CSCI 2951-G: Computational Protein Folding Fall 2012

Class 6 — 09/24/2012

Prof. Paul Valiant

Announcements:

The homework that was just assigned will be postponed by a week, since people have been itching for details about what's actually going on with the `proteins` simulation. Instead, due one week from today, the homework will be "implement physics yourself", as in, the input to your function will be something like the `p2` variable that the `proteins` code has been using to describe physics, specifying force parameters for bonds, angles, dihedrals, and then each atom's charge, and the Lennard-Jones radius and scaling factor. From this you will have to compute the force on each atom. This should take roughly two kilobytes of code, as in, roughly 50 lines of code. Coding this yourself should make the details a lot clearer. In particular, a student has been curious, "how are these simulations possibly able to run fast on my laptop?" And thus part of the assignment will be counting the number of arithmetic operations your code takes, and how fast you might expect this to run on a gigahertz machine in a fast language instead of Matlab.

Concepts from the readings:

Because this is an interdisciplinary course, the readings will require you to look up terminology on your own since mostly the readings are written within the context of a certain field, and not for an audience with interdisciplinary backgrounds like this class. Here are some concepts which were not necessarily in the reading (Chapter 1), but which are necessary to make sense of the reading.

Enzyme: This was a word that was used many times in the reading, and if you understood the reading, you should understand this word. Enzyme is a phenomenological word – back in the 19th century, before people had any idea what was going on biochemically, they just started pointing to molecules and saying "this is a thing that makes reactions go faster" – "this is a thing which, if it weren't there, your body would just grind to a halt". They called these things enzymes. Eventually once biotechnology got up to speed, people realized that essentially all of what people had been calling enzymes were actually proteins. (There are exceptions, like in all things biological, but....) So in this course, "enzyme" means "protein".

Covalent and non-covalent: The textbook distinguishes between "covalent" and "non-covalent" interactions. Eventually we will start reading research papers and it will be useful to be fluent with this kind of technical terminology. What's the difference? One student puts forward that "covalent ones are the strong ones and non-covalent ones are the weaker ones". This answer refers to bonds – we can talk about many types of bonds between atoms, and "covalent ones are the strong ones". Specifically, "covalent" describes all the bonds that we draw when we draw a molecule (the "tubes" that we draw between atoms), the graph theory structure that doesn't change. And, further, "covalent interactions" describes every interaction related to the graph-theoretic structure of the molecule. In the energy equation

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

the covalent interactions are the first three terms, dealing with bond length, bond angles, and bond dihedral angles. The final term, dealing with Lennard-Jones and electrostatic interactions describes the non-covalent interactions, interactions that occur between *every* pair of atoms, regardless of bond structure.

Hydrophobic/Hydrophilic: Again, these words are vitally important to proteins,but not properly defined by the textbook. The words are from the Greek: hydro = water; phobic = fear, like "-phobia"; philic =

“love”, like “-philia”. So, literally, water-fearing, and water-loving. What do these words mean in the language of this course? Again, the textbook kept using these words. It means something very simple in our language, actually, it’s sort of like “enzyme” before, a phenomonological term to describe a complicated class of behavior that turns out to boil down to something very simple in modern language. We’ll talk about water more explicitly in the next lecture, don’t worry. One student says that the hydrophobic/hydrophilic distinction is whether or not the molecule will mix with water. But this is still a phenomonological definition. How can you tell, without running a simulation or asking a biology lab? The student ventures that it has something to do with “polar” versus “nonpolar” – which we will define now, before returning to the meaning hydrophobic and hydrophilic.

Polar/non-polar A student says that “polar” refers to a separation of charges. With this in mind, we can draw a taxonomy of charges for a region of a protein:

- 1: The region has no charge anywhere;
- 2a: The region has positive charges;
- 2b: The region has negative charges;
- 3: The region is *polar*, meaning it has both positive and negative charges, in different places.

Case 2 we in general can just call “charged.” This taxonomy of charges is very much related to the hydrophobic/hydrophilic distinction. A student says that water is polar. Water is very polar – it has an oxygen and two hydrogens bonded to it, with an angle of around 105 degrees between them, and the oxygen is negatively charged, while the hydrogens are positively charged. The question we must ask is, of the taxonomy of charges, which cases will a polar (water) molecule be attracted to? The answer is polar and charged (Cases 2 and 3): if you have, say, a positively charged atom, then the oxygen side of water will be attracted to it, and the two hydrogen atoms will trail behind as the oxygen loosely attaches itself to our positive atom; Conversely, if you have a negatively charged atom, then one of the two hydrogen atoms will end up stuck to it, with the other atoms trailing behind; if you have a polar region, then a water molecule can end up stuck to either the positive or negative atoms, because it is attracted to both.

So, with this in mind, we have answered our problem; this is what hydrophilic is: water is attracted to something if it has a charge anywhere on it (a positive, negative, or polar region). Conversely, if it has no charge on it anywhere, then we call it hydrophobic because water will not be attracted to it at all. A student points out that, in fact, because water is polar, water is also strongly attracted to itself, and thus given a choice between being next to an uncharged region, versus being surrounded by other water molecules, a water molecule would prefer to be around other water,

The net result of this is that the regions of the protein that are not charged at all are going to end up clumped at the center of the protein away from any water, while the regions that are positive, negative, or polarized are going to end up on the surface of the protein. That is the distinction between hydrophobic and hydrophilic.

One unusual effect of this is that inert nowhere-charged-regions effectively attract each other, because the bulk effect of water not wanting to be near them will end up pushing them together. This has already been mentioned in Chapter 1, that hydrophobic regions attract each other. And at first glance this is a strange statement to make, but you just need to untangle it: hydrophobic means nowhere-charged; why do nowhere-charged regions attract each other? Because water is so bored with them that it forces them together by exclusion.

A student asked a question about Homework 3, and more generally, how one can robustly estimate whether a protein has good (low) energy, or not, given that proteins get very big with many interactions? The answer is that all of the strong interactions in proteins are local, so if you just want a rough notion of energy, rough local estimates will serve you quite well. Also, proteins keep reusing the same familiar parts – proteins do not have much variety, in many senses of the word. This is why we spend time studying particular parts of proteins, because there are so few basic parts. Leveraging this will be quite important.

Masses of particles: What is the effect of *changing the masses of the atoms* in a molecular dynamics simulation – as in, `p2.weights(27)=100`, making the weight of the 27th atom equal to 100. First, how does mass enter into physics simulations? Only via $F = ma$; the forces do not depend on masses at all (remember that potential energy is a function just of position $PE(p)$, and kinetic energy is just a function of velocities (and the masses), $KE_m(v)$); however acceleration is computed as F/m. So if we change the masses, the accelerations, velocities, and paths of the particles will change.

However, in light of what we covered in classes 3+4, what we really have to be asking about a system is, what is its Boltzmann distribution – because this describes the folded state, what happens to the system when you leave it for a long enough time. Boltzmann's distribution is $e^{-E/(k_B T)}$, specifically $e^{-(PE(p)+KE_m(v))/(k_B T)}$. We can split this apart as $Pr_m[p, v] \sim e^{-PE(p)/(k_B T)} e^{-KE_m(v)/(k_B T)}$. Note that since this is a product of two terms, one depending only on p, the other depending only on v, the distributions of p and v are *independent*, meaning that $Pr[p] \sim e^{-PE(p)/(k_B T)}$ and $Pr_m[v] \sim e^{-KE_m(v)/(k_B T)}$, and the shocking thing here is that the distribution $Pr[p]$, the distribution of the position of the folded state of the protein, does not depend on the masses (but the distribution of velocity will, since if we make a certain atom lighter, it will move faster).

This is a very unusual fact: the masses of the atoms do not affect the folded state of the protein. One interesting way in which we can use this observation to speed up protein folding (slightly) is to note that in some sense the "clock" of physics simulation is determined by the thrashings of hydrogen atoms. Hydrogens are 1/12th the weight of the next lightest atom, meaning that they oscillate at a shorter time period than any of the other atoms; recall from the second class that we cannot use a timestep much above around a fifth of the shortest period of oscillation, so the hydrogens act as a global barrier to the simulation clock. However, knowing that masses do not affect the folded state of a protein, we can now run our simulations with "heavy hydrogens", which will now oscillate slower, and let us use bigger timesteps and may speed up the simulation by a factor of $\sqrt{12}$ – though in practice the number is closer to 2.

This perspective that the velocities of the particles have a completely separate role in the Boltzmann distribution from the positions of the particles is also related to why we can hope to have an implicit water model – it is conceivable that we can model the average effect of water on the Boltzmann distribution by only keeping track of the position of water, without paying attention to its velocity or side effects like the viscosity of water.

Think about this paradox: changing masses really changes how physics looks – if you're pushing on a hydrogen atom and suddenly you're pushing on a boulder, this changes physics drastically. But paradoxically, it does not change the ultimate steady-state distribution.

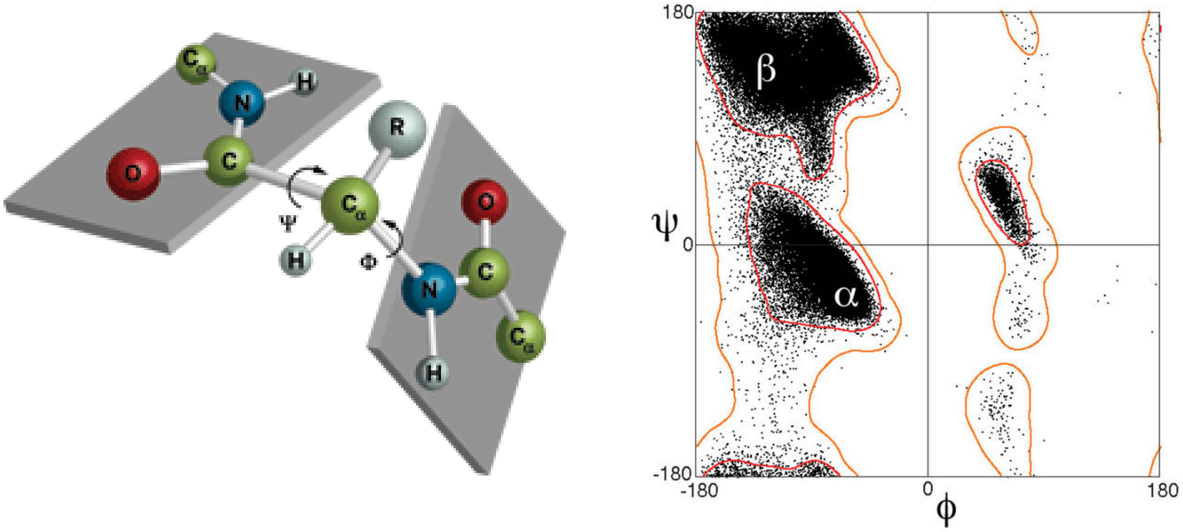
Ramachandran plots:

Most of the excitement of proteins happens in the backbone – when people draw pictures of proteins, they generally don't draw atoms hanging off the backbone, they just draw the backbone.

We will focus on the *dihedral angles* of the backbone, because this turns out to essentially describe what is going on in the whole protein. See Figure 1, where "R" denotes the side-chain of the amino acid, whether it is 1 atom or 20.

Each amino acid has 3 backbone atoms, named N, C_α, and C (nitrogen, the carbon to which the side-chain is attached, and another carbon). Chains of amino acids have these three backbone atoms repeating in series, $... - N - C_\alpha - C - N - C_\alpha - C - ....$ There are thus three different sorts of dihedral angles formed along the backbone, those centered around the bonds $N - C_\alpha$, $C_\alpha - C$, and $C - N$, which are denoted phi, psi, and omega respectively (ϕ, ψ, ω). In Figure 1, if you think of the locations of the central C_α atom along with everything immediately attached to it as being fixed, then ϕ describes how much to rotate everything before the nitrogen atom (the gray rectangle on the right) about the bond connecting the nitrogen to the center C_α; ψ describes how much to rotate everything after the carbon (C) about the bond connecting the carbon to the center C_α. (The backbone in Figure 1 goes right-to-left.)

A Ramachandran plot (see Figure 2) is a scatter-plot among folded proteins of the pairs of phi-psi angles situated around a center C_α. The densely populated areas represent popular combinations of phi-psi angles.



(a) The backbone dihedral angles ϕ and ψ

(b) Ramachandran plot for amino acids except proline and glycine

Figure 2, from the Wikipedia entry for Ramachandran plots looks complicated (and indeed it is, representing billions of dollars of meticulously collected data about protein structure), but most of it is explainable in fairly simple terms from local properties of the energy function from our force field. Remember, in some sense the density of points in any region should be related to the Boltzmann distribution, $e^{-E/(k_BT)}$, so the claim is that the populated regions of the Ramachandran plot correspond to slightly lower energy than the unpopulated regions. (Since k_BT is roughly 0.6 in our units of kcal/mol, energy difference as small as 2 to 5 kcal/mol can wipe a region off the map.)

The two main populated regions of the Ramachandran plot are labeled α and β, which correspond to the structural elements of *alpha helices* and *beta sheets*, which we will read about in chapter 2, and which are stable structures for a variety of reasons involving long-range interactions of proteins. The emphasis here is that in some sense these are the only allowed configurations for very simple local reasons. We will see more of this next lecture, but the biggest cause for the forbidden regions on this plot are that they would rotate certain atoms to be too close to each other: in some cases it is backbone atoms colliding with other backbone atoms; in other cases the side chain is involved – and thus amino acids with unusual side chains will have different Ramachandran plots. We will see this for proline and glycine in the next lecture.