

Preparing for the CS 598AGB Final Exam

Overview: Please review all the material in the webpage for the course, including all the class presentations, assigned reading, and homework assignments. The final exam will be comprehensive and closed book, so you are expected to be able to answer any question that is covered by any of the assigned reading or in the homework. Note that you need to understand and apply the concepts and theory from the course. You also need to be able to provide mathematical definitions (of models, concepts, etc.), explain why a method is or is not statistically consistent, describe and use algorithms you've been taught, remember what is in a paper you've been assigned, etc. Please come to class with questions so we can go over the material *before* the exam!

Assigned reading from the textbook

- Chapters 1-3
- Chapter 4.1-4.4, 4.6, 4.7
- Chapter 5
- Chapter 6.1, 6.3, 6.5, 6.6, 6.8
- Chapter 7.1-7.6, 7.8-7.9
- Chapter 8
- Chapter 9.1-9.13
- Chapter 10.1-10.5
- Chapter 11.1, 11.2.1-11.2.7
- Chapter 14

Assigned papers: For the papers you were assigned to read, you should remember what they were about, what the authors claimed, what kinds of arguments they used to support their conclusions, etc. If they performed simulation studies you should know things about the simulations: for example, how big the datasets were, what methods they used, and what their performance criteria were. You should be able to make comparisons between papers, and have an opinion on whether you agree with the conclusions. For theoretical papers, you should know the theorems that are established. You should also be able to sketch the proof.

- “The Hobgoblin of Phylogenetics”, by David M. Hillis, John P. Huelsenbeck, and David L. Swofford (Nature, Vol. 369, 2 June 1994)

- “Cases in which parsimony and compatibility methods will be positively misleading”, by Joseph Felsenstein, *Systematic Zoology*, Volume 27, No. 4 (1978), pp. 401-410.
- “Is It Better to Add Taxa or Characters to a Difficult Phylogenetic Problem?”, by Anna Graybeal, *Systematic Biology*, 47(1):9-17, 1998.
- “Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega”, by Sievers et al., DOI 10.1038/msb.2011.75, *Molecular Systems Biology* (2011) 7: 539.
- “Estimating species trees from unrooted gene trees”, by L. Liu and L. Yu, *Syst. Biol.* 60(5):661-667, 2011.
- “How to apply de Bruijn graphs to genome assembly”, Phillip E C Compeau, Pavel A Pevzner, and Glenn Tesler, *Nature Biotechnology*, Volume 29, number 11, pages 987-991, 2011
- “Genome assembly reborn: recent computational challenges”, by Mihai Pop, *Briefings in Bioinformatics* (2009) 10(4):354-366.
- Papers 84, 95, and 114 from my publication list

You are also expected to have read the papers presented by students in the class, but for these papers the questions will be somewhat less detailed.

Homework: Please review all the homework you were assigned. But also see the textbook homeworks, including ones you were not assigned.

Quizzes: Quiz 2 (given below) had you answering three of the following four problems. Be able to answer all four for the final exam.

1. Sketch the proof for why UPGMA is not statistically consistent under the Jukes-Cantor model.
2. Describe a polynomial time and statistically consistent method for estimating a rooted species tree from a collection of rooted gene trees, under the assumption that the rooted gene trees are drawn from the distribution defined by the multi-species coalescent model. Sketch the proof for why your algorithm is statistically consistent.
3. Give a polynomial time and statistically consistent method for estimating an unrooted gene tree from a collection of sequences that evolve down the gene tree under the Jukes-Cantor model of sequence evolution. Sketch the proof for why your algorithm is statistically consistent.
4. Give a polynomial time dynamic programming (DP) algorithm to figure out which player has a winning strategy for the following two person game. The starting position is two piles of rocks, where at least one pile has at

least one rock. The player that makes both piles empty by removing one or more rocks wins. When it is a player's turn, they have three options – they can remove two rocks off one pile, one rock off each of two piles, or one rock off one pile. Your DP algorithm should let us know who has a winning strategy – player 1 or player 2 – as a function of the number of rocks on each pile.