is to make sure that you submit your *own independent HWs* each week and understand how it works.

Communication skills in bioinformatics. Communication skills are important in every discipline but they are even more crucial in interdisciplinary fields like bioinformatics. That is why you will be given (partial) credit even for unsolved HW problems if you master the art of asking well-formulated questions that describe your learning breakdowns and that prevent you from solving these problems.

Textbook: Phillip Compeau and Pavel Pevzner. *Bioinformatics Algorithms: An Active Learning Approach.* 3rd edition. Active Learning Publishers 2018.


BIOINFORMATICS ALGORITHMS
An Active Learning Approach
3rd Edition
by Phillip Compeau & Pavel Pevzner

We will be using the 3rd edition of the book (a chicken-dinosaur on the cover) rather than the outdated 2nd or 1st editions.

Although it was a required textbook for CSE 181 in the past, this year you DO NOT NEED to buy the hard copy of this textbook. Since some students are now located in various countries where the book may not be available, Active Learning Publishers partnered with Stepik to make the book (along with all additional materials that include videos, programming challenges, etc.) available as an online MOOCBook at the discounted price of $69.95. Note that the online textbook on Stepik slightly differs from the hardcopy version as it represents an intermediate version between the 3rd and 4th edition that is being prepared.

Please go to BioinformaticsAlgorithms@UCSD at Stepik to enroll in the course.

Online resources:

The Stepik platform provides all online resources for this course. You can also find some additional resources online at the following locations:

- A link to most lessons is available from the Bioinformatics online specialization web page at coursera.org.
- Private YouTube channel: http://www.youtube.com/user/bioinfalgorithms/
- FAQs: http://bioinformaticsalgorithms.com/faqs.htm
- Rosalind platform

Use of external packages to solve HWs. You are not allowed to use any external packages (e.g. NumPy, JGraphT, etc.) to solve the HW programming challenges. Using the native implementations of basic data structures (e.g. hashmap/dictionary, array/list, queue, stack, heap, etc.) is fine, but using things like full-fledged graph libraries is not allowed. That being said, you are free to implement your own data structures, e.g., you can implement your own Node/Edge/Graph classes as you see fit. Make sure that, outside of things you implement yourself, you only use native basic data structures to solve the problems.

Grading: The total score will be composed of the following components:

- **HWs** (70% of the score). Homeworks will be issued each week and will include from 5 to 8 coding challenges. Homeworks are assumed to be the result of individual work. You are NOT allowed to search for solutions of home works on any online resources. HW submissions will be subjected to the automatic plagiarism checking. You can discuss HWs with your classmates *after* the HW deadline when you are preparing the survey question.
  - **Scoring HWs.** Every HW problem is 1 point. If you solved N out of M problems in a specific HW, your score for this HW is computed as (N/M)*100%. Your total HW score is the average of the scores of all individual HWs (after dropping your lowest HW score; see below).
  - **Communication skills** in bioinformatics. If you haven't solved all HW problems in a specific HW, you can add at most 1 point to this HW by asking a well-formulated question (as described elsewhere in this syllabus).
- **Midterm** exam (20% of the score)
- **Assessments based on invited lectures** (10% of the score).
- **Final** exam (Pass or Fail). IMPORTANT: Failing the Final exam implies failing the

class independently of your other scores.

Missing classes. We understand that you may miss some sessions of the class due to unforeseen circumstances, illness, graduate school interviews, etc. To help you deal with these circumstances, your overall HW score will be computed from your top *n*-1 individual scores, where *n* is the total number of HWs in this class. Thus, you can miss one HW and one quiz in this class, no questions asked, to account for medical or family-related absences, job, and graduate school interviews, etc. However, you will have to provide an official justification for each missed session if you miss more HWs than specified above.

Learning breakdowns. An important goal of this class is to teach students how to diagnose their INDIVIDUAL *learning breakdowns* and to resolve them by asking well-formulated questions. A learning breakdown refers to a concept that students have struggled with even after spending significant time trying to address this breakdown (e.g., thinking deeply about this concept, checking FAQs and other learning materials, etc.). In this class, we will particularly focus on learning breakdowns that prevented students from solving some HW problems.

Communication skills. Each student who experienced a learning breakdown may file a *single* well-formulated question related to their breakdowns by the Survey deadlines specified below. We understand that you may have multiple breakdowns, instead of just a single breakdown: and there will be time to address all of these concerns during our class meetings. However, the partial HW credit (up to 1 point) will be given only for a description of a single breakdown that you have submitted.

It is important that you invest time in formulating the question so that the instructor can address it based *only on your formulation,* without additional clarification. The description of your breakdown should explain how it prevented you from solving a specific HW problem.

There will be nine Survey deadlines in this class. Please file your questions at the class canvas website, under Assignments>Surveys. A "well-defined question" means that your peers (and the instructor!) are able to understand the specific difficulty you are having and to help you to overcome the learning breakdown. For example, "I don’t understand how this algorithm works, can you please explain it again?" is not a well-formulated question and will not be given credit for the communication session because it does not describe your *specific* learning breakdown, does not allow an instructor to diagnose what caused it, and does not describe why this specific breakdown prevented you from solving a specific HW problem.

Some students may prefer not to file any learning breakdowns (e.g., students who did not experience any learning breakdowns on a given week’s HW) and this is perfectly fine. Note that all students will be answering questions of the instructor and other students during class meetings or via email and answers to these questions are taken into account when deciding on whether to waive the Final.

Learning breakdowns versus curiosity questions. Learning breakdowns reflect challenges that make it difficult for a student to solve a specific HW problem - you have to specify which

HW problem and explain how your learning breakdown prevents you form solving this problem. “Curiosity questions" like:

- Are there any alternative ways of estimating the location of the replication origin?
- How do we select the size of the window in the Clump Finding Problem?
- How do we select the constant *K* for the partial suffix array?
- Why does this example assume that *K*=5 and not 10?

are not classified as learning breakdowns because they do not affect the understanding of the follow-up materials. If you only face curiosity questions while going through the chapter, this means you have not really had a breakdown. Also, you cannot file questions that simply repeat “Exercise Breaks," “STOP and Think" boxes, FAQs, or indirectly ask to provide hints for homework problems.

All books have small errors that should not be filed as learning breakdowns unless an error prevents you from solving a HW problem.

Filing a Survey report. As discussed above, you can optionally file a Survey report by the one of the Survey deadlines specified below. This report specifies a learning breakdown that a student is struggling with. The following information is required for each Survey report:

- Information about the HW problem you failed to solve (specify the name of the problem) because of the specific learning breakdown you intend to describe.
- A detailed description of the learning breakdown (pointing to a specific page/paragraph).
- A well-formulated question that will explain to the instructor how to help you to address this specific learning breakdown.

Additionally, we ask that your Survey report follows these guidelines:

- Your description should explain how this specific breakdown prevented you from solving the specific HW problem. The description of the breakdown should describe the algorithmic rather than programming challenges – e.g., questions like “How do I implement the Burrows-Wheeler Transform in Java” will not be given credit.
- Your breakdown report should be self-contained, i.e., the instructor should be able to understand the cause of the breakdown without additional verbal clarifications from you. If you file a breakdown/question that has been already addressed in the available resources, there will be no credit for the communication session.
- It is a student’s responsibility to check the FAQs and Charging Stations (not to mention the text of the entire chapter) to ensure that this question has not been addressed yet (otherwise, there will be no credit for the communication session).
- Your questions should refer to the textbook rather than the videos (or power points) since videos represent incomplete and error-prone versions of the learning materials. It is important that the question relates to the specific learning breakdown (and a specific paragraph in the textbook) and specific HW

problem rather than being an open-ended question. For example, we appreciate questions like “What is the future of this sequencing technology?” or “Can I apply Hidden Markov Models to gene prediction?” and they will be answered in the follow-up communication session. However, no credit will be given for such general open-ended questions.

You will be given a credit (up to **1 point**) for good questions that comply with the above description provided the instructor understands what your question is about.

Reviewing online materials. Students can review the online materials (Stepik, FAQs, etc.) at their convenience (all materials are available on the first day of classes) but should be prepared to answer questions about all the materials by the Communication Session dates specified below. The Study periods below are merely the suggestions to help you get organized for this class - you can work on whatever schedule you find convenient, for example, you can solve all HWs in the first week of classes.

Academic Integrity. To detect instances of academic integrity violations in programming assignments we will use third-party plagiarism detection software. You may find the tutorial “Plagiarizing the source code” at the link:

https://libraries.ucsd.edu/assets/elearning/cse/cseplagiarismexternal/story.html

All the work in the course should be your own. Since plagiarism was detected in previous sessions of this class (with serious long-term consequences for the students involved), we invest significant effort in checking your code and comparing it with a database of existing solutions. Using various web resources (that provide solutions to coding challenges) for solving HWs is considered a violation of the academic integrity policy.

Please do not post your solutions on the Internet and do not share your solutions with classmates since this may trigger a violation of the academic integrity policy, for example in the case when your schoolmate uses your solution in a homework. Please note that, if you solved a HW before the start of the class (e.g., in the Fall 2020 quarter) and used web resources for solving it, this may also trigger a violation of the academic integrity policy. If this is the case, you have to redo the program from scratch since otherwise it may be marked as a violation by our plagiarism checking tool.

Invited lectures. Since 160+ instructors who adopted our textbook cover similar materials, there is an opportunity to share various educational materials that extend the topics covered in the textbook and enhance the educational experience of students across various universities.

In Winter 2022, we will cover additional materials contributed by the following instructors:

- Phillip Compeau, Department of Computational Biology, **Carnegie Mellon University**
- Alexey Gurevich, Center for Algorithmic Biotechnology, **Saint Petersburg University**

- Ben Raphael, Computer Science Department, **Princeton University**,
- Rachel Sherman, Department of Computer Science, **John Hopkins University**

Here is information about the research-oriented lectures by collaborating professors from other universities that use the same textbook:

- Phillip Compeau (CMU): ***How do we measure gene expression: transcript assembly and quantification.***

  This lecture is related to the topics of sequence comparison (chapter 5), clustering (chapter 8), and read mapping (chapter 9).
- Alexey Gurevich (SPBU): ***How do we compare LONG genomic sequences?***

  This lecture is related to the topics of genome assembly (chapter 3) and sequence comparison (chapter 5).
- Pavel Pevzner (UCSD): ***The long-read revolution in genome sequencing.***

  This lecture is related to the topic of genome assembly (chapter 3).
- Ben Raphael (Princeton): ***Cancer evolution.***

  This lecture is related to the topics of genome assembly (chapter 3), genome rearrangements (chapter 6), and evolutionary tree construction (chapter 7)

In addition, we will have two COVID-19-related lectures that discuss how bioinformatics contributes to emerging problems in personalized immunogenomics and antibody discovery:

- Pavel Pevzner (UCSD). ***Personalized immunogenomics***
- Stefano Bonissone (Digital Proteomics). ***Immunoinformatics: how informatics can be central to antibody discovery and characterization.***

Course schedule (subject to change). Below is the class schedule.

Homework deadlines are at 11:59 pm on the specified dates. Please note that HWs do not necessarily include ALL PROBLEMS from a given chapter. Read information below for the list of excluded problems for each chapter – the HWs contain only 59 problems while the relevant chapters contain over 100 problems.

The "excluded problems" here are listed based on the 3rd edition of the book and the Rosalind platform; however, there are a few differences between this edition and the Stepik course (which is based on an in-progress 4th edition of the book). Please always check the NAME of the excluded problem (rather than its number) to see which problems are excluded. In the event of discrepancies, please go with what is on Stepik; for a list of which exact problems are required in the course, please see this spreadsheet (just the problems labelled with "Kept" in their "Used in class?" column are required).

As discussed above in the section "Automated homework testing", please submit solutions generated by your code to Stepik, and the code files used for these problems to Canvas.

Survey deadlines are at 3 pm (PT) on the specified dates.

Replication Origin (Chapter 1)

- Communication Session, Mon, Jan 10
- HW deadline: Tue, Jan 11
- Survey deadline: Wed, Jan 12, 3 pm
- Study: Mon, Jan 3 - Tue, Jan 11
- Excluded HW problems: 1C (Find reverse complement), 1D (Find all occurences of a pattern in a string),1G (Hamming distance), 1J (Find frequent words with mismatches and reverse complements), 1L (PatternToNumber) and 1M (NumberToPattern)
- Total: 8 problems.
- Wed, January 5. Seminar - Pavel Pevzner ***Personalized immunogenomics***

Martin Luther King Jr. Day. Monday, January 17 (no class)

Regulatory Motifs (Chapter 2)

- Communication Session, Wed, Jan 19
- HW deadline: Thu, Jan 20
- Survey deadline: Fri, Jan 21, 3 pm
- Study: Tue, Jan 11 – Tue, Jan 18
- Excluded HW problems: 2D (Greedy motif search), 2E (Greedy motif search with pseudocounts)
- Total: 6 problems.
- Wed. January 12. Seminar- Stefano Bonissone (Abterra Biosciences). ***Immunoinformatics: how informatics can be central to antibody discovery and characterization.***

Assembly (Chapter 3)

- Communication Session, Mon, Jan 24
- HW deadline: Tue, Jan 25
- Survey deadline: Wed, Jan 26, 3 pm
- Study: Tue, Jan 18 – Tue, Jan 25
- Excluded HW problems: 3A (Generate the k-mer composition of a string), 3B (Reconstruct a string from its genome path) 3K (Generate contigs), 3L (Generate a string spelled by a gapped genome path), and 3M (Generate all maximal non-branching paths in a graph).
- Total: 8 problems.
- Wed, Jan 26. Seminar- Pavel Pevzner (UCSD) ***The long-read revolution in genome sequencing.***

Alignment, Part 1 (Chapter 5, ending before (not including) “Penalizing Insertions and Deletions")

- Communication Session, Mon, Jan 31
- HW deadline: Tue, Feb 1
- Survey deadline: Wed, Feb 2, 3 pm
- Study: Tue, Jan 25 – Tue, Feb 1
- Excluded HW problems: 5A (Find the minimum number of coins)

- Total: 8 problems (5B-5I). You may want to solve 5N from the next HW (Finding topological ordering of a graph) before embarking on this HW since it may help you to solve this HW.
- Wed, Feb 2. Seminar- Invited talk: Alexey Gurevich (Saint Petersburg University) ***How do we compare LONG genomic sequences?***

Alignment, Part 2 (Chapter 5, starting from (including) “Penalizing Insertions and Deletions")

- Communication Session, Mon, Feb 7
- HW deadline: Tue, Feb 8
- Survey deadline: Wed, Feb 9, 3 pm
- Study: Tue, Feb 1 – Tue, Feb 8
- Total: 5 problems (5J-5N)

***Midterm exam*: Wed Feb 9**

Rearrangements (Chapter 6)

- Communication Session, Mon, Feb 14
- HW deadline: Tue, Feb 15
- Survey deadline: Wed, Feb 16, 3 pm
- Study: Tue, Feb 8 – Tue, Feb 15
- Excluded HW problems: 6I (GraphToGenome) 6J (2-BreakOnGenomeGraph) and 6K (2-BreakOnGenome)
- Total: 8 problems
- Wed, February 16. Seminar- Invited talk: Ben Raphael (Princeton) ***Cancer Evolution***

President’s Day. Monday, February 21 (no class)

Detecting Mutations, Part 1 (Chapter 9 before (not including) “Inverting Burrows-Wheeler Transform")

- Communication Session, Wed, Feb 23
- HW deadline: Thu, Feb 24
- Survey deadline: Fri, Feb 25, 3 pm
- Study: Tue, Feb 15 – Tue, Feb 22
- Excluded HW problems: 9F (Finding shortest non-shared substring), 9H, (Pattern Matching with the Suffix Array)
- Total: 7 problems (9A-9E and 9G, 9I)

Detecting Mutations, Part 2 (Chapter 9, starting from (including) “Inverting Burrows-Wheeler Transform")

- Communication Session, Mon, Feb 28
- HW deadline: Tue, March 1
- Survey deadline: Wed, March 2, 3 pm
- Study: Tue, Feb 22 – Tue, March 1
- Excluded HW problems: 9P (TreeColoring), 9Q (Partial Suffix Array of a String), and 9R (Suffix Tree from a Suffix Array)

- Total: 6 problems (9J-9O)
- Wed, March 2. Seminar- Invited talk: Phillip Compeau (Carnegie Mellon University) ***How do we measure gene expression: transcript assembly and quantification.***

Clustering (entire Chapter 8) and *Hidden Markov Models* (Chapter 10) before (not including) "Classifying proteins with profile HMMs."

- Communication Session, Mon, March 7
- HW deadline: Tue, March 8
- Survey deadline: Wed, March 9, 3 pm
- Study: Tue, March 1 – Tue, March 8
- Excluded HW Problems. Implement all problems from Chapter 8 except for
- 8D, (Implement the Soft k-Means Clustering Algorithm) and 8E (Hierarchical Clustering). **Do not** implement any problems from Chapter 10 except for the only problem 10C (Viterbi algorithm).
- Total: 4 problems (8A-8C and 10C)
- Wed, March 9. Seminar - Invited talk: Rachel Sherman (Johns Hopkins University) ***What's in a mutt: an intro to dog DNA analysis.***

Final: Mon, March 14, 7 pm -10 pm

Schedule at a glance:

- Mon, Jan 3. First class
- Wed, January 5. Pavel Pevzner. ***Personalized immunogenomics***

- Mon, Jan 10. Communication Session (Replication)
- Wed, Jan 12. Stefano Bonissone (Abterra Biosciences). ***Immunoinformatics: how informatics can be central to antibody discovery and characterization.***

- Mon, Jan 17 *Martin Luther King Jr. Day* (no class)
- Wed, Jan 19, Communication Session (Motifs)

- Mon, Jan 24, Communication Session (Assembly)
- Mon, January 26. Pavel Pevzner. ***The long-read revolution in genome sequencing.***

- Mon, Jan 31, Communication Session (*Alignment, Part 1*)
- Wed, Feb 2. Invited talk: Alexey Gurevich (Saint Petersburg University) ***How do we compare LONG genomic sequences?***

- Mon, Feb 7, Communication Session (*Alignment, Part 2*)
- **Wed, Feb 9, *Midterm***

- Mon, Feb 14 Communication Session (*Rearrangements*)
- Wed, February 16. Ben Raphael (Princeton) ***Cancer Evolution***

- Mon, February 21, President’s Day (no class)
- Wed, Feb 23, Communication Session (*Detecting Mutations, Part 1*)

- Mon, Feb 28, Communication Session, (*Detecting Mutations*, *Part 2*)
- Wed, March 2. Phillip Compeau (Carnegie Mellon University) ***How do we measure gene expression: transcript assembly and quantification.***

- Mon, March 7, Communication Session (*Clustering* and *Hidden Markov Models*)
- Wed, March 9. Rachel Sherman (Johns Hopkins University) ***What’s in a mutt: an intro to dog DNA analysis.***

- **Mon, March 14 (7 pm -10 pm) Final**