PROFESSOR: OK. So we've been talking about predicting structure proteins. At the end of the last lecture we started to talk a little bit about predicting interactions, and that's going to be the focus of today's lecture. And we identified a couple of different possible prediction challenges.

One was quantitative predictions of what happens when you make specific mutations in a known protein complex. We talked about trying to predict the structure of, say, just a pair of proteins, and then trying to do that on the global scale for all known proteins.

And so last time, if you recall, we thought that initially maybe this would be a simple problem. We have proteins of known structure with a complex. Structure of the complex is also known. And we want to make predictions as to the change in affinity when there's a specific mutation made.

In principle, this should be easy because we have all those different formulations for the potential energy function. And so if we figure out what the local structural changes are that are due to the insertion or deletion of some side chain, then we should be able to predict the change in the potential energy, and therefore the change in the energy of the complex. But in fact, it turned out that it was very, very hard to do that.

And so this plot compared-- the black circles were the prediction algorithms for this problem, compared to just simply a substitution matrix, the BLOSUM substitution matrix defined in terms of the area under the curve for beneficial mutations and deleterious mutations. And you can see that very, very few of the black dots get far away from what is the really simple default model. A lot of them do worse.

So OK, well maybe that's not such a simple problem because it requires a highly quantitative prediction. Maybe we'll do better just trying to predict which proteins interact at all. And so that's going to be the focus of today's lecture.

Now, that also had a problem, right? Because even if I know the structure of two proteins, I don't know necessarily what surfaces of those proteins interact. And so I have to figure out this docking problem of which part of protein A interacts with which part of protein B.

That's the beginning of my problem, and then I have to make a series of subsequent decisions. So I'm going to have to figure out for any potential partner of my protein, I need to figure out the docking problem, the relative position orientation. Now, in this little cartoon, it's shown as a completely static protein that approaches another static protein. The only thing that's changing is the relative coordinates.

But of course, there will be local changes in confirmation, perhaps even global ones. And so we need to be able to make some estimates as to what those structural rearrangements will be when the two proteins interact. And then after we've come up with our best estimate of the structural rearrangements, only then can we come up with an estimate of the energy interaction and decide whether it's better than some threshold.

OK. So one of the problems that's pretty obvious from this is that this kind of approach in principle, if we do it rigorously through all the steps, would be extremely slow. Now, another part that's perhaps a little bit less obvious is that it's going to be very prone to false positives. And why do you think that might be? What am I not taking into account here?

AUDIENCE: Are you not taking into account the desolvation [INAUDIBLE].

PROFESSOR: So one answer is I'm not taking account of the desolvation, but in fact, I can do that. Right? So some of the potential energy functions we looked at, the statistician's version rather than the physicist's makes it pretty easy to incorporate the

desolvation. Any other thoughts as to what I'm not taking into account? What other protein should I be considering when I'm considering an interaction problem?

So I've isolated, in this case, two proteins. I'm saying, in a universe where these are the only two proteins that exist, will they have a favorable energy interaction? What I really need to know is whether that energy interaction is more favorable than all the competing interactions that they could have.

So even if I find something that's potentially a good interaction, it may not be the best possible interaction. And if I consider then the concentration of this protein and the concentration of all the other molecules out there that have a higher affinity, then it could turn out that this is actually a rather poor substrate for my protein, a rather poor interaction partner. So we have that false positive problem. OK.

But let's focus on the computational efficiency problem, because that's at least one that we can come up with some nice algorithms to try to solve. So what we want to do is try to limit our search space. If I want to figure out-- I have a query protein and I want to ask, what does it interact with, instead of trying to do the pairwise comparison of this protein with every other protein in the database, and doing very precise structural calculations on all of those, maybe there's some way that I can prefilter the set of proteins that it might interact with.

And that's what we're going to look at. So we're going to try to officially choose potential partners before we're doing any structural comparison. And then once we have those partners, we're going to try to avoid having to do detailed calculations until we have a relatively high degree of confidence that these proteins could interact by other criteria.

And we're going to look at two papers that describe algorithms for solving this problem, and they're both uploaded to the website. The first thing that we'll look at is called PRISM that actually uses structural calculations. And then we'll look at PrePPI, which deals with everything purely at-- without actually explicitly calculating the structures.

OK. So what does PRISM do? Well, it's based on the notion that there are a limited number of architectures that we could look at for which proteins can interact. And so if we can identify those architectures, then we can try to figure out whether a protein is a potential partner of another one before we do the detailed, costly calculations.

In addition, in those architectures, not all amino acids are going to be equal, but there are going to be some that contribute more to the energy than others. And so by identifying those critical residues, we can once again focus our computational energy on those complexes that are most likely to be important.

OK. So it has these two components-- a rigid-body structural comparison. So that's that two proteins are not changing their own coordinates, they're just being brought together in different conformations. And then once the proteins have passed a series of checks, then we allow for flexible refinement using the kinds of energies we looked at in the previous lectures to decide how high affinity this complex could be.

And the critical thing is that we're going to make some of these early decisions after the rigid-body comparison using structural similarity, evolutionary conservation, and particularly looking at these regions that are called hotspots. These are sites where most of the free energy of interaction occurs during an interface. So it's not, as I said, uniformly distributed.

So I showed you this slide last time. It shows chymotrypsin in a light gray and its interaction with some protein partners. These two share some global similarity to each other, whereas this partner is quite different from either of these two globally. But you can see that at the interface, it's actually quite similar. And so this gives you hope that even if you can't find a direct homologue-- so if you were trying to figure out, what does this protein in yellow interact with, and you searched the database and you couldn't find anything that was its structural homologue, but if you could figure out to look for homologues of the lower regions that interact, you might be able to figure out that it interacts with the same protein as this one and this one. OK.

So what about this idea of hotspots? And this was an idea that was first developed

in 1995 by this paper, Clackson and Wells, where they were looking at the interaction of a cell surface receptor with its ligand approaching. And they did systematic mutagenesis across the surface of the interface to see when I mutate any single amino acid to alanine, how much it affects the energy of interaction.

What they found was things were highly non-uniform. So this lower curve shows the change in free energy when you mutate particular individual amino acids to alanine. And you can see there are big losses of free energy at some places, and other places there's almost no change in the free energy binding. In a few places you actually get a benefit from mutating a side chain to alanine.

So in this particular case, and it's held up over many, many cases then, the free energy of binding is not uniform across the surface, but it's distributed in what has been called hotspots. So here is a structure of the human growth hormone and its receptor. And in red are the few amino acids that contribute very, very large amounts-- more than one and a half kcals per mole-- to the energy of interaction.

And it doesn't correspond with any simple structural parameter. So it's not the amino acids that have the biggest surface area, for example, or anything like that. So it's not trivial to figure out what these regions are, although there are some prediction algorithms.

So there are studies, and subsequent ones have indicated that roughly 10% of the amino acids at the interface are the ones that have the biggest contribution. There are some trends, but none of these are hard rules. These tend to be rich in these three amino acids-- tryptophan, arginine, and tyrosine.

If you might imagine, these are regions of the protein that are highly complimentary. So there'll be a patch on one side that's a hotspot matching up with another patch on the other protein that's also a hotspot. And it's kind of an interesting note that around these regions where the hotspots occur, there are other amino acids that exclude solvent from the interface. And they call that an o-ring. So these are some of the features that tend to occur with protein interfaces.

So in this PRISM algorithm, what they do is the following. They start off with a template-- two proteins that are known to interact-- and they define the interface simply by close approach of amino acids in one chain to amino acids in the other. So in this case, shown in these balls are regions of the proteins that interact.

And then they isolate the interfacial residues. Ignore the rest of the protein, because we said that the parts that interact in different proteins could be homologous even if the global structures of the proteins are not, right? So we're going to do our structural similarity calculations purely on the interface residues and not on the entire structure.

So then with that template, you can then look at lots of proteins and see whether they have any structural match to pieces that interact. So here they've identified this protein, ASPP2, which has structural homology to I kappa b at the interface. Although globally it's quite different.

And now, once they have this potential partner for NF kappa b, this ASPP2, they're going to test whether there's a good structural match, whether specifically in the regions that are hotspots-- they have an algorithm for predicting hotspots-- whether the match is good, whether it's sequence conservation at those hotspots. And only then do they do the refinement to do the flexible refinement of the type that we looked at in the previous lecture, energy minimization, and other approaches to figure out what the best possible structure of this complex would be, and then what it's free energy would be.

So here's their description of the problem. They have template proteins and targets. They do a structure alignment. They asked whether it passes some thresholds. These are very, very fast calculations to do. And only if they pass these fast calculations do you do more detailed calculations. And finally, only if it passes this do you do the very computationally expensive refinement.

And then one critical thing to remember from this algorithm is that it doesn't require the template and its query to be perfectly matched in structure. In fact, the elements of the structure at the interface could come from different parts of the chain. So they

don't take into account the chain order.

So if I had a beta sheet structure in one protein that looks like this, in my query these two proteins could be very indirectly connected. I don't care that there's a huge gap in the insertion. I just care that locally at the interface, one protein looks a lot like the other. There was a question in the back.

AUDIENCE: How do you search a database for 3d structures? Are you just looking at all the [INAUDIBLE]?

PROFESSOR: That's right. So the question was, how do you search a database for 3D structure? You do structural similarity comparisons that are based on the 3D coordinates. The simplest way to do it, but not the most efficient, is to find the rigid-body superpositions that minimize the root mean squared deviation, which was a metric we gave in one of the previous lectures.

There are faster things you can do as well. You could imagine that you could look at certain global features of elements of secondary structure and so on. And there's been a lot of work making those algorithms very fast. Other questions? Good question.

So they give an example in their papers that starting off with this known structural complex, cyclin-dependent kinase, the cyclin, and p27, the inhibitor. And then looking for structural matches. So we can identify this potential structure match. You refined it, get an energy of interaction. Try another one that has no global structural similarity. Again, once it passes all the checks, you compute the refinement and the energy. And similarly with this side.

And so from this initial complex, where we had these two proteins which were known to interact in the PDP they can make predictions that these other proteins are likely to interact even though, again, at the global level, there's very little sequence similarity. Is that clear?

OK. So the advantage of this is that it eventually does do these structural refinements that allow us to figure out the best match between two potential

interacting proteins. But that's also its weakness because that takes a lot of computational time.

So this other approach called PrePPI never actually does those structural refinements of the type we talked about in the previous lecture. So if so, how does it figure out whether the two proteins are likely to interact? So this is their schematic, and we'll go through the steps.

So you start off with two query proteins that you want to know if they interact. And you do sequence similarity to a database of known structures. So you find sequence homologues to those proteins. And so they call those homology models. MA and MB.

And now they look through the database for all the structural homologues, not sequence homologues, but structural homologues of MA and MB. So they get a series of neighbors that they call NA 1 through n and NB 1 to n. So these are the neighbors of these homologues.

And they asked whether any of these neighbors, anything in this row, anything in this row, are known to interact. And that potential interaction then could be a model for the interaction of the query, right? So far so good.

Then they do a sequence alignment. They sequence alignment of MA and MB, which are the known structural homologues of the queries, and the two proteins that are known to interact. And so now they've got this potential model for the interaction of the queries made up of two proteins of known structure that have homologues that are known to interact. OK? So it's two steps removed from the actual interaction.

Now, while their figure says that they do a structural superposition, that's not, in fact, what they do. If you look at it carefully, it's a sequence analysis. And I'll take you through the steps in a second. So they mean structured in a rather loose way. So they're only doing sequence comparisons here. They're never actually building a homology model for the queries. OK

So this figure comes from the supplement where, for some mysterious reason, they've changed all the nomenclature. So things that previously were called NA and NB have now been called TA and TB. Take what you get. So this is a pair of interacting proteins where the structure of the interaction is known. And they're structural neighbors of NA and NB, which you don't know whether they interact or not.

They identify interacting residues in this structure. That's why it's represented by these black lines connecting blue dots. So these are interacting residues from the two template proteins and neighbors NA and NB. And they asked whether the amino acids in MA and MB also are good matches for this interface. And they have a number of criteria for doing that.

So they come up with five measures. The first of those measures is a structural similarity between these MA proteins and the MA and MB and NA and NB. Then similarity-- OK, similarity is the structural similarity. Then they asked, how many of the amino acids at this interface, and what fraction of the amino acids at the interface can be aligned? So this is a sequence-based alignment of MA and-- well, it's here called TA, but was previously called MA. Just to make life complicated. So this is the sequence-based alignment.

These are they interacting residues, all the blue ones in the structure of TA and TB interacting. And they asked, what fraction and what number of these amino acids are aligned in this sequence alignment? So here they come up with a number. In this case, I guess, it's four amino acids in this-- four pairs, I should say, of the amino acids-- one, two, three, and four, indicated by these four lines-- are both interacting in the structure of the complex and can be aligned to sequences in MA and MB.

And then they use these other algorithms that are based primarily on machine learning looking at protein interfaces to decide whether the sequence of the amino acids that are going to sit at those places in the interface are likely to be residues that typically occur at interfaces. So this is the kind of statistics that I showed you before from those old papers that said 10% of the amino acids are in these

hotspots. Certain kinds of amino acids are predominant there. So the number of algorithms, and they list a bunch, that they use to come up with a score to decide whether these residues, in fact, are statistically likely to be good matches. So they have these criteria and they decide then that some fraction of the amino acids at this interface in MA and MB are likely to be reasonable ones to be at the interface.

So with all that done, they then use all of these different scores with a Bayesian classifier, and we'll talk a little bit later in this lecture and probably the next lecture as well as to what a Bayesian classifier is. But they plug all those scores in that they've derived from these proteins to decide whether these two proteins are likely to interact.

So the advantage of this approach is it's extremely fast. Everything we've talked about are very, very quick calculations. Even the structural alignments are fast. The sequence alignments, of course, are. So we get through the whole database very quickly. So they've actually computed the potential attraction partners of every pair of proteins in various genomes based solely on these alignments.

The disadvantage-- so what's the disadvantage of this method?

AUDIENCE: Can't get a de novo interaction?

PROFESSOR: We can't get any de novo interaction, so if there's no neighboring structures that interact, they'll never come up with it. So that's an important point. And then the other problem is, because it doesn't have the structural refinement, it's given up on that slow calculation, so also loses a lot of potential specificity. All the conformational changes that can occur will be lost to an algorithm like this.

So we have these two competing approaches. Yes, questions in the back.

AUDIENCE: Couldn't this method actually be used as an input to, say, a refinement step, for example?

PROFESSOR: The question was, could you use this kind of approach as an input to the refinement step? And absolutely one could. Is there another question back there? Other

questions?

All right. So we're going to take a slight turn here in the course lecture and move away from a purely computational approach and actually look at how interaction measurements are made. One of the big changes of the last decade or so is that we've gone from an era when interactions were measured pairwise to interactions being measured in bulk. So through high throughput measurements. And we'll see that that leads us to some statistical problems which eventually bring us back to some computational issues as well.

So if you want to measure all the proteins that interact in an organism, turns out to be, obviously, very difficult. One big advance that's helped with this is the idea of tagging proteins and using mass spectrometry to figure out what they interact with. So in these two sets of papers, which were some of the early ones being done in yeast, they took one protein at a time and attached a tag to it. And I'll talk about exactly what those tags are, but those are labels that allow you to attach it to a solid support.

And then by attaching to a solid support, you could then purify any proteins that stuck to protein one here. And then after you purify them, you can run them out on a gel, cut them out, and figure out what the identity of those interacting proteins were by mass spec. So this sounds very labor intensive, but it's still a lot faster than anything that came before it. And with this approach, they were able to go through entire genomes, proteomes I should say, and figure out all the interacting partners for very, very large fractions of all the proteins there.

So with this approach, what kinds of proteins do you think are likely to be false positives? Any thoughts? Yes.

AUDIENCE: Proteins stuck on the column that has nothing to do with interaction [INAUDIBLE].

PROFESSOR: Exactly. So one thing that can be quite problematic are proteins that stick to the column regardless of which protein you put there. And we'll see an approach to getting rid of that. Other kinds of problems? A variant of that. Thoughts?

What about proteins that tend to stick to other proteins non-specifically, right? Those are going to be quite problematic too. And what are the likely false negatives in an approach like this? The proteins that really do interact with the blue one but aren't picked up. Yes.

AUDIENCE: Weak interaction partners [INAUDIBLE]

PROFESSOR: Weak interaction partners, things, particularly with short half lives. Because you do a lot of washing, so it's going to be dependent on half-life. Very good. What else? Yeah.

AUDIENCE: Maybe something that interacts in tag region?

PROFESSOR: Something interacts in the tag region, right. So something interacts right around here would be lost because this would sterically interfere. Very good. Anything else? What about the concentration of proteins. How does that influence whether they show up here?

All right. So if I have a very high concentration protein, it may interact even though naturally it doesn't. They never see each other. They're in different compartments. But when [INAUDIBLE] and do this. But low abundance proteins are going to be quite problematic because there'll be very little of them in these complexes compared to the high abundance proteins. It won't be detected by this method. They will never get to the mass spec, and so on. So we've got both false positives and false negatives in these approaches.

Now, one of the things that came up was proteins that stick non-specifically to the column. And there was a clever approach in one of these early papers that got picked up to avoid that. And this is called tandem affinity purification, or TAP-tags. And the idea is the following.

We have some gene. And we use homologous recombination-- this was done in yeast where this is easy-- to insert this sequence, which codes for the following. A piece of protein of no particular function, as far as anyone knows, a spacer, followed by this calmodulin-binding protein, followed by a protease recognition site, and then

by protein A.

So once this protein gets expressed-- and it gets expressed in it's native levels because you're inserting this into the genome. So it's not on an exogenous promoter. It's in its normal position. Whatever that protein was, then has it as C terminus all these pieces. So how does that help?

In the purification, we start with something, IgG IGG, that binds to protein A. So now that's what attaches us to the solid support. And attached to the solid support will be all those things that are nonspecific binders.

And so if I have some nonspecific binder that just likes my solid support, it'll be here. Nonspecific. And if I just acid washed everything off the column and ran my gels with that, or boiled it off in SDS, I would get the nonspecific protein too. But what they do instead is they instead cleave here with a very specific protease that recognizes this site. It's called a tobacco etch virus protease. It has a very long recognition sequence. You can make sure it doesn't cut anywhere in any other protein.

And so now, instead of alluding non-specifically with acid or detergent, you allude specifically with TEV, and then this part of the protein will fall off. And then you do a second purification that relies on this piece of the protein. So you pull out only the things that you want that have the CBP, the calmodulin binding protein, by having different kind of solid support that has calmodulin attached to it.

And so through this process, you can get rid of a lot of nonspecific binders. It doesn't help you with the false negatives, right? You've made the wash conditions even harsher so you're going to lose more proteins. But you'll pick up fewer false positives.

And then finally, the last purification procedure actually uses EGTA, which is a chelating agent. So this interaction between CBP and calmodulin depends on calcium. EGTA sucks the calcium out of that interaction. And so it's, again, a very specific way of alluding rather nonspecific one, like heat, salt, acid, or detergent.

So this has been one technology, affinity purification followed by mass spec, that's given us a lot of information on protein-protein interactions. And a computing technology that's also contributed quite a lot is called yeast two-hybrid.

So in this approach, you have a reporter gene that normally is not going to be transcribed. It has at a design DNA binding site, a DNA binding protein, and your bait protein. And you want to figure out every protein that can interact with this prey. So the prey now is attached to an activation domain.

If these two proteins don't interact, the activation domain never gets recruited to this reporter, there's no transcription. But if the green protein and the blue protein interact, then the activation domain is going to be recruited to this promoter and it's going to turn on transcription, and then you'll get a signal.

So what are some of the advantages of this approach? It doesn't require you to purify anything. So it should be much more sensitive to low abundance proteins. So that's definitely an advantage.

It'll pick up a lot of those transient interactions. You may not get continuous activation, but you'll get transient activation. And if you've set the conditions up properly, you can pick up the transient activation.

But it has its own biases, so none of these techniques are going to be perfect. It's going to be biased against proteins that don't express well. This is, as the name implies, typically done in yeast. So if you have human proteins and you express them in yeast, or plant proteins that you express in yeast, there could be some proteins that just will not express well in that organism.

What else can be a problem? Some proteins don't do well in the nucleus, right? So if you're interested in interactions with membrane proteins, it's going to be very hard to get them to express in the nucleus, and therefore, you'll never pick up those interactions.

OK. So we've got these two different technologies-- the affinity capture mass spec

and the two-hybrid. Questions on those technologies? Yes.

AUDIENCE: Could another control be for the mass spec purification just to subtract out everything that alludes non-specifically.

PROFESSOR: The question was, could you subtract out anything that's nonspecific. And yes, if you've got what you might call frequent flyers, proteins that show up in every single purification, then you can simply ignore those. And that is often done. So that'll help you with things that are very nonspecific for the surface.

What's more of a problem are proteins that have some affinity for your protein x but are not really highly specific for it. So they tend to bind in certain kinds of patches. Those would be harder to figure out because they won't stick to everything. Good question. Other questions?

All right. So we've got these different technologies. What we'd really like to be able do is we know that there are problems in each approach. We'd like to be able to compute the probability that two proteins interact based on the data. So now we're turning back to the more mathematical computational approaches.

So if we just consider one experiment-- and we're going to talk about gold standard. So what's a gold standard? It's a set of proteins that we have extremely high confidence interact because it was analyzed by some other technology. Not two-hybrid, non-affinity capture mass spec, but much, much more direct interactions. By physical measurements, maybe the structural work. So the number of criteria that go into it.

So we have this gold standard data set where we know the proteins definitely interact, and we have our experiment. So clearly anything in the overlap, we can count as true positives, right? We detected it. It's in the database of gold standards. And things that are in the gold standard that we missed are obviously false negatives. We report them as non-interacting, but in fact they do.

The question is, how much of this is true positive? Everything that's detected in the experiment but we have no information for it in the database. So that could be for

one of two reasons, right? That could be that they really don't interact. Or it could be that no one's measured it. The whole point of this experiment is to find new things.

So is there any way to estimate what fraction of all the things that are unique to this experiment are true positives, and what fraction are false positives? Those we'd like to try to figure out.

Now, if we just had one experiment, that would be very challenging. But what happens when we've got two experiments? So we have these two affinity capture mass spec experiments, or maybe affinity capture mass spec and a two-hybrid. So now let's think about the overlap of those two experiments with the gold standard.

So I've got this region of overlap between experiment 1 and experiment 2, and then this region that's overlapping between all three things. Experiment 1, experiment 2, and the gold standard. So these clearly are two positives, right? They're high confidence because I picked them up in both experiments, and they're in the gold standard.

What about all these things in what I've labeled here region 2? Well, if we believe that these two experiments are independent of each other in a rigorous way-- so let's say one's a two-hybrid and one's an affinity capture mass spec, there's no particular reason that the false positives for one would be false positives in the other. In that case, I can call this region 2 my consensus true positives. I have a very high confidence that these are true interactors. Everyone buy that? Seem reasonable?

OK. So here's where the trick comes in. What fraction of all these consensus true positives are picked up in the gold standard? This ratio, right? Region 1 over region 2. OK.

So now I've got this region of things that are picked up-- the true positives from this experiment, then the gold standard. And then I've got this region that's unique to experiment 2 and it's going to be some mix of true positives and false positives. And the authors of this paper that are cited here make the following argument.

We're going to assume that the ratio of I to II is the same as the ratio of III to IV. So the fraction of consensus true positives that are picked-- these are independent experiments. So the fraction of true positives that are picked up in the gold standard is going to be constant, whether they're in the consensus or not.

So the fraction at ratio of I to II is going to be the same as the ratio of III to IV. So by that then, I can figure out how much of this region consists of true positives and how much consists of false positives. Everyone buy that? Yeah.

AUDIENCE: Can I check-- are we not saying that the gold standard represents all true positives?

PROFESSOR: Correct. Well, we're saying that the gold standard consists of things that we know to interact--

AUDIENCE: But there may be more.

PROFESSOR: But there may be more. And the goal of our experiment is to find those other ones. All right. So if you accept that premise, which seems plausible, then you can compute what fraction of all the things that are picked up in each of these experiments are likely to be true positives.

So drum roll please. It turns out that the number's not that high. So the fraction of things in the consensus was 347 out of almost 2000. And if you do the math then, what you end up with is that the true fraction in this region, for which we have no data, is 1,123 out of-- and the false piece in this is going to be almost 15,000.

And they went ahead and did this for a number of different experiments and computed the fraction of derived false positives for these data-- might be a little bit hard to see on this screen. But the numbers range from 50% false positives to, in some cases, over 90% false positives. That's a little disturbing, right? So these technologies are good at picking up interactions, but there's reason to be very skeptical.

OK. So now we've got a serious problem, because how are we going to figure out which of these interactions to trust when we know that a very, very large fraction of

them are false positives? So what could you do? Well, you could take only the little bit of overlap. You could say, I have that Venn diagram-- method 1, method 2. They did agree on a bunch of things. So I could take only those.

That obviously throws away a lot. Someone else suggested we could throw away the sticky proteins, right? So maybe there are nonspecific proteins that don't show up in every experiment, but they show up in a very, very large fraction of all experiments. Maybe I toss those out. That's another possibility.

But what we really want to do is actually come up with a probability estimate. To not have to make a hard decision, but come up with an estimate of the probability that things interact based on all the data. So how do we go about doing that?

So first of all, what happens if you just require a consensus? So this plot shows accuracy and coverage of the gold standard for individual experiments with different thresholds for deciding what's interacting, different cutoffs and things. So the individual experiments are shown here.

And then if you acquire two methods to pick something up, or three methods to pick something up, you can get better and better in your accuracy. This is a log-log plot. So if you require three methods to agree before you call something a true positive, you can get up to-- I'm not sure exactly what this is, but 80%, 90% possibly. Right? But look at where you at the y-axis. You'd only get about less than 1% coverage of the gold standard. So that's not a great approach.

So what we really want to do, as I said, is to try to estimate the probability that proteins interact given all of our available data. And the data could be specific experiments. Say the two different mass spec experiments we just referred to. Or as we'll see a little bit later in this lecture and possibly the next one, other kinds of extraneous data that are not direct physical measurements of interaction, but might give us confidence that things interact based on similarity in annotation, or similarity in gene expression, and so on. And we'll get into details of that.

OK. So to do this, we need to have a little bit of a refresher on Bayesian statistics.

So I want to measure the probability that an interaction is true given the available data. Right? And I can estimate that based on the probability of observing the data for things that I know to be true and these prior estimates. So what's the prior probability that an interaction is true and the prior probability of observing a particular data set.

Now, this by itself isn't really that helpful. I haven't told you yet how to calculate any of the terms on the right. But bear with me. If I want to decide the likelihood that a protein interacts-- how likely is it? Is it more likely that it interacts or not? I can compute this ratio. The probability that the interaction is true given the data over the probability an interaction is false given the data. That's the likelihood ratio.

So by this formula, I then cancel out this probability of the data, the prior probability of the data. And if I had a way of calculating this, and we'll get to it in a second, then if it's more likely than not to be a true interaction, I can call it an interaction, right, if it's less likely. So if this ratio is greater than 1, I accept it as a true interaction. If this ratio is less than 1, then I reject it.

OK. So now our challenge is to figure out how to compute these terms. One more thing to note is if all I want to do is be able to rank every interaction by this likelihood ratio, rather than coming up with a hard threshold, then I actually don't need all these terms. So this is the likelihood ratio. I can convert it to a log space. So it's going to be the sum of these two terms.

And if I'm simply ranking everything by this log likelihood ratio, this term is the same for every interaction. It's just composed of prior probabilities. So it's not going to affect the ranking at all. Any questions on that? Is that clear? Good.

So if I just want to come up with a ranking function, all I need to do-- all-- I need to do is to be able to estimate the probability of observing data for true interactions and the probability of observing that set of data for false interactions. Everybody buy that? Yes, please.

AUDIENCE: When you say that prior probability is the same for all interactions, we're saying

we're assuming the same prior probability for all, or is this [INAUDIBLE]?

PROFESSOR: That's its definition. We mean, what is the prior probability that proteins interact versus the prior probability? So it's independent of the proteins that we're looking at. Other questions?

All right. So we need a way of computing this piece of all the things we've looked at before. So how do we get an estimate of the probability observing a particular configuration of the data? Meaning, I detect it in experiment 1 and not in experiment 2, but in experiment 3. What's the probability of that given it's a true interaction? So that's what we're going to dive into right now.

OK. So one thing we could do to make life simpler, and then we'll remove this simplification later, but let's, for the time being, assume that all of my data are independent. So the two-hybrid is going to have completely different mistakes than the affinity capture mass spec. So those two data sets are going to be completely independent of each other.

So I can write this as a product of a particular observation-- a particular mass spec experiment and a particular two-hybrid experiment for true attractions and false interactions. So it's the product of the probability that a particular experiment would detect an interaction if the interaction is true over the probability that that particular experiment would detect it if there was no interaction. I'm just going to multiply all of those probabilities. Yes.

AUDIENCE: [INAUDIBLE]. This is one interaction pair?

PROFESSOR: That's right.

AUDIENCE: And you take the product over all the interaction pairs within one run of the experiment. Is that correct?

PROFESSOR: If I want to determine whether a particular interaction pair-- I want to compute this log likelihood ratio, or this, actually, ranking ratio, because I've thrown away the priors. I want to compute this ranking ratio for a particular pair. So I've got protein A

and protein B. And I want to determine whether I believe it to be more likely to interact or not, and rank it with all the others, right? So I'm doing this for a pair of proteins now. So far so good?

Now, for that pair of proteins, I have a series of observations, or lack of observations, right? I have a whole bunch of experiments. This experiment detected it, that experiment didn't detect it, this one did. So what's the probability of these proteins-- these A and B really interact given that yes, no, yes in my experiments? And then for new protein, it might be no, no, yes, and what I want to figure out the probability for this pair.

AUDIENCE: So is the scale of the big letter M, is it on the order of like 10 experiments, 100 experiments, or thousands of experiments?

PROFESSOR: Ah. So the question is, what's the scale of this. So obviously, that's going to depend on what kind of data I bring in, but in these cases, it's small. So we have a handful of these high throughput experiments over entire genomes and proteomes. So there's not to be a lot. So in some of these early papers, there were four interaction experiments that they were looking at. Now the numbers might be a little bit bigger, but not significantly greater.

All right. So now to compute this, we need a set of gold standards. But now we don't just need gold standard positive interactions, proteins that we know really do interact. We also need proteins that we know really don't interact. Because I want to compute the probability of an observation given that some interaction is definitely wrong.

So precisely how I compute these terms is going to depend on the kinds of data. The experiments I've just been talking about, these high throughput mass spec, which were the ones which we looked at the ratio of the consensus, true positives, and estimated that 96% of all the data were possibly in error. The details of how to do those calculations are here. I leave you to look that up if you're interested.

But now what we're going to do is we're going to see how, if we were to rank

interactions based on this term, we can avoid having to throw out most of our data. So we said if we require all the experiments to agree, we're going to have very, very low coverage. Now we're instead going to rank everything based on this likelihood ratio, or something derived from the likelihood ratio.

So in this paper where they were simply looking at the protein-protein interaction data sets to compute these interactions, they ranked everything based on that ranking function we just described. And then as you vary your threshold, you can figure out how many true positives you have and how many false positives you have in the gold standard. True interactors and false interactors. And you can compute this curve, right? For any particular value of that ranking ratio, what's my sensitivity and what's my specificity? Are you clear what this plot means?

And here they've plotted the values for individual experiments. And this is the value for an independent database of gold standard interactions. And so now, where do they come up with their true positives and their false positives? A lot of this is going to depend on how representative those are. And all these numbers are subject to revision if you decide that the true positives and false positives that people are using are not accurate enough.

So they used two well annotated databases of interactions. One from MIPS and one from SGD. And you can play those off against each other as the database of true positives. In some ways, that's the easier thing because people like to report that proteins interact. They tend not to like to report the proteins don't interact. You don't see a lot of nature papers saying protein x doesn't interact with protein y.

So how are you going to figure out, then, what are your true negatives? So the strategies that they used-- well, one possibility is they're annotated to be in complexes, and those complexes are different from each other. That's not bad, right? But it's not a guarantee either.

Or this is a little bit better. They're annotated to be in different parts of the cell. Of course, if those annotations aren't perfect, low concentrations, you could still be wrong. Or that they have anti-correlated gene expression. I kind of like this one. So

it's one thing to be not correlated, but if you're anti-correlated, seems pretty suggestive that these two proteins are never in a complex together.

Again, it's no guarantee because, as we'll talk about in some detail later, RNA levels are not very good predictors of protein levels. But if you apply enough of these criteria, you can come up with a set of proteins that you have fairly high confidence really don't interact. You combine that with the databases of proteins with very high confidence that they do interact, and you can get the true positives and false positives that you need for this analysis.

all right. So that's a way of combining some information. We're going to see a generalization of that called Bayesian networks. We've mentioned this already in at least two different contexts, and it'll come up again later in the course as well.

So these are very general methods for reasoning probabilistically. We will see them in the context here of predicting interactions. We'll see them later in the context of gene regulation and signaling as well.

What we fundamentally need to do a Bayesian network is a graphical structure that represents our understanding what the relationship is between causes and effects. And a set of probabilities that allow us to compute things on this network. We'll show you examples where those networks are derived from our prior understanding of the problem, but also ones where the structure of the network is learned from the data.

And we're going to see two primary contexts. First we have this question of whether proteins interact. That's what we've just been talking about. So here are four experiments, the in vitro pulldown experiments and yeast two-hybrid experiments, that give us relatively independent information about whether proteins interact. And we're going to look at a paper that used those data with a Bayesian network to compute the probability that two proteins really do interact based on the combination of all the data, rather than throwing out anything that doesn't fall in the overlap, which could be a very, very small number.

And then later on we'll see examples of using Bayesian networks to understand biological networks. So this might be a set of transcription factors that are regulating a set of differentially expressed genes. And the structure of the graphical network for a Bayesian network has a lot of similarities to the way we normally think about transcriptional regulatory networks. So there's sort of a natural way of transferring our regulatory problem into a graphical network problem.

But we're going to focus on these prediction problems for protein-protein interactions first. Now, if I just want to compute the probability of detecting an interaction in various experiments, given that it's true or false, I could explicitly compute that probability. And we saw examples of that just now.

But some of these Bayesian network problems become much, much too large to do that. This is a little tiny piece of a Bayesian network that is supposed to represent I believe it's transcriptional regulatory network. You could never possibly write down all of the terms in this probability, where every node could, in principle depend on every other node in the network. It would just be a ridiculously large problem.

In fact, how large would it be if I've got N binary variables, my gene is on or off, my interaction is true or false, I have 2 to the N possible states? Right? And the only constraint I have, in principle, is that all the probabilities have to add up to one. So I have 2 to the N minus 1. 2 to the N minus 1 possible variables that I need to set. So that's a ridiculously large number in most contexts.

So how do Bayesian networks help us solve this problem? Well, we represent our understanding of the problem in a graphical structure where we have causes and effects. And there'll be a direct arrow from a cause to an effect. I don't always know the cause. So in our context, we were trying to figure out whether two proteins interact. What do we measure?

We actually don't measure interactions. We measure the result of a particular experiment, which is a combination of whether interacted and all sorts of noise that we've just discussed. So the effects that we observe are detected in experiment one or detected in experiment two. The cause is, did it interact or not? So the cause is

hidden, the effects are observed.

Now, in the case we were looking at before, we treated all these probabilities as being independent. But we might know something about the structure of our experiments, the kinds of experiments we're doing, that might lead us to have a different structure. So we could have an interaction that gives rise to all different kinds of data.

But depending on whether the protein's a membrane protein or highly expressed, it might influence the results of certain experiments and not influence the results of others, right? So like a two-hybrid would be very biased by which one of these? The membrane, right? And then the affinity capture mass spec could be very influenced by proteins that are expressed at very high levels or very low levels.

If we assume that all the interactions are independent, then we multiply probabilities. And we'll go into more detail, but this is what we're looking at up until now. In cases where we believe that all the observations are not independent, then we're not going to simply multiply things. We'll see there's a more precise way of computing the probabilities.

Now in this case, I've drawn the graphical structure because I believe that I know what's going on. But in the more general case that we'll look at, we'll actually derive the structure from the data.

One of the nice things about Bayesian networks is that it removes the need to have all 2 to the N minus 1 possible parameters, because it tells us there are certain independence conditions. So node is independent of its ancestors given its parents. What does that mean?

If I'm trying to reason about the expression of one of the genes down here, and I know that this transcription factor is on, I don't really care what the probability is that any particular parent of that transcription factor is on, right? So I don't need to know anything of transcription factor B1 if I know the state of B2. If this is on, then that's the only thing that's going to affect whether it's turning on these genes, regardless

of what the activation state of its parent was. Is that clear? Yes.

AUDIENCE: The slide's saying TF B1. [INAUDIBLE] TF B2? It says TF A1.

PROFESSOR: Yeah, sorry. That should say TF B1. Thank you. OK. So we'll do a little example. It's admission season both for graduate school and undergraduate. So let's do a little toy example where we're going to get rid of the admissions committees and just do automated admissions.

So we're going to collect various data about students, and then we're going to build a Bayesian network. And that network is going to decide whether to admit students into this simplified version. And the only information that will go into our decision will be the grades on the transcript and the GREs. Hopefully that's not the case.

And we believe that certain things influenced your grades and your GREs. Whether or not the student is smart certainly should have some influence, but also the great inflation at their school will have some influence.

So a prediction problem in a Bayesian network is going from the causes to the effects. So if I want to predict whether a student's admitted, I only need to look upstream. So we want to predict-- we observe the things on the top. Say, grades and GREs, and we want to predict whether this student should be admitted or not.

There's another problem called an inference problem, which is when we observe the effect and we want to make inferences about the causes. So an example of that would be, you apply for an internship and they say, oh, she's a student at MIT. I bet she's smart. Right? They're doing an inference problem.

We'll leave it for you to decide whether you and your colleagues are as smart as everyone thinks, but hopefully you are. OK. So we've got these two different kinds of problems. We've got prediction problems from top to bottom, and inference problems from bottom to top.

And we're going to talk about conditional probability. So if I've got some very small piece of this network with just two nodes, I could write out all the possible

probabilities for any pair of those nodes. So the probability that a student is not smart given that that student has low grades, the probability that the student is not smart given that the student has good grades, and so on, for all possible pairwise comparisons.

Or I could write this as a conditional probability, which tends to be an easier way to think about the problem. What's the conditional probability of a student being smart given that they've got good grades or given that they have bad grades? They have the same information. For this one, I need additional information about the total probability of students being smart or not.

And the total number of variables, as I said, in either case is the same. So these are completely interchangeable, but it's a lot easier to reason with conditional probabilities than with the joint probability tables. Those we'll see in a second.

So as I've said, you don't need a full probability table for a Bayesian network. You don't need two N to the minus 1 variables. And the fundamental reason for that is that the joint probability is only going to depend on the parents. So in this toy example, the GRE scores over here are not dependent on grade inflation.

Now, that all hopefully makes sense. Questions? Bayesian networks get a little murky next, so I'm going to try to give you into-- oh, yes. Question, please.

AUDIENCE: You said that the parents don't affect their children, but if grade inflation affects the grades, how does that influence-- will that influence the grade [INAUDIBLE]?

PROFESSOR: Sorry, can you say the question again?

AUDIENCE: I guess I'm just confused by this particular example. What do you mean by the joint probability? The joint probability of what?

PROFESSOR: So if I want to figure out the probability of some particular configuration of all the nodes in my network, I don't necessarily need to consider all possibilities. Because for example, if I want to consider all of the joint probability samples with settings for the GREs, whether the student had good GRE scores or not, that's not going be

influenced by the student's school's grade inflation policies.

AUDIENCE: But wouldn't the grades be influenced by the--

PROFESSOR: But the grades would be. That's right. So some of the variables I can remove and others-- some of the joint probability statements I don't need to worry about and others I do. And which ones I need to consider is determined by the graph structure. Yes.

AUDIENCE: How is the graph structure determined?

PROFESSOR: OK. So how is the graph structure determined? So it's determined in one of two ways. I can draw it in advance because I believe that I know something about my setting, I believe that these data are independent. Then it has that structure like this. Cause and a bunch of independent effects.

Or perhaps I claim to know that actually two of these things have a common parent as well. In some cases I know. We'll also talk about how to learn the structure from the data, which is the more common setting in regulatory networks. So in these kinds of problems when trying to decide how to integrate different proteomic data sets, typically people make arbitrary decisions about what the structure is based on their knowledge of the system.

But if you're trying to figure out de novo which proteins interact with which, which proteins regulate which genes, then you have to learn it from the data. And we'll talk about how to do that in a second. Great questions. Any other questions? Anything in the quiet half of the room?

OK. So as I said, this part of it, I think you can usually come up with cases that give you fairly good intuition. One of the things that is true in these Bayesian networks which most people find a little bit surprising at first is something called explaining away. So let's look at this Bayesian network.

I go outside and I detect that things are slippery on the grass. So that could be for a lot of reasons, but one possible reason is that the grass is wet. OK. What are the

causes of the grass being wet? Well, it could have rained or the sprinklers might have been on.

And depending on this as an example-- so a lot of the Bayesian networks were developed in Stanford by Judea Pearl and colleagues. And of course, in California it doesn't rain that often. So there the season is a strong determiner of these things. Not so much around here.

So in this example that they like to do, so does the probability that it's raining depend on whether the sprinkler is on or not? Now, the answer should be no, right? I mean, in reality, when you think about-- there's no causal relationship between the sprinkler being on and the rain. But in fact, when we're reasoning over these networks, we actually are influenced.

In a probabilistic model, if I know that it's raining, and I know the grass is wet, then what do I think about the sprinkler being on? Do I think it's just as likely? No, I think it's less likely, right? If I go outside and see the grass is wet, there are clouds, the rain is coming down, is the sprinkler likely to be on or not? It's likely to be off, right?

So there's no causal relationship, but there's the probabilistic relationship through the graph structure. And that's called explaining away. And you can take a whole course on how to understand which relationships you can detect and which not. This is not the place to try to go into that, but I hope you'll be familiar with this problem. And I'll try to give you a toy example that makes it a little bit more obvious in terms of the equations where this comes from.

So imagine this very silly game where we play, we toss coins. We toss a coin twice. And if it turns up heads both times, you get a point. If it turns up tails both times, you get a point. But if one's a head and one's a tail, you don't get any points.

Now, does the probability that I tossed a head on the first time depend on whether I toss a tail on the second time? So causally, obviously not, right? First of all, it happened earlier in time. And secondly, the coin tosses are completely independent.

But what happens when I know the outcome? What if I know what score you got? So if I know your score, then is the probability that I tossed the heads on the first time independent of whether I got a tail on the second time? What do you think? How many people think it is independent then?

How many people think it's not independent. Very good. It's not independent. And obviously, here's the math to prove it, but your intuition does the same thing. So what's the probability that I tossed a head on the second time given that I got a one, I scored, and I tossed a tail on the first time? Obviously, it's zero, right?

So here's the probability of getting a head in the first time and scoring one, and tails on the second time is exactly zero. So that's called explaining away. You can reduce your belief in certain parents based on what you know about the children. Think of this coin toss example or the rain in California and the sprinklers.

All right. So as this come up several times, how do we obtain the Bayesian network structure? There are two problems that we need to be able to solve. We need to be able to learn the structure, and we need to be able to learn these probability tables.

If we know structure, how do we get the probabilities? Well, we need to identify some objective function we're going to try to optimize, and then choose values for all probability distributions that optimize that objective function. And that's the kind of thing we've been doing all along, just like in the Gibbs sampler. We need some objective function or protein structure. We need some objective function that we're going to try to optimize.

So there are two common ones that are used a lot. There's maximum likelihood and the maximum posterior. So maximum likelihood is defined as the set of param-- theta is all the parameters, all the probability distributions, the probability of getting a score of one given that you had heads and tails, whatever it may be. The probability of getting admitted given that you had certain GREs and certain grades.

So we want to find the set of parameters, all those probability distributions, that maximize this. The probability of the data, our training data, given those

parameters. That's a pretty obvious one.

And the maximum posterior includes some of our beliefs about the prior probability of the data and the prior probability of the parameters. This is a little bit less intuitive because you have to ask, well, where do those numbers come from? And that, again, is a whole course unto itself.

OK. Now, how do you find these parameters? Again, it's the kinds of search problems that we've looked at before, various kinds of hill climbing. So gradient descent, expectation maximization, Gibbs sampling, which you've looked at explicitly. And again, the full details of how to do that are outside of our scope today.

OK. So in our example of this coin toss game, we would use one of these two functions to try to decide what's the probability of getting heads or tails for any given score. That's what the kinds of parameters are.

Now, the structure problem actually turns out to be really, really hard, because there are a very exponentially large number of potential structures to draw from. And unless you've got some prior knowledge, it can be impossible, depending on how much data you have, to actually build this structure.

So there are many algorithms that have been proposed. And a lot of our settings, we're going to use some kind of prior knowledge to reduce the search space. So if we're trying to talk about transcriptional regulatory networks, it's very common to assume that there are only some kinds of nodes that can be causes and other kinds of nodes that can be effects, right?

So in gene expression it would be effect, and then you would limit your causes to only be transcription factors. It would generally be signaling molecules or something like that, and not allow all 20,000 genes to be causes and all 20,000 genes to be effects.

So there are lot of resources to learn more about Bayesian networks. As I said, you can have whole courses on this. I think there are a lot of good tutorials at this website. I've also put in the notes a little toy example for you to work through all the

probabilities, which I think, in the interest of time, we won't go through in detail.

All right. So to motivate what we're going to do in the next lecture, I just want to talk about other kinds of data that you could bring to bear on this problem of predicting which proteins interact. We'll see, then, how that gets fed into an interaction Bayesian network to make the predictions.

So we've talked about affinity capture and two-hybrid, but what other kinds of data could we use to predict the probability interaction? Well, one thing you could use would be gene expression data. And the idea is that if two proteins interact, they should be present in the cell at the same time, right?

So we talked about this a little bit. If they're anti-correlated, it seems very unlikely they interact. What about if they're correlated, but not perfectly correlated? So here's a plot that shows a histogram of proteins that are known to interact, proteins that are known not to interact. So empty circles are known interacting proteins, the dark circles are non-interacting proteins, and the other ones are based on the experimental data.

And the distance here is the difference between expression profiles. And we'll talk in coming lecture about exactly how to compute distance between expression profiles. But the further to the right it is, the less similar the expression profiles are across large data sets. So what you see is the interacting proteins tend to be shifted more to the left, more similar expression profiles than the non-interacting ones.

But what do you notice about this? There's no way to draw a line and say, everything to the right of this is in one class and everything to the left is another, right? So by itself, it's not going to get us very far. There are plenty of non-interacting proteins that have very highly correlated gene expression and plenty of interacting proteins that have poorly correlated gene expression. So it's a trend, not a rule.

Now, what about evolution? So if I look over many, many organisms, I might expect what? The proteins that interact with each other are going to appear in the same

species, right? So let's look at these two cases. We've got a bunch of-- eight different genomes. And I've got gene 1 and gene 2, which I suspect might interact, and gene 3 and gene 4, which I suspect might interact.

Now, looking at these two patterns of evolution, which one do we have more confidence in that it interacts? The red one or the green one? So what do we notice about the difference between them? What's true of the red one compared to the green one? Yeah.

AUDIENCE: The red one is only in one branch of the tree.

PROFESSOR: The red one is only one branch in the tree and the green one is scattered across. So let's take a vote. Do we believe that the red one is better evidence of interaction or the green one is better evidence of interaction? Red? Green? Can I have an advocate of green. Someone explain their rationale? Anyone in the quiet side of the room? All right, Ed.

AUDIENCE: Because red is only on one branch of the tree, I'd expect that they're naturally more correlated with each other. They have less-- they appear together in [INAUDIBLE] so I'd expect [INAUDIBLE].

PROFESSOR: OK. So the argument is that red only occurs in one part of the tree. And so there could be a very simple explanation for all the reds being in one part of the tree and one not, which would be a single loss and gain event. Right? Somewhere early on, perhaps here, I gain those two proteins. And then they're inherited throughout the genome, like most of genes get inherited throughout the genome.

Whereas here, we've got independent events of gain and loss. And at each one of these independent events, we're getting them moving jointly, either in or out of the genome. So there's more evidence for green to be interacting than red. Everyone buy that? Even some of the advocates of red? Questions? Yes.

AUDIENCE: Could there be a way of either objectively or mathematically [INAUDIBLE] that way, or is it just the reasoning [INAUDIBLE]?

PROFESSOR: One can do the statistics on it with known ones, right? I think that's probably the best way. And we'll actually see that in one of these papers that uses-- well, actually, now I don't recall whether they use this co-evolution. But yeah, there are plenty of papers that actually have done the statistics on that. So it is supported.

And a related kind of question is what's called the Rosetta Stone approach. Unfortunately, of the term Rosetta gets used far too much in computational biology. So this has nothing to do with the other Rosetta that we've been talking about. And this has to do with how often you find the same pair of genes in the same genome versus split up in different genomes. OK.

So what we're going to look at next time then is an approach that combines these kinds of data with the protein interaction physical measurements through the two-hybrid and the affinity capture mass spec that actually uses the Bayesian networks we talked about this time to predict whether two proteins are likely to interact based on all of the available data. These evolutionary arguments, the [? sentiality ?] arguments, and then the interaction data. Any final questions? OK, see you next time.