PROFESSOR: Welcome back, everyone. I hope you had a good break. Hopefully you also remember a little bit about what we did last time.

So if you'll recall, last time we did an introduction to protein structure. We talked a little bit about some of the issues in predicting protein structure. Now we're going to go into that in more detail.

And last time, we'd broken down the structure prediction problem into a couple of sub-problems. So there was a problem of secondary structure prediction, which we discussed a little bit last time. And remember that the early algorithms developed in the '70s get about 60% accuracy, and decades of research has only marginally improved that. But we're going to see that some of the work on the main structure recognition and predicting novel three-dimensional structures has really advanced very dramatically in the last few years.

Now, the other thing I hope you'll recall is that we had this dichotomy between two approaches to the energetics of protein structure. We had the physicist's approach and we the statistician's approach, right? Now, what were some of the key differences between these two approaches?

Anyone want to volunteer a difference between the statistical approach to parametrizing the energy of a structure? So we're trying to come up with an equation that will convert coordinates into energy, right? And what were some of the differences between the physics approach and the statistical approach? Any volunteers? Yes.

AUDIENCE: I think the statistical approach didn't change the phi and psi angles, right? It just changed other variables.

PROFESSOR: So you're close. Right. So the statistical-- or maybe you said the right thing, actually. So the statistical approach keeps a lot of the pieces the protein rigid, whereas the physics approach allows all the atoms to move independently. So one of the key differences, then, is that in the physics approach, two atoms that are bonded to each other still move apart based on a spring function. It's a very stiff spring, but the atoms move independently.

In the statistical approach, we just fix the distance between them. Similarly for a tetrahedrally coordinated atom, in the physics approach those angles can deform. In the statistical approach, they're fixed. Right? So in the statistical approach, we have more or less fixed geometry. In the physics approach, every atom moves independently.

Anyone else remember another key difference? Where do the energy functions come from? Volunteers? All right.

So in the physics approach, they're all derived as much as possible from physical principles, you might imagine. Whereas in the statistical approach, we're trying to recreate what we see in nature, even if we don't have a good physical grounding for it.

So this is most dramatic in trying to predict the solvation free energies. Right? How much does it cost you if you put a hydrophobic atom into a polar environment? Right? So in the physics approach, you actually have to have water molecules. They have to interact with matter. That turns out to be really, really hard to do.

In the statistical approach, we come up with an approximation. How much solvent accessible surface area is there on the polar atom when it's free? When it's in the protein structure? And then we scale the transfer energies by that amount.

OK, so these are then the main differences. Gotta be careful here. So we've got fixed geometry this the statistical approach. We often use discrete rotamers. Remember? The side-chain angles, in principle, can rotate freely. But there were only a few confirmations are typically observed, so we often restrict ourselves to the most commonly observed combinations of the psi angles.

And then we have the statistical potential that depends on the frequency at which we observe things in the database. And that could be the frequency at which we observe particular atoms at precise distances. It could be the fraction of time that something's solvent accessible versus not.

And the other thing that we talked about a little bit last time was this thought problem. If I have a protein sequence and I have two potential structures, how could I use these potential energies-- whether they're derived from the physics approach or from the statistical approach-- how could I use these potential energies to decide which of the two structures is correct?

So one possibility is that I have two structures. One of them is truly the structure and the other is not. Right? Your fiendish lab mate knows the structure but refuses to tell you. So in that case, what would I do? I know that one of these structures is correct. I don't know which one. How could I use the potential energy function to decide which one's correct? What's going to be true of the correct structure?

AUDIENCE: Minimal energy.

PROFESSOR: It's going to have lower energy. So is that sufficient? No. Right? There's a subtlety we have to face here.

So if I just plug my protein sequence onto one of these two structures and compute the free energy, there's no guarantee that the correct one will have lower free energy. Why? What decision do I have to make when I put a protein sequence onto a backbone structure?

Yes.

AUDIENCE: How to orient the side chain.

PROFESSOR: Exactly. I need to decide how to orient the side chains. If I orient the side chains wrong, then I'll have side chains literally overlapping with each other. That'll have incredibly high energy, right? So there's no guarantee that simply having the right structure will give you the minimal free energy until you correctly place all the side chains.

OK, but that's the simple case. Now, that's in the case where you've got this fiendish friend who knows the correct structure. But of course, in the general domain recognition problem, we don't know the correct structure. We have homologues. So we have some sequence, and we believe that it's either homologous to Protein A or to Protein B, and I want to decide which one's correct. So in both cases, the structure's wrong. It's this question of how wrong it is, right?

So now the problem actually becomes harder, because not only do I need to get the right side chain confirmations, but I need to get the right backbone confirmation. It's going to close to one of these structures, perhaps, but it's never going to be identical.

So both of these situations are examples where have to do some kind of refinement of an initial starting structure. And what we're going to talk about for the next part of the lecture are alternative strategies for refining a partially correct structure.

And we're going to look at three strategies. The simplest one is called energy minimization. Then we're going to look at molecular dynamics and simulated annealing.

So energy minimization starts with this principle that we talked about last time I remember that came up here, that a stable structure has to be a minimum of free energy. Right? Because if it's not, then there are forces acting on the atoms and that are going to drive it away from that structure to some other structure.

Now, the fact that it is a minimum of free energy does not guarantee that is the minimum of free energy. So it's possible that there are other energetic minima. Right? The protein structure, if it's stable, is at the very least a local energetic minimum. It may also be the global free energy minimum. We just don't know the answer to that.

Now, this was a big area of debate in the early days of the protein structure field, whether proteins could fold spontaneously. If they did, then it meant that they were at least apparently global free energy minima. Chris Anfinsen actually won the Nobel Prize for demonstrating that some proteins could fold independently outside of the cell. So at least some proteins had all the structural information implicit in their sequence, right? And that seems to imply that there are global free energy minimum.

But there are other proteins, we now know, where the most commonly observed structure has only a local free energy minimum. And it's got very high energetic barriers that prevent it from actually getting to the global free energy minimum. But regardless of the case, if we have an initial starting structure, we could try to find the nearest local free energy minimum, and perhaps that is the stable structure.

So in our context, we were talking about packing the side chains on the surface of the protein that we believe might be the right structure. So imagine that this is the true structure and we've got the side chain, and it's making the dashed green lines represent hydrogen bonds. It's making a series of hydrogen bonds from this nitrogen and this oxygen to pieces of the rest of the protein.

Now, we get the crude backbone structure. We pop in our side chains. We don't necessarily-- in fact, we almost never-- will choose randomly to have the right confirmation to pick up all these hydrogen bonds. So we'll start off with some structure that looks like this, where it's rotated, so that instead of seeing both the nitrogen and the oxygen, you can only see the profile.

And so the question is whether we can get from one to by following the energetic minima. So that's the question. How would we go about doing this?

Well, we have this function that tells us the potential energy for every XYZ coordinate of the atom. That's what we talked about last time, and you can go back and look at your notes for those two approaches. So how could we minimize this free energy minimum? Well, it's no different from other functions that we want to minimize, right? We take the first derivative. We look for places where the first derivative is zero.

The one difference is that we can't write out analytically what this function looks like and choose directions and locations in space that are the minima. So we're going to have to take an approach that has a series of perturbations to a structure that try to improve the free energy systematically.

The simplest understanding is this gradient descent approach, which says that I have some initial coordinates that I choose and I take a step in the direction of the first derivative of the function. So what does that look like?

So here are two possibilities. I've got this function. If I start off at x equals 2, this minus some epsilon, some small value times the first derivative, is going to point me to the left. And I'm going to take steps to the left until this function, f prime, the first derivative, is zero. Then I'm going to stop moving. So I move from my initial coordinate a little bit each time to the left until I get to the minimum. And similarly, if I start off on the right, I'll move a little bit further to the right each time until the first derivative is zero.

So that looks pretty good. It can take a lot of steps, though. And it's not actually guaranteed to have great convergence properties. Because of the number of steps you might have to take, it might take quite a long time. So that's the first derivative, in a simple one-dimensional case. We're dealing with a multi-dimensional vector, so instead of doing the first derivative we use the gradient, which is a set of partial first derivatives.

And I think one thing that's useful to point out here is that, of course, the force is negative of the gradient of the potential energy. So when we do gradient descent, you can think of it from a physical perspective as always moving in the direction of the force. So I have some structure. It's not the true native structure, but I take incremental steps in the direction of the force and I move towards some local minima.

And we've done this in the case of a continuous energy, but you can actually also do this for discrete ones.

Now, the critical point was that you're not guaranteed to get to the correct energetic structure. So in the case that I showed you before where we had the side chain side-on, if you actually do the minimization there, you actually end up with the side chain rotated 180 degrees where it's supposed to be. So it eliminates all the steric clashes, but it doesn't actually pick up all the hydrogen bonds. So this is an example of a local energetic minima that's not the global energetic minima.

Any questions on that? Yes.

AUDIENCE: Where do all these n-dimensional equations come from?

PROFESSOR: Where do what come from?

AUDIENCE: The n-dimensional equations.

PROFESSOR: So these are the equations for the energy in terms of every single atom in the protein if you're allowing the atoms to move, or in terms of every rotatable bond, if you're allowing only bonds to rotate.

So the question was, where do the multi-dimensional equations come from. Other questions? OK.

All right, so that's the simplest approach. Literally minimize the energy. But we said it has this problem that it's not guaranteed to find the global free energy minimum.

Another approach is molecular dynamics. So this actually attempts to simulate what's going on in a protein structure in vitro, by simulating the force in every atom and the velocity. Previously, there was no measure of velocity. Right? All the atoms were static. We looked at what the gradient of the energy was and we move by some arbitrary step function in the direction of the force.

Now we're actually going to have velocities associated with all the atoms. They're going to be moving around in space. And we'll have the coordinate at any time t is going to be determined by the coordinates of the previous time, t of i minus 1 plus a velocity times the time step. And the velocities are going to be determined by the forces, which are determined by the gradient of the potential energy. Right?

So we start off, always, with that potential energy function, which is either from the physics approach or the statistical approach. That gives us velocities, eventually giving us the coordinates.

So we start off with the protein. There are some serious questions of how you equilibrate the atoms. So you start off with a completely static structure. You want to apply forces to it. There are some subtleties as to how you go about doing that, but then you actually end up simulating the motion of all the atoms.

And just give you a sense of what that looks like, I'll show you a quick movie. So this is the simulation of the folding of a protein structure. And the backbone is mostly highlighted. Most of the side chains are not being shown. Actually, in bold, but you can see the stick figures. And slowly it's accumulating its three-dimensional structure.

[VIDEO PLAYBACK]

[LAUGHTER]

[END VIDEO PLAYBACK]

PROFESSOR: OK, I think you get the idea here. Oh, it won't let me give up. OK, here we go.

OK, so these are the equations that are governing the motion in an example like that. Now, the advantage of this is we're actually simulating the protein folding. So if we do it correctly, we should always get the right answer. Of course, that's not what happens in reality.

Probably the biggest problem is just computational speed. So these simulations-- even very, very short ones like the one I showed you-- so how long does it take a protein to fold in vitro? A long folding might take a millisecond, and for a very small protein like that it might be orders of magnitude faster. But to actually compute that could take many, many, many days. So a lot of computing resources going into this.

Also, if we want to accurately represent solvation-- the interaction of the protein with water, which is what causes the hydrophobic collapse, as we saw-- then you actually would have to have water in those simulations. And each water molecule adds a lot of degrees of freedom, so that increases the computational cost, as well.

So all of these things determine the radius of convergence. How far away can you be from the true structure and still get there? For very small proteins like this, with a lot of computational resources, you can get from an unfolded protein to the folded state. We'll see some important advances that allow us to get around this, but in most cases we only can do relatively local changes.

So that brings us to our third approach for refining protein structures, which is called simulated annealing. And the inspiration for this name comes from metallurgy and how to get the best atomic structure in a metal. I don't know if any of you have ever done any metalworking. Anyone?

Oh, OK, well one person. That's better than most years. I have not, but I understand that in metallurgy-- and you can correct me if I'm wrong-- that by repeatedly raising and lowering the temperature, you can get better metal structures. Is that reasonably accurate? OK. You can talk to one of your fellow students for more details if you're interested.

So this similar idea is going to be used in this competition approach. We're going to try to find the most probable confirmation of atoms by trying to get out of some local minima by raising the energy of the system and then changing the temperatures, or raising and lowering it according to some heating and cooling schedule to get the atoms into their most probable confirmation, the most stable conformation.

And this goes back to this idea that we started with the local minima. If we're just doing energy minimization, we're not going to be able to get from this minimum to this minimum, because these energetic barriers are in the way. So we need to raise the energy of the system to jump over these energetic barriers before we can get to the global free energy minimum.

But if we just move at very high temperature all the time, we will sample the entire energetic space but it's going to take a long time. We're going to be sampling a lot of confirmations that are low probability, as well. So this approach allows us to balance the need for speed and the need to be at high temperature where we can overcome some of these barriers.

So one thing that I want to stress here is that we've made a physical analogy to this metallurgy process. We're talking about raising the temperature of the system and let the atoms evolve under forces, but it's in no way meant to simulate what's going on in protein folding. So molecular dynamics would try to say, this is what's actually happening to this protein as it folds in water.

Simulated annealing is using high temperature to search over spaces and then low temperature. But these temperatures much, much higher than the protein would ever encounter, so it's not a simulation. It's a search strategy.

OK, so the key to this-- and I'll tell you the full algorithm in a second-- but at various steps in the algorithm we're trying to make decisions about how to move from our current set of coordinates to some alternative set of coordinates. Now, that new set of coordinates we're going to call test state. And we're going to decide whether the new state is more or less probable than the current one. Right?

If it's lower in energy, then what's it going to be? It's going to be more probable, right? And so in this algorithm, we're always going to accept those states that are lower in free energy than our current state.

What happens when the state is higher in free energy than our current state? So it turns out we are going to accept it probabilistically. Sometimes it's going to move up in energy and sometimes not, and that is going to allow us to go over some those energetic barriers and try to get to new energetic states that would not be accessible to purely minimization.

So the form of this is the Boltzmann equation, right? The probability of some test state compared to the probability of a reference state is going to be the ratio of these two Boltzmann equations-- the energy of the test state over the energy of the current state. So it's the e to the minus difference in energy over KT. And we'll come back to where this temperature term comes from in a second.

OK, so here's the full algorithm. We will either iterate for a fixed number of steps or until convergence. We'll see we don't always converge. We have some initial confirmation. Our current confirmation will be state n, and that we can compute as energy from those potential energy functions that we discussed in the last meeting.

We're going to choose a neighboring state at random. So what does neighboring mean? So if I'm defining this in terms of XYZ coordinates, for every atom I've got a set of XYZ coordinates I'm going to change them a few of them by small amount. Right? If I change them all by large amounts, I have a completely different structure. So I'm going to make small perturbations. And if I'm doing this with fixed backbone angles and just rotating the side chains, then what would a neighboring state be?

Any thoughts? What would a neighboring state be? Anyone? Change a few of the side chain angles, right? So we don't want to globally change the structure. We want some continuity between the current state and the next state.

So we're going to chose an adjacent state in that sense, so the state space. And then here are the rules. If the new state has an energy that's lower than the current state, we simply accept the new state. If not, this is where it gets interesting. Then, we accept that higher energy with a probability that's associated with the difference in the energies. So if the difference is very, very large, there's a low probability it'll accept. If the differences are slightly higher, than there's a higher probability that we accept. If we reject it, we just drop back to our current state and we look for a new test state. OK? Any questions on how we do this?

Question, yes.

AUDIENCE: How far away do we search for neighbors?

PROFESSOR: That's the art of this process, so I gave you a straight answer. Different approaches will use different thresholds. Any other questions?

OK, so the key thing I want you realize, then, is there's this distinction between the minimization approach and simulated annealing approach. Minimization can only go from state one to the local free energy minimum, whereas the simulated annealing has the potential to go much further afield, and potentially to get to the global free energy minimum. But it's not guaranteed to find it.

OK, so let's say we start in state one and our neighbor state was state two. So we'd accept that with 100% probability, right? Because it's lower in energy. Then let's say the neighboring state turns out to be state three. that's higher in energy, so there's a probability that we'll accept it, based on the difference between the energy of state two and state three. Similarly from state three to state four, so we might drop back to state two. We might go up. And then we can eventually get over the hump this way with sum probability. It's a sum of each of those steps. OK?

OK, so if this is our function for deciding whether to accept a new state, how does temperature affect our decisions? What happens when the temperature is very, very high, if you look at that equation? So it's minus e to the delta. The difference in the energy over kT. So if t is very, very large, then what happens that exponent?

It approaches zero. So e to the minus zero is going to be approximately 1, right? So at very high temperatures, we almost always take the high energy state. So that's what allows us to climb those energetic hills. If I have a very high temperature in my simulated annealing, then I'm always going over those barriers.

So conversely, what happens, then, when I set the temperature very low? Then there's a very, very low probability of accepting those changes, right? So if I have a very low temperature-- temperature approximately zero-- then I'll never go uphill. Almost never go uphill. So we have a lot of control over how much of the space this algorithm explores by how we set the temperature.

So this is again a little bit of the art simulated annealing-- decide exactly what annealing schedule to use, what temperature program you use. Do you start off high and go literally down? Do you use some other, more complicated function to decide the temperature? We won't go into exactly how to choose these. [INAUDIBLE] you could track some of these things down from the references that are in the notes.

So we have this choice. But the basic idea is, we're going to start at higher temperatures. We're going to explore most of the space. And then, as we lower the temperature, we freeze ourselves into the most probable confirmations.

Now, there's nothing that restricts simulated annealing to protein structure. This approach is actually quite general. It's called the Metropolis Hastings algorithm. It's often used in cases where there's no energy whatsoever and it's thought of purely in probabilistic terms.

So if I have some probabilistic function-- some probability of being in some state S-- I can choose a neighboring state at random. Then I can compute an acceptance ratio, which is the probability of being a state S test over the probability of being in a current state.

This is what we did in terms of the Boltzmann equation, but if I some other formulation for the probabilities I'll just use that. And then, just like in our protein folding example, if this acceptance ratio is greater than 1, we accept the new state. If it's less than 1, then we accept it with a probabilistic statement.

And so this is a very general approach. I think you might see it in your problem sets. We certainly have done this on past exams-- asked you to apply this algorithm to other probabilistic settings. So it's a very, very general way to search the sample across a probabilistic landscape.

OK, so we've seen these three separate approaches, starting with an approximate structure and trying to get to the correct structure. We have energy minimization, which will move towards the local confirmation. So it's very fast compared the other two, but it's restricted to local changes. We have molecular dynamics, which actually tries to simulate the biological process. Connotationally very intensive.

And then we have simulated annealing, which tries to shortcut the root to some of these global free energy minima by raising the temperature, pretending at this very high temperature so we can sample all the space, and then cooling down so we trap a high probability confirmation.

Any questions on any of these three approaches? OK.

All right, so I'm going to go through now some of the approaches that have already been used to try to solve protein structures. We started off with a sequence. We'd like to figure out what the structure is. And this field has had a tremendous advance, because in 1995 a group got together and came up with an objective way of evaluating whether these methods were working.

So lots of people have proposed methods for predicting protein structure, and what the CASP group did in '95 was they said, we will collect structures from crystallographers, NMR spectroscopists, that they have not yet published but they know they're likely to be able to get within the time scale of this project. We will send out those sequences to the modelers.

The modelers will attempt to predict the structure, and then at the end of the competition we'll go back to the crystallographers and the spectroscopists and say, OK, give us a structure and now we'll compare the predicted answers the real ones. So no one knows are the answer is until all the submissions are there, and then you can see objectively which of the approaches did the best.

And one of the approaches that's consistently has done very well, which we'll look at in some detail, is this approach called Rosetta. So you can look at the details online. They split this modeling problem into two types. There are ones for which you can come up with a reasonable homology model. This can be very, very low sequence homology, but there's something in the database of known structure that it's sequenced similarly to the query. And then ones where it's completely de novo.

So how do they go about predicting these structures? So if there's homology, you can imagine the first thing you want to do is align your sequence to the sequence of the protein that has a known structure. Now, if it's high homology this is not a hard problem, right? We just need to do a few tweaks. But we get to places-- what's called the Twilight Zone, in fact-- where there's a high probability that you're wrong, that your sequence alignments could be to entirely the wrong structure. And that's where things get interesting.

So they've got high sequence similarity-- greater than 50% sequence similarity that are considered relatively easy problems. These medium problems that are 20% to 50% sequence similarity. And then very low sequence similar problems-- less than 20% sequence similarity.

OK, so you've already seen this course methods for doing sequence alignment, so we don't have to go into that in any detail. But there are a lot of different specific approaches for how to do those alignments. You could do anything from blast to highly sophisticated Markov models to try to decide what's most similar to your protein structure.

And one of the important things that Rosetta found was not to align on any single method but to try a bunch of different alignment approaches and then follow through with many of the different alignments. And then we get this problem of how do you refine the models, which is what we've already started to talk about.

So in the general refinement procedure, when you have a protein that's relatively in good shape they apply random perturbations to the backbone torsion angle. So this is again the statistical approach, the not allowing every atom to move. They're just rotating a certain number of the rotatable side chains. So we've got the fine psi angles in the backbone, and some of the side channels.

They do what's called rotamer optimization of the side chain. So what does that mean? Remember that we could allow the side chains to rotate freely, but very, very few of those rotations are frequently observed. So we're going to choose, as these three choices, among the best possible rotamers, rotational isomers. And then once we've found a nearly optimal side chain confirmation from those highly probable ones, then we allow more continuous optimization of the side chains.

So when you have a very, very high sequence homology template, you don't need to do a lot of work on most of the structure. Right? Most of it's going to be correct. So we're going to focus on those places where the alignment is poor. That seems pretty intuitive.

Things get a little bit more interesting when you've got these medium sequence similarity templates. So here, even your basic alignment might not be right. So they actually proceed with multiple alignments and carry them through the refinement process.

And then, how do you decide which one's the best? You use the potential energy function. Right? So you've already taken a whole bunch of starting confirmations. We've taken them through this refinery procedure. You now believe that those energies represent the probability that the structure is correct, so you're going to choose which of those confirmations to use based on the energy.

OK, in these medium sequence similarity templates, the refinement doesn't do the entire protein structure, but it focuses on particular region. So places where there are gaps, insertions, and deletions in the alignment. Right? So your alignment is uncertain, so that's where you need to refine the structure. Places that were loops in the starting models, so they weren't highly constrained.

So it's plausible that they're going to be different in the starting structure from some homologous protein and in the final structure. And then, regions where the sequence conservation is low. So even if there is a reasonably good alignment, there's some probability that things have changed during evolution.

Now, when they do a refinement, how they do that? In these places that we've just outlined, they don't simply randomly perturb all of the angles. But actually, they take a segment of the protein, and exactly how long those segments are has changed over the course of the Rosetta algorithm's refinement. But say something on the order of three to six amino acids. And you look in the database for proteins that have known structure that contain the same amino acid sequence.

So it could be completely unrelated protein structure, but you develop a peptide library for all of those short sequences for all the different possible structures that they've adopted. So you know that those are at least structures that are consistent with that local sequence, although they might be completely wrong for this individual protein. So you pop in all of those alternative possible structures.

So OK, we replace the torsion angles with those of peptides of known structure, and then we do a local optimization using the kinds of minimization algorithms we just talked about to see whether there is a structure that's roughly compatible with that little peptide that you took from the database that's also consistent with the rest the structure. And after you've done that, then you do a global refinement.

Questions on that approach?

OK, so does this work? One of the best competitors in this CASP competition. So here are examples where the native structure's in blue. The best model they produced was in red, and the best template-- that's the homologous protein-- is in green. And you can see that they agree remarkably well. OK?

So this is very impressive, especially compared to some of the other algorithms. But again, it's focusing on proteins where there's at least some decent homology to start with.

If you look here at the center of these proteins, you can see the original structure, I believe, is blue, and their model's in red. You can see they also get the side chain confirmations more or less correct, which is quite remarkable.

Now, what gets really interesting is when they work on these proteins that have very low sequence homologies. So we're talking about 20% sequence similarity or less. So quite often, you'll actually have globally the wrong fold-- a 20% sequence similarity.

So what do they do here? They start by saying, OK, we have no guarantee that our templates are even remotely correct. So they're going to start with a lot of templates and they're going to refine all of these in parallel in hopes that some of them come out right at the other end.

And these are what they call more aggressive refinement strategies. So before, where did we focus our refinement energies? We focused on places that were poorly constrained, either by evolution or regions of the structure that weren't well-constrained, or places where the alignment wasn't good.

Here, they actually go after the relatively well-defined secondary structure elements, as well. And so they will allow something that was a clear alpha helix in all of the templates to change some of the structure by taking peptides out of the database that have other structures. OK?

So you take a very, very aggressive approach to the refinement. You rebuild the secondary structure elements, as well as these gaps, insertions, loops, and regions with low sequence conservation. And I think the really remarkable thing is that this approach also works. It doesn't work quite as well, but here's a side by side comparison of a native structure and the best model.

So this is the hidden structure that was only known to the crystallographer, or the spectroscopist, who agreed to participate in this CASP competition. And here is the model they submitted blind without knowing what it was. And you can see again and again that there's a pretty good global similarity between the structures that they propose and the actual ones. Not always. I mean, here's an example where the good parts are highlighted and the not-so-good parts are shown in white so you can barely see them.

[LAUGHTER]

PROFESSOR: But even so, give them that. Give them their credit. It's a remarkably good agreement.

Now, we've looked at cases where there's very high sequence similarity, where there's medium sequence similarity, where there's low sequence similarity. But the hardest category are ones where there's actually nothing in the structural database that's a detectable homologue to the protein of interest.

So how do you go about doing that? That's the de novo case. So in that case, they take the following strategy. They do a Monte Carlo search for backbone angles. So specifically, they take short regions-- and again, this is the exact length. Changes in different versions of the algorithm, but it's either three to nine amino acids in the backbone.

They find similar peptides in the database of known structure. They take the backbone confirmations from the database. They set the angles to match those. And then, they use those Metropolis criteria that we looked at in simulated annealing. Right? The relative probability of the states, determined by the Boltzmann energy, to decide whether to accept or not.

If it's lower energy, what happens? Do you accept? Do you not accept?

AUDIENCE: Accept.

PROFESSOR: You accept. And if it's high energy, how do you decide?

AUDIENCE: [INAUDIBLE]

PROFESSOR: [INAUDIBLE], probability. Very good.

OK, so they do a fixed number of Monte Carlo steps-- 36,000. And then they repeat this entire process to get 2,000 final structures. OK? Because they really have very, very low confidence in any individual one of these structures.

OK, now you've got 2,000 structures, but you're allowed to submit one. So what do you do? So they cluster them to try to see whether there are common patterns that emerge, and then they refine the clusters and they submit each cluster as a potential solution to this problem.

OK, questions on the Rosetta approach? Yes.

AUDIENCE: Can you mention again why the short region of three to nine amino acids, and whether [INAUDIBLE].

PROFESSOR: So the question is, what's the motivation for taking these short regions from the structural database? Ultimately, this is a modeling choice that they made that seems to work well. So it's an empirical choice. But what possibly motivated them, you might ask, right?

So, the thought has been in this field for a long time, and it's still, I think, unproven, that certain sequences will have a certain propensity to certain structures. We saw this in the secondary structure prediction algorithms, that there were certain amino acids that occurred much more frequently in alpha helixes.

So it could be that there are certain structures that are very likely to occur for short peptides, and other ones that almost never occur. And so if you had a large enough database of protein structures, then that would be a sensible sampling approach. Now, in practice, could you have gotten some good answer in some other approach? We don't know. This is what actually worked well. So there's no real theoretical justification for it other than that crude observation that there is some information content that's local, and then a lot of information content that's global.

Yes?

AUDIENCE: So when you're doing a de novo approach, is it general that you come up with a bunch of different clusters as your answer, whereas with the homology approach, you are more confident of structure answer?

PROFESSOR: So the question was, if you're doing a de novo approach, is it generally the case that you have lots of individual, or clusters of structures, whereas in homology you tend not to. And yes, that's correct. So in the de novo, there are frequently going to be multiple solutions that look equally plausible to you, whereas the homology tends to drive you to certain classes.

Good questions. Any other questions?

All, right so that was CASP. One was in 1995, which seems like an eon ago. So how have things improved over the course of the last decade or two?

So there was an interesting paper that came out recently that just looked at the differences between CASP 10, one of are the most recent ones, and CASP 5. They're every two years, so that's a decade. So how have things improved or not over the last decade in this challenge?

So in this chart, the y-axis is the percent of the residues that were modeled and that were not in the template. OK? So I've got some template. Some fraction of the amino acids have no match in the template.

How many of those do I get correct? As a function of target difficulty, they have their own definition for target difficulty. You can look in the actual paper to find out what is in the CASP competition, but it's a combination of structural and sequence data. So let's just take them that they made some reasonable choices here. They actually put a lot of effort into coming up with a criteria for evaluation.

Every point in this diagram represents some submitted structure. The CASP5, a decade ago, are the triangles. CASP 9, two years ago, were the squares, and the CASP10 are the circles. And then they have trend lines for CASP9 and CASP10 are shown here-- these two lines.

And you can see that they do better for the easier structures and worse for the harder structures, which is what you'd expect, whereas CASP5 was pretty much flat across all of them and did about as well even on on the easy structures as these ones are doing on the hard structures.

So in terms of the fraction of the protein that they don't have a template for that they're able to get correct, they're doing much, much better in the later CASPs than they did a decade earlier. So that's kind of encouraging. Unfortunately, the story isn't always that straightforward.

So this chart is, again, target difficulty on the x-axis. The y-axis is what they call the Global Distance Test, and it's a model of accuracy. It's the percent of the carbon alpha atoms in the predictions that are close-- and they have a precise definition of close that you can look up-- that are close to the true structure.

So for a perfect model, it would be up here in the 90% to 100% range, and then random models would be down here. You can see a lot of them are close to random. But more important here are the trend lines. So the trend line for CASP10, the most recent one in this report, is black. And fore CASP5, it's this yellow one, which is not that different from the black.

So what this shows is that, over the course of a decade, the actual prediction accuracy overall has not improved that much. It's a little bit shocking. So they tried in this paper to try to figure out, why is that? I mean, the percentage of the amino acids that you're getting correct is going up, but overall accuracy has not.

And so they make some claims that it could be that target difficulty is not really a fair measure, because a lot of the proteins that are being submitted are now actually much harder in different sense, in that they're not single domain proteins initially. So in CASP5, a lot of them were proteins that had independent structures.

By the time of CASP10, a lot of the proteins that are being submitted are more interesting structural problems in that they're folding is contingent on interactions with lots of other things. So maybe all the information you need is not composed entirely in the sequence of the peptide that you've been given to test but depends more on the interactions of it with its partners.

So those were for homology models. These are the free modeling results. So in free modeling, there's no homology to look at, so they don't have a measure of difficulty except for length. They're using, again, that Global Distance Test. So up here are perfect models. Down here are nearly random models. CASP10 is in red. CASP5, a decade earlier, is in green. And you can see the trend lines are very, very similar. And CASP9, which is the dashed line here, looks almost identical to CASP5.

So again, this is not very encouraging. It says that the accuracy the models has not approved very much over the last decade. And then, they do point out that if you focus on the short structures, then it's kind of interesting. So in CASP5, which are the triangles, only one of these was above 60%. CASP9, they had 5 out of 11 were pretty good. But then you get to CASP10 and now only three are greater than 60%. So it's been fluctuating quite a lot.

So modeling de novo is still a very, very hard problem. And they have a whole bunch of theories as to why that could be. They proposed, as I already said, that maybe the models that they're trying to solve have gotten harder in ways that are not easy to assess.

A lot of the proteins that previously wouldn't have had a homologue now already do, because there has been a decade of structural work trying to fill in missing domain structures. And that these targets tend to have more irregularity. Tendency be part of larger proteins. So again, there's not enough information in the sequence of what you're given to make the full prediction.

Questions?

So what we've seen so far has been the Rosetta approach to solving protein structures. And it really is, throw everything at it. Any trick that you've got. Let's look into the databases. Let's take homologous proteins. Right? So we have these high, medium, low levels homologues. And even when we're doing a homologue, we don't restrict ourselves to that protein structure.

But for certain parts, we'll go into the database and find the structures of peptides of length three to nine. Pull those out of the [? betas. ?] Plug those in. Our potential energy functions are grab bag information, some of which has strong physical principles, some which is just curve fitting to make sure that we keep the hydrophobics inside and hydrophilics outside.

So we throw any information that we have at the problem, whereas our physicist has disdain for that approach. He says, no, no. We're going to this purely by the book. All of our equations are going to have some physical grounding to them. We're not going to start with homology models. We're going to try to do the simulation that I showed you a little movie of for every single protein we want to know the structure of.

Now, why is that problem hard? It's because these potential energy landscapes are incredibly complex. Right? They're very rugged. Trying to get from any current position to any other position requires a go over many, many minima.

So the reason it's hard to do, then, is it's primarily a computing power issue. There's just not enough computer power to solve all of these problems. So what one group, DE Shaw, did was they said, well, we can solve that by just spending a lot of money, which fortunately they had.

So they designed hardware that actually solves individual components of the potential energy function in hardware rather than in software. So they have a chip that they call Anton that actually has parts of it that solve the electrostatic function, the van der Waals function.

And so in these chips, rather than in software, you are doing as fast as you conceivably can to solve the energy terms. And that allows you to sample much, much more space. Run your simulations for much, much longer in terms of real time.

And they do remarkably well. So here are some pictures from a paper of theirs-- a couple of years ago now-- with the predicted and the actual structures. I don't even remember which color is which, but you can see it doesn't much matter. They get them down to very, very high resolution.

Now, what do you notice about all these structures?

AUDIENCE: They're small.

PROFESSOR: They're small, right? So obviously there's a reason for that. That's when you can do in reasonable compute time, even with a high-end computing that's special purpose. So we're still not in a state where they can fold any arbitrary structure.

What else do you notice about them? Yeah, in the back.

AUDIENCE: They have very well-defined secondary structures.

PROFESSOR: They have very well-defined secondary structures. And they're specifically what, mostly?

AUDIENCE: Alpha helixes.

PROFESSOR: Alpha helixes, right. And it turns out that a lot more information is encoded locally in an alpha helix than in a beta sheet, which is going to be contingent on what that piece of protein comes up against. Right? Whereas in the alpha helix, we saw that you can get 60% accuracy with very crude algorithms, right?

So we're going to do best with these physics approaches when we have small proteins that are largely alpha helical. But in later papers-- well here's even an example. Here's one that has a certain amount of beta sheet. And the structures are going to get larger with time. So it's not an inherent problem. It's just a question of how fast the hardware is today versus tomorrow.

OK, a third approach. So we had the statistical approach. We have the physics approach. The third approach, that I won't go into detail but you can play around was literally yourselves, is a game where we have humans who try to identify the right structure, just as humans do very well in other kinds of pattern recognition problems.

So you can try this video game where you're given structures to try to solve and say, oh, should I make that helical? Should I rotate that side chain? So give it a try. Just Google FoldIT, and you can find out whether you can be the best gamers and beat the hardware.

All right. So so far we've been talking about solving the structures of individual proteins. We've seen there is some success in this field. It's improved a lot in some ways. Between CASP1 and CASP5 I think there's been huge improvements. Between CASP5 and CASP10, maybe the problems have gotten hard. Maybe there have been no improvements. We'll leave that for others to decide.

What I'd like to look at in the end of this lecture and the beginning of the next lecture are problems of proteins interacting with each other, and can we predict those interactions? And that'll, then, lead us towards even larger systems and network problems.

So we're going to break this down to three separate prediction problems. The first of these is predicting the effect of a point mutation on the stability of a known complex. So in some ways, you might think this is an easy problem. I've got two proteins. I know their structure. I know they contract. I want to predict whether a mutation stabilizes that interaction or makes it fall apart. That's the first of the problems.

We can try to predict the structure of particular complexes, and we can then try to generalize that and try to predict every protein that interacts with every other protein. We'll see how we do on all of those.

So we'll go into one of these competition papers, which are very good at evaluating the fields. This competition paper looked at what I call the simple problem. So you've got two proteins of known structure. The authors of the paper, who issued the challenge, knew the answer for the effect of every possible mutation at a whole bunch of positions along these proteins on the-- well, an approximation to the free energy of binding.

So they challenged the competitors to try to figure out, we give you the structure, we tell you all the positions we've mutated, and you tell us whether those mutations made the complex more stable or made the complex less stable. Now specifically, they had two separate protein structures.

They mutated 53 positions in one. 45 positions in another. They didn't directly measure the free energy of binding for every possible complex, but they used a high throughput assay. We won't go into the details, but it should track, more or less, with the free energy. So things that seem to be more stable directors here probably are lower free energy complexes.

OK, so how would you go about trying to solve this? So using these potential energy functions that we've already seen, you could try to plug in the mutation into the structure. And what would you have to do then in order to evaluate the energy? Before you evaluate the energy.

So I've got known structure. I say, position 23 I'm mutating from phenylalanine to alanine. I'll say alanine to phenylalanine. Make it a little more interesting. OK? So I'm now stuck on this big side chain. So what do I need to do before I can evaluate the structure energy?

AUDIENCE: Make sure there's no clashes.

PROFESSOR: Make sure no clashes, right? So I have to do one of those methods that we already described for optimizing the side chain confirmation, and then I can decide, based on the free energy, whether it's an improvement or makes things worse.

OK, so let's see how they do. So here's an example of a solution. The submitter, the person who has the algorithm for making a prediction, decides on some cutoff in their energy function, whether they think this is improving things or making things worse. So they decide on the color. Each one of these dots represents a different mutation.

On the y-axis is the actual change in binding, the observed change in binding. So things above zero are improved binding. Below zero are worse binding. And here are the predictions on the submitter scale. And here the submitter said that everything in red should be worse and everything green should be better. And you can see that there's some trend. They're doing reasonably well in predicting all these red guys as being bad, but they're not doing so well in the neutral ones, clearly, and certainly not doing that well in the improved ones.

Now, is this one of the better submitters or one of the worst? You'd hope that this is one of the worst, but in fact this is one of the top submitters. In fact, not just the top submitter but top submitter looking at mutations that are right at the interface where you'd think they'd do the best, right?

So if there's some mutation on the backside of the protein, there's less structural information about what that's going to be doing in the complex. There could be some surprising results. But here, these are amino acid mutations right at the interface.

So here's an example of the top performer. This is the graph I just showed you, focusing only at the [? residues ?] of the interface, and all sites. And here's an average group. And you can see the average groups are really doing rather abysmally. So this blue cluster that's almost entirely below zero were supposed to be neutral. And these green ones were supposed to be improved, and they're almost entirely below zero. This is not encouraging story.

So how do we evaluate objectively whether they're really doing well? So we have some sort of baseline measure. What is it the sort of baseline algorithm you could use to predict whether a mutation is improving or hurting this interface? So all of their algorithms are going to use some kind of energy function. What have we already seen in earlier parts of this course that we could use?

Well, we could use the substitution matrices, right? We have the BLOSUM substitution matrix that tells us how surprised we should be when we see an evolution, that Amino Acid A turns into Amino Acid B. So we could use, in this case, the BLOSUM matrix. That gives us for each mutation a score. It ranges from minus 4 to 11. And we can rank every mutation based on the BLOSUM matrix for the substitution and say, OK, at some value in this range things should be getting better or getting worse.

So here's an area under the curve plot where we've plotted the false positives and true positive rates as I change my threshold for that BLOSUM matrix. So I compute what the mutation BLOSUM matrix is, and then I say, OK, is a value of 11 bad or is it good? Is a value of 10 bad or good? That's what this curve represents. As I vary that threshold, how many do I get right and how many do I get wrong?

If I'm doing the decisions at random, then I'll be getting roughly equal true positives and false positives. They do slightly better in the random using this matrix. Now, the best algorithm at predicting that uses energies only does marginally better. So this is the best algorithm at predicting. This is this baseline algorithm using just the BLOSUM matrix. You can see that the green curve predicting beneficial mutations is really hard. They don't do much better than random. And for the deleterious mutations, they do somewhat better.

So we could make these plots for every single one of the algorithms, but a little easier is to just compute the area under the curve. So how much of the area? If I were doing perfectly, I would get 100% true positives and no false positives, right? So my line would go straight up and across and the area under the curve would be one.

And if I'm doing terribly, I'll get no true positives and all false positives. I'd be flatlining and my area would be zero. So the area under the curve, which is normalized between zero and one, will give me a sense of how well these algorithms are doing.

So this plot-- focus first on the black dots-- shows at each one of these algorithms what the area under the curve is for beneficial and deleterious mutations. Beneficial on the x-axis, deleterious mutations on the y-axis. The BLOSUM matrix is here.

So good algorithms should be above that and to the right. They should having a better area under the curve. And you can see the perfect algorithm would have been all the way up here. None of the black dots are even remotely close. The G21, which we'll talk about a little bit in a minute, is somewhat better than the BLOSUM matrix, but not a lot.

Now, I'm going to ignore the second round in much detail, because this is a case where people weren't doing so well in the first round so they went out and gave them some of the information about mutations at all the positions. And that really changes the nature of problem, because then you have a tremendous amount of information about which positions are important and how much those mutations are making. So we'll ignore the second round, which I think is an overly generous way of comparing these algorithms.

OK, so what did the authors of this paper observe? They observed that the best algorithms were only doing marginally better than random choice. So three times better. And that there seemed to be a particular problem looking at mutations that affect polar positions.

One of the things that I think was particularly interesting and quite relevant when we think about these things in a thermodynamic context is that the algorithms that did better-- none of them could be really considered to do really well-- but the algorithms that did better didn't just focus on the energetic change between forming the native complex over here and forming this mutant complex indicated by the star. But they also focused on the affect of the mutation on the stability of the mutated protein.

So there's an equilibrium not just moving between the free proteins and the complex, but also between moving between the free proteins that are folded and the free proteins that are unfolded. And some of these mutations are affecting the energy of the folded state, and so they're driving things to the left, to the unfolded. And if you don't include that, then you actually get into trouble.

And I've put a link here to some lecture notes from a different course that I teach where you can look up some details and more sophisticated approaches that actually do take into account a lot of the unfolded states.

So the best approach-- best of a bad lot-- consider the effects of mutations on stability. They also model packing, electrostacks, and solvation. But the actual algorithms that they used were a whole mishmash of approaches. So there didn't seem to emerge a common pattern in what they were doing, and I thought I would take you through one of these to see what actually they were doing.

So the best one was this machine learning approach, G21. So this is how they solved the problem. First of all, they dug through the literature and found 930 cases where they could associate a mutation with a change in energy. These had nothing to do with proteins under consideration. They were completely different structures.

But they were cases where they actually had energetic information for each mutation.

Then we go through and try to predict what the structural change will be in the protein, using somebody else's algorithm, FoldX. And now, they describe each mutant, not just with a single energy-- we have focused, for example, on PyRosetta, which you'll use in process-- but they actually had 85 different features from a whole bunch of different programs.

So they're taking a pretty agnostic view. They're saying, we don't know which of these energy functions is the best, so let's let the machine learning decide. So every single mutation that's posed to them as a problem, they have 85 different parameters as to whether it's improving things or not.

And then, they had their database of 930 mutations. For each one of those they had 85 parameters. So those are label trending data. They know whether things are getting better or worse. They actually don't even rely on a single machine learning method. These actually used five different approaches.

We'll discuss Bayesian nets later in this course. Most of these others we won't cover at all, but they used a lot of different computational approaches to try to decide how to go from those 85 parameters to a prediction of whether the structures improved or not.

So this actually shows the complexity of this apparently simple problem, right? Here's a case where I have two proteins of known structure. I'm making very specific point mutations, and even so I do only marginally better than random. And even throwing at it all the best machine learning techniques. So there's clearly a lot in protein structure that we don't yet have parametrized in these energy functions.

So maybe some of these other problems are actually not as hard as we thought. Maybe instead of trying to be very precise in terms of the energetic change for a single mutation at an interface, we'd do better trying to predict rather crude parameters of which two proteins interact with each other. So that's what we're going to look at in the next part of the course. We're going to look at whether we can use structural data to predict which two proteins will interact.

So here we've got a problem, which is a docking problem. I've got two proteins. Say they're of known structure, but I've never seen them interact with each other. So how do they come together? Which faces of the proteins are interacting with each other? That's called a docking problem.

And if I wanted to try to systematically figure out whether Protein A and Protein B interact with each other, I would have to do a search over all possible confirmations, right? Then I could use the energy functions to try to predict which one has the lowest energy. But it actually would be a computationally very inefficient way to do things.

So we could imagine we wanted to solve this problem. For each potential partner, we could evaluate all relative positions and orientations. Then, when they come together we can't just rely on that, but as we've seen several times now we're going to have to do local confirmational changes to see how they fit together for each possible docking. And then, once we've done that, we can say, OK, which of these has the lowest energy of interaction?

So that, obviously, is going to be too computationally intensive to do on a large scale. It could work very well if you've got a particular pair or proteins that you need to study. But on a big sale, if we wanted to predict all possible interactions, we wouldn't really be able to get very far. So what people typically do is use other kinds of information to reduce the search space. And what we'll see in the next lecture, then, are different ways to approach this problem.

Now, one question we should ask is, what role is structural homology going to play? Should I expect that any two proteins that interact with each other-- let's say that that Protein A and I know its interactors. So I've got A known to interact with B. Right? So I know this interface.

And now I have protein C, and I'm not sure if it interacts or not. Should I expect the interface of C, that touches A, to match the interface of B? Should these be homologous? And if not precisely homologous, then are there properties that we can expect that should be similar between them?

So different approaches we can take. And there are certainly cases where you have proteins that interact with a common target that have no overall structure similarity to each other but do have local structural similarity. So here's an example of subtilisn, which is shown in light gray, and pieces of it that interactive with the target are shown in red.

So here are two proteins that are relatively structurally homologous-- they interact at the same region. That's not too surprising. But here's a subtilisn inhibitor that has no global structural similarity to these two proteins, and yet its interactions with subtilisn are quite similar.

So we might expect, even if C and B don't look globally anything like each other, they might have this local similarity.

OK, actually I think we'd like to turn back your exams. So maybe I'll stop here. We'll return the exams in the class, and then we'll pick up at this point in the next lecture.