CSE 40531/60531

Computational Biophysics and Systems Biology

Spring Semester 2006
Faculty-in-Charge: Jesús A. Izaguirre
Fitzpatrick Hall of Engineering 356A
T H 9:30 – 10:45 AM

This course studies the use of computer modeling and simulation of proteins (biophysics) and protein networks (systems biology). Three applications serve as examples to introduce computational methods: protein folding, docking of ligands to proteins (including protein-protein docking), and the study of protein-protein interaction networks. The emphasis is on methods that are truly useful to the study of biological molecules. There is no long baggage of methods of historical importance that are not applicable to biomolecules. From the engineering perspective, both mathematical and computational issues are considered. An introduction to the biological issues is also provided. The course is structured in modules and tutorials. Each module introduces a computational technique: students apply the method under study to a biologically relevant example, and then the theory of the method is presented. Tutorials go along with the modules and provide background on the computational and mathematical background needed to successfully complete each module. Finally, there are assignments and a final project and paper to put it all together. The final project will ideally include science students with a system relevant to their research and computational or engineering students who can contribute algorithms and software. Software especially developed for this course as well as existing tools will be used for the projects, modules, and tutorials. The course will introduce students to the scripting language Python. There will also be guest lectures by different experts in the field.

Textbook:

Tamar Schlick, *Molecular Modeling and Simulation: An Interdisciplinary Guide,* Springer-Verlag, 2002. ISBN 0-387-95404-X
Papers and Tutorials

Course Goals:
At the end of the course, the student should be able to: (1) Apply simulations to the study of problems such as protein folding, docking, the prediction of protein-protein interaction networks, and the automatic annotation of protein function; (2) Understand the best approaches to do equilibration, sampling, free energy computation, long time dynamics, or kinetics of biological molecules; (3) Understand deterministic approaches to solving some of these problems (e.g., molecular dynamics); (4) Understand stochastic approaches (e.g., Markov Chain Monte Carlo methods); (5) Understand combinatorial approaches for solving biological network problems (e.g., graph algorithms applied in biology); (6) Write scripts in Python to perform analysis of biomolecular simulations; (7) Use databases of molecular dynamics trajectories of simulations (e.g., BioSimGrid) and

visualization tools (e.g., Python Molecular Viewer) to perform analysis of biomolecular simulations; (8) Use Condor and Chirp to perform distributed simulations in the computational and storage grid; (9) Undertake some research projects or take more advanced classes in computational biology and bioinformatics.

Prerequisites: Familiarity with a modern programming language is desirable, as well as basic knowledge of probability, linear algebra, and differential equations.

Weekly lecture schedule:

I. Introduction to computational biophysics and systems biology:
- (1) Protein folding and docking
- (2) Biological networks: protein interaction networks

II. Equilibration:
- (3) Molecular dynamics ensembles
- (4) Langevin dynamics vs. Nosé-like thermostats

III. Conformational Sampling:
- (5) Multicanonical approaches: parallel tempering / replica exchange
- (6) Capturing rare events: Transition path sampling
- (7) Free energy computation: Thermodynamic integration

IV. Long time dynamics:
- (8) Multiple time stepping integrators: r-RESPA and alternatives
- (9) Elastic network methods: Gaussian networks and anisotropic networks
- (10) Comparison to experiments: NMR and X-ray crystallography
- (11) Kinetics of protein folding: Folding @ Home

V. Protein interaction networks:
- (12) Inferring protein interaction networks
- (13) Inferring protein function from networks
- (14) Validating protein interactions experimentally and computationally

Tutorials schedule:

I. Crash course on Python
II. The mathematics of molecular dynamics and related applications
III. Using the Molecular Dynamics Language (MDL) interface and the Python Molecular Viewer (PMV)
IV. Using Condor and Chirp to manage distributed simulations and storage
V. Using Evolutionary Trace to construct phylogeny trees
VI. Using the protein-protein interaction portal

Homework: The homework consists of 5 assignments. Some of the assignments require programming. Students will be able to use and extend software developed in Python to provide an easy to use interface to biomolecular simulation and visualization.

Laboratory usage: All students will obtain accounts in high performance clusters at Notre Dame and elsewhere, as well as being able to use the Condor and Chirp pools for distributed computing in the department of computer science and engineering.