CSCI 2951-G: Computational Protein Folding Fall 2012

Lecture 1 — 09/06/2012

Prof. Paul Valiant *Scribe: Alexandra Papoutsaki*

1 Overview

In this lecture we gave an overview of the general direction that this course will follow. We started with the responsibilities of the students, the textbook that we are going to use, the grading, and the collaboration policy. We then moved to an introduction of the main problem that this class is going to examine: ”Protein Folding”, and the different fields that are involved in it. We studied the principles of Biology that participate in the production of proteins and we briefly analysed a few factors that contribute to the final structure of a protein. During the whole lecture we got engaged with different simulations of proteins through Matlab.

2 General Information on the course

Textbook: ”Introduction to Proteins - Structure, Function, and Motion”, A. Kessel & N. Ben-Tal. The textbook is available from the bookstore, the library, or (soon) from my admin assistant Saara Moskowitz, in CIT 546.

Responsibilities: Class participation, homeworks, engagement in discussions either in class or through emails, final projects, and producing scribe notes like this one.

Collaboration Policy: Guidelines may vary for individual assignments, but the ground rules are that you should be open about your sources, so that we as a class can move forward. If you work on an assignment as a group, say so; if you bring interesting ideas from the internet into class discussion, the class would love to hear them, but name your source.

Red-cyan glasses: Be sure to get your own pair (from the professor), they are going to be useful when examining stereoscopic images.

3D mouse: We can borrow them from Saara Moskowitz, CIT 546.

3 Protein Folding - An interdisciplinary field

When studying the problem of “Protein Folding” we integrate four different scientific fields: Biology, Computer Science, Mathematics, and Physics.

Further, there are at least three aspects of computer science:

1. *Design of algorithms* – This is the “theory” side, and ultimately this course will be about the challenge of finding a new algorithm for protein folding.

2. *Software/Systems* – Using a computer as an experimental tool is invaluable at providing insight when designing these algorithms.

3. *High Performance Computing* – Since protein folding is currently beyond all our computational resources, highly parallel computing systems are a natural way to explore the limits.

3.1 Biology Background

The textbook has a lot more detail, and parts of it will be assigned as reading later. To get started with proteins, though, we will review some of the basics about what proteins are, and why they are important.

The playing field on which the action takes place is a *cell.* Typical cells in our bodies are roughly 10^{-5} meters across ($10\mu m$), slightly thinner than a human hair. There are many structures in a cell which biology textbooks will depict for you, though none of them are relevant for us now.

One of the main triumphs of computational biology is sequencing *DNA*, often referred to as "the book of life". DNA contains the genetic information that is responsible (through proteins) for the development and functionality of the organism. Its double helix consists of four nucleobases (adenine, thymine, cytosine, guanine) whose sequence encodes information, as a base-4 code [1]. Each nucleobase consists of just a couple dozen atoms. The scale of an atom (and of bonds between atoms) is roughly one *angstrom*, defined as $10^{-10}m$ and often denoted "A". Thus to a rough approximation, you can think of each nucleobase as occupying a space that is about 10A ($= 10^{-9}m$) big.

The human genome has approximately 3 billion base pairs of DNA arranged into 23 pairs of chromosomes; if all this DNA were somehow stretched out, it would stretch a few meters ($\approx 10^{-9}m$ per nucleosome times a few billion). Crammed inside a cell just 10^{-5} meters across, things are *very* crowded, if you can visualize that.

DNA itself is not very functional. Its main purpose is to encode (and control the production of) proteins. The *universal genetic code* describes how triples of nucleobases (4^3 possibilities) can be decoded into a choice of 20 amino acids, and sequences of amino acids form proteins. DNA has roughly 20,000 segments that code for proteins, so, modulo some biological complications which we won't go into now, you can think of humans as having roughly 20,000 proteins, that collectively enact all the biological process that run our body. Each protein is typically a chain of between 50 and 300 amino acids.

You have to remember that molecules in general (and proteins in our case) are active, moving around in the cell, and reactive, willing to interact with their environment. You can think of them as machines wandering in the cell and accomplishing tasks. The individual thrashing around of atoms happens on around the femtosecond ($10^{-15}s$) timescale, so even relatively fast biological processes, taking thousandths to millionths of a second, are only the bulk result of a *lot* of thrashing around at the atomic level. Biological molecules look "purposeful" at the millisecond to microsecond level, but at the femtosecond level, a lot of the motion seems useless. (Is this a challenge, or an opportunity, from the algorithmic perspective?)

3.2 The Problem

The first human genome was sequenced about 10 years ago, and the universal genetic code has been known for a lot longer. By combining these, we can read off the sequence of amino acids that form each protein. (There are some complications here, where some proteins arise by more complicated processes, but that is beyond the scope of the class.) The huge challenge is, knowing what these proteins are made of, can we we figure out what they do? The most basic part of this is that, within about a second of being made, most proteins will fold into a compact and stable form; knowing this final folded form would let us understand how the protein behaves. Biology labs have been making slow and steady progress at discovering the folded forms of various proteins, but it is an enormously expensive process that often requires new methodological discoveries for each new protein. Meanwhile, computers have been relatively useless at this task of discovering new protein structures. The challenge is to predict the final form of a few thousand atoms after being left alone for about a second. In this course we will consider potential algorithmic approaches to this.

A different perspective on what happens during this second is provided by statistical physics: the protein will generally seek out the global minimum of its (potential) energy function. So instead of thinking of this as a simulation problem, you could think of this as a minimization problem. The fact that all these proteins evolved is a big bonus here – finding the global minimum in a complicated landscape is in general a very hard problem, but it is evolutionarily advantageous for proteins to be "easy to fold", so that the proteins will actually function robustly in the body. Thus evolution provides us a guarantee that, at least in some sense, all the proteins we care about will be "easy" to fold.

3.3 The Model

The potential energy function that is used to simulate proteins (see [2]) is:

$$E = \sum_{\text{bonds}} K_r(r - r_{\text{eq}})^2 + \sum_{\text{angles}} K_\theta(\theta - \theta_{\text{eq}})^2 + \sum_{\text{dihedrals}} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\epsilon R_{ij}} \right]$$

The notation is a bit opaque, but intuitive once you realize what we are trying to compute: each bond between atoms is treated as a spring, so if r is the length of the bond in question, and r_{eq} is the equilibrium length of the bond, and K_r is the spring constant of the bond, then the first term just treats each bond as a spring. The second term does the same with angles between three adjacent (bonded) atoms, with θ being the angle, θ_{eq} being the equilibrium angle, and K_θ being the associated spring constant. The next term computes a function of the dihedral angles in the protein, represented here by the letter ϕ; a dihedral angle is formed by a sequence of 4 atoms, and is defined as the angle between the planes formed by the first 3 atoms, and the last 3 atoms respectively. Each dihedral angle might have its energy represented as the sum of several terms here, in the manner of a Fourier expansion. The final sum, over all pairs of atoms, contains two different types of terms. The first pair of terms is the *Lennard-Jones* potential, in general $\frac{1}{r^{12}} - \frac{1}{r^6}$, which serves primarily as a very strong short range repulsive force to prevent atoms from overlapping. The final term is Coulomb's law, describing the electrostatic force between a pair of particles with charges q_i and q_j respectively.

Each of the parameters in these equations is specified by data files, which have been slowly honed as the result of at least 30 years of simulating proteins along these lines. The parameters do not necessarily correspond to physical quantities, but are the parameters that have been found to yield “reasonable” simulations.

To compute the forces on atoms, given such an energy function, take its *gradient* (the multivariate version of a derivative).

One thing we have not discussed yet, is that proteins are surrounded by (mostly) water. There are several approaches to simulating water; the easiest is to just surround the protein with thousands of water molecules, and simulate each of the interactions. Water is crucial to protein folding, as you would not expect proteins to function in the vacuum of empty space. Details of this will be covered in later lectures.

References

[1] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell.* 2002.

[2] Wendy Cornell, Piotr Cieplak, Christopher Bayly, Ian Gould, Kenneth Merz, David Ferguson, David Spellmeyer, Thomas Fox, James Caldwell, and Peter Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.*, 1995.