



JOHNS HOPKINS

WHITING SCHOOL
of ENGINEERING

Computational Protein Structure Prediction and Design ChemBE 540.414/614 — Spring 2016

- Instructor: Professor Jeffrey J. Gray
Maryland Hall 208, jgray@jhu.edu, 410-516-5313, [@jeffreyjgray](https://twitter.com/jeffreyjgray)
- Office Hours: M 4:15-5:15 p.m. or by appointment
- TA: Joey Lubin, Maryland Hall 306, jlubin3@jhu.edu
- TA Office Hours: T 4:00-5:00 p.m. or by appointment
- Class Meetings: MW 3:00-4:15 p.m., Hodson 305 (lectures) and Krieger 307 (workshops)
- Course Websites: <http://graylab.jhu.edu/courses/540.414>, *Piazza*, and *Blackboard*
- Hashtag: [#ChemBE414](https://twitter.com/ChemBE414)
- Software: *PyRosetta*, available for download (free) from <http://www.pyrosetta.org> and available in Krieger 307 and the Krieger lab.
- PyMOL*, available for download (free for use as a full-time student) from <http://www.pymol.org> and available in Krieger 307 and the Krieger academic lab.
- Workbook: *The PyRosetta Interactive Platform for Protein Structure Prediction and Design: A Set of Educational Modules*, J. J. Gray, S. Chaudhury, J. Labonte & S. Lyskov. Second edition, PyRosetta4 update, 2016. Available on Amazon or *CreateSpace*. PDFs of the individual chapters are available on <http://www.pyrosetta.org/tutorials>.
- Background Texts: (on reserve at the MSEL)

- *Introduction to Protein Structure*, C. Branden & J. Tooze, Garland Publishing, Second edition, 2012. (Available to rent on Amazon)
- *Molecular Modeling and Simulations*, Tamar Schlick, Springer, 2002.
- *Protein Engineering and Design*, Eds. Sheldon Park & Jennifer Cochran, CRC Press, 2010.

Literature readings will be assigned weekly. Most will be posted as PDF files on the website.

Overview: The prediction of protein structure from the amino acid sequence has been a grand challenge problem for over fifty years. With recent progress in research, it is now possible to blindly predict many protein structures and to design new structures from scratch. *This class will introduce the fundamental concepts in protein structure, biophysics, optimization and informatics that have enabled the breakthroughs in computational structure prediction and design.* Problems covered will include protein folding and docking, design of ligand-binding sites, design of turns and folds, design of protein

interfaces. Class will consist of lectures and hands-on computer workshops. Students will learn to use molecular visualization tools and write programs with the PyRosetta protein structure software suite, including a computational project. Programming experience is recommended.

Prerequisites:

- AS.020.305 Biochemistry. Students should have basic knowledge of biological molecules and the central dogma of biology.
- EN.540.203 Engineering Thermodynamics or equivalent physical chemistry course. Students should understand the first and second laws of thermodynamics.
- Computer programming skills highly recommended. The majority of the homework assignments and the final project will require programming in Python. Students familiar with a different computing language will probably be able to learn the Python basics and adapt to this language, as it is a recently developed language optimized for rapid and easy programming. Some distinguishing advantages of Python include clear, readable syntax; intuitive object orientation, full modularity, high-level dynamic data types, and extensive standard libraries. It is suggested that you review Python basics during the first week of class (see <http://docs.python.org/tutorial>).
 - To bolster your Python programming skills, the TA will hold an optional **Python Primer**, tentatively scheduled for Thursday, February 2 from 4:00-5:15.

Workshops: To gain hands-on experience with protein visualization and modeling tools, many classes will be held in a computer classroom. You will be expected to work through a tutorial assignment during the workshop. Selected questions and additional computer exercises will be a part of the weekly homework assignment.

Grading: Grades will be determined from an *absolute* scale: 90% = A-; 80% = B-; 70% = C- etc. We reserve the right to shift these percentages down (but not up). For example, if a particular exam is more difficult than we originally planned, an A- may be achieved with an 88. It is not advisable to rely on a grade shift, however.

Your course grade will be computed as follows:

Homework	20%
Exam I	25%
Exam II	25%
Project	25%
Quiz	1%
<u>Participation</u>	<u>4%</u>
Total	100%

Homework: Homework will be assigned regularly and will be due after approximately one week. Half of the homework grade will be awarded simply for completing the problem; the other half will be awarded based on the correctness of the result and approach. Homework not received by five minutes after the start of class on the due date will be penalized 25%; an additional 25% penalty will be incurred for every additional day late.

Homework assignments should be submitted by uploading to Blackboard. Analytical work can be done by hand and then scanned. Appropriate software should be used to make figures and tables as needed (PyMOL, Excel, Matplotlib, MATLAB, or R) instead of hand-sketching/estimating answers. Computer code and output supporting the homework assignments should also be uploaded to Blackboard, following code templates when provided.

Exams: Two exams will be given, the first during a class period and the second during final exam week.

Quiz: There will be one short quiz during the first few weeks of class (see homework 1).

Regrades: Any regrade requests (exam or homework) must be submitted in writing within one week of the return of the exam or assignment. Homework regrades will be performed by the grader, with review by the professor. Exam regrades will be performed by Prof. Gray. Note that in the event of a regrade, the entire submission (not just the item of contention) is subject to regarding, at the discretion of the reviewer.

Class Participation: I will use a variety of individual and small-group active learning exercises in class, therefore you are expected to participate. I will frequently call on students for answers and examples. Thinking time will be provided, but answers of “I don’t know” are not allowed. Finally, you are encouraged to ask questions during lecture.

Project: For the course project, you will be able to choose a structural modeling or design problem of your own interest to pursue using computational tools. Projects will be completed in groups of two people, and topics can be found in your own research, the current literature, or other sources. The project will be assigned in late February, and a project proposal will be due in mid-March (before Spring Break). Further details will be provided later in the semester.

Teamwork on Homework and Project Assignments: I encourage teamwork and collaboration to increase your learning and to improve your interpersonal skills. The project is explicitly designed to allow you practice interpersonal as well as technical skills. I expect each member of the team to contribute significantly to the group project. It is the responsibility of the team to resolve internal conflicts and balance the workload between team members. At the end of the project, each member of the team will evaluate the contribution of their peers, and these peer evaluations will be used to calculate the final group and individual grades.

Collaboration on Homework: Homework is provided as practice of your individual problem solving skills, and exam problems will test skills developed through the homework. Therefore, you are expected to attempt to solve the problems individually, and each student must submit his or her own work.

However, the test of whether one truly understands something is the ability to explain it to others. Therefore, you may consult with your colleagues if you have difficulty with a problem, and you may also act as a consultant. For example, legitimate consulting questions might be, “How do you select all the lysines in PyMol?” or “Is this the correct syntax for the loop mover?” or “Do you know what this error message means?” However, copying of another person’s homework (“I don’t understand problem 3, can I see yours?”) or giving a colleague a copy of your results (or computer code) are strictly

forbidden. Sharing code for homework assignments before their due date is strictly forbidden; for this reason, do not invite others to share code repositories (i.e. GitHub or similar services).

Violations will be addressed in accordance to the university's policies on academic ethics. If you consult another member of the class on a particular homework assignment, please put the name of that person (or persons) under your name at the top of your homework. The number of consultants will not alter your homework grade.

Accommodations: Inform the professor and TA as soon as possible about any travel, disability, or extenuating circumstances that would affect submission of assignments or exams. We will make our best attempt to accommodate student needs.

Ethics: Cheating is wrong. Cheating hurts our community by undermining academic integrity, creating mistrust, and fostering unfair competition. The university will punish cheaters with failure on an assignment, failure in a course, permanent transcript notation, suspension, and/or expulsion. Offenses may be reported to medical, law or other professional or graduate schools when a cheater applies.

Violations can include cheating on exams, plagiarism, reuse of assignments without permission, improper use of the Internet and electronic devices, unauthorized collaboration, alteration of graded assignments, forgery and falsification, lying, facilitating academic dishonesty, and unfair competition.

Copying of any online material is strictly forbidden. If you find any online solutions to assigned homework problems, you are required to report these websites to the course instructor and refrain from using such material. Beware of online "tutoring" sites (e.g. Chegg) that often pass plagiarized material to their customers. You are solely responsible for any material you submit for class.

Ignorance of these rules is not an excuse.

You may collaborate with other students in this course, but only on homework, the literature review and the lab as detailed above. If you have questions about this policy, please ask the instructor.

On every exam, you will sign the following pledge: "I agree to complete this exam without unauthorized assistance from any person, materials or device. [Signed and dated]"

For more information, see the guide on "Academic Ethics for Undergraduates" and the Ethics Board web site (<http://ethics.jhu.edu>).

Video Lectures: Lectures and mini-lectures during workshops were recorded for this course in the Fall of 2014. These lectures will be similar to those delivered in class, but some content may differ, especially later in the course. Please use these materials to review concepts as needed.

Topics: Tentative material to be covered:

Introduction:

1. Why do we want to engineer proteins?
2. Protein structure and geometry
3. Visualization / Interactive Molecular Graphics (PyMol)

Fundamentals:

4. Molecular energies and forces
5. Protein structural modeling software (PyRosetta)
6. Ab-initio folding; Monte Carlo
7. Refinement of protein structures; Monte Carlo minimization
8. Side-chain prediction from rotamer libraries

Applications:

9. Computational design of novel folds
10. Computational design for function
11. Docking: protein-protein
12. Loop modeling
13. Graph theoretic approaches to protein structure (fold trees)
14. Symmetry

Special topics (as time permits):

1. Post-translational modifications
2. Glycoproteins
3. Membrane proteins
4. RNA structure prediction
5. Protein interface design
6. Enzyme design
7. Homology modeling
8. Docking: small molecules
9. Materials design

Special Dates

MW	3:00-4:15	Class
M	Jan 30	First day of class
F	Feb 3	Python Primer
MW	Mar 20-24	No class—Spring Break
W	Mar 15	Exam I
W	May 3	Last class; Projects due
Sat	May 13	Exam II (2-5 PM)