20.320, notes for 12/11

Tuesday, December 11, 2012
9:41 AM

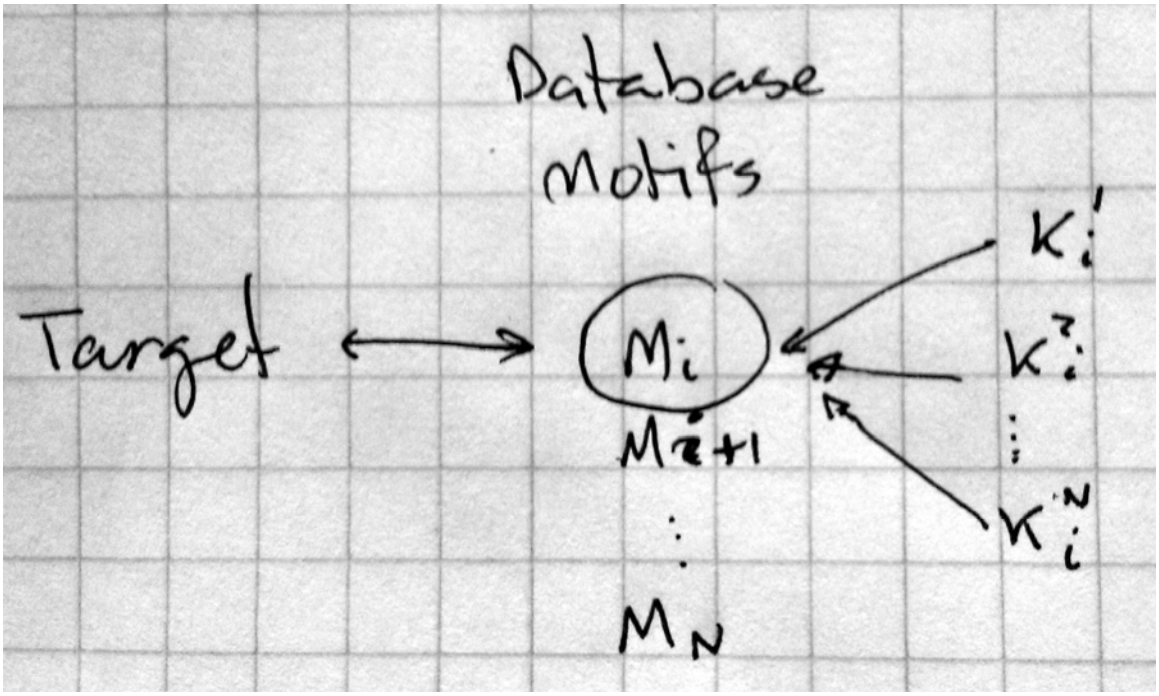
Last class

We talked about the most successful ways of predicting protein structure, homology modification with refinement using PyRosetta. This strategy, turns out, can also be used to predict protein interactions.

Kinases

Kinases are a great model protein. How do we predict what their targets will look like? One way to find out is through multiple sequence alignment. Start with a handful of known target sequences, then from that build a probability matrix (sometimes called a motif). You might wonder how big the target ought to be defined; one way of doing that is to stop once the information content of the next base is close to zero (the amino acids there are pretty close to random, indicating that they are not conserved). Remember to add pseudocounts to give a non-zero probability to amino acids that didn't show up in the known sequences.

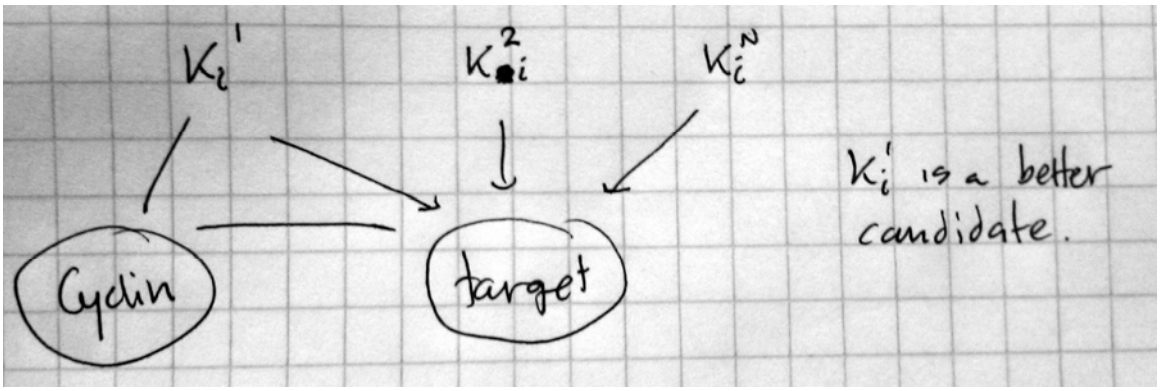
So starting from motifs, we scan the proteome for known motifs and we assign them an odds ratio. There are plenty of websites that can identify known motifs from any protein structure you might have. It will find the motifs in your sequence, but the job is not done. The reason is that a given motif will *almost always* have more than one kinase that acts on it. What we attempt to do today, then, is to determine which of the candidate proteins is actually the real kinase that acts on that motif.



Thus, we need to talk again about what gives kinases their specificity. The active site is the greatest determinant of specificity, of course. All our candidate kinases share a similar active site, which is why they were flagged in the first place. There are also other specificity-determinant sites away from the active site. Finally, we must account for protein-protein interactions (SH2/SH3 specificity-granting domains, helper proteins like Cyclins, etc.).

One possibility is to search for interactions with other proteins (like cyclins). If a target sequenc has specificity not just for the kinase, but also for a cyclin that the kinase works with, then we have a good

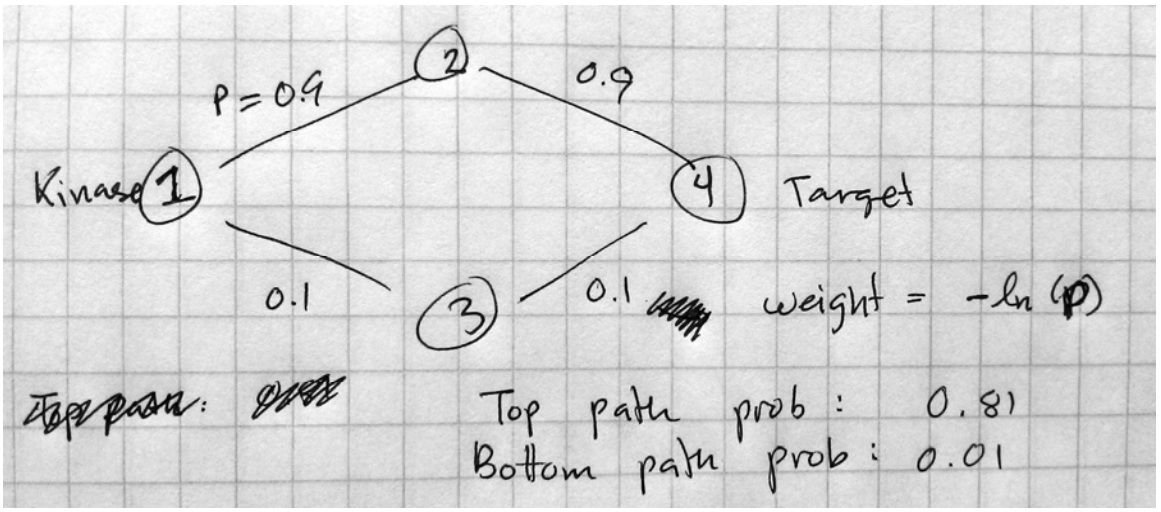
lead.

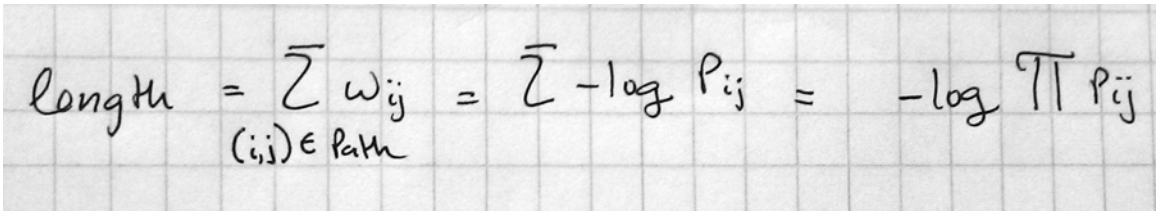


Graph theory

This is a very popular area in computer science, which has come in vogue due to the utility in modeling computer networks but also because it's useful in biology, where we deal with a lot of networks. It relies on representations of systems as graphs, which are ordered sets of nodes and vertices. Nodes are things, vertices are interactions between them. Paths are ways of getting from one node to another, going only through paths. If you use each mode once at most, it's called a simple path. Graph theory provides formal tools for mathematically analyzing large networks, and allows us to account for the confidence we might have in paths (based on the reliability of our experiments). What we are trying to do is determine the most likely path between kinase and target, the most likely actual interaction.

Remember that we are determining the existence of interactions by fallible in vitro experiments (SPR, FRET, Mass Spec). Based on our confidence in the experimental result and its applicability in vivo, we assign different probabilities (weights) to the vertices in our graphs. The weights may mean something different in other contexts, but this is an abstraction; therefore, all the same rules apply. For example, lots of algorithms calculate the shortest path between nodes by summing the weights of the intermediate vertices. Since we want the product of probabilities, not their sum, we define the weights to be the negative natural log of our probabilities.





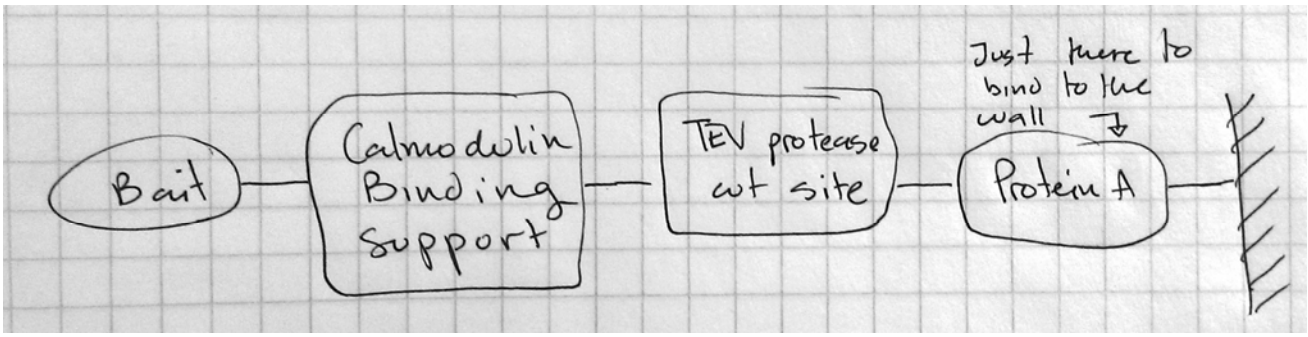
Note that we are not looking for the simplest path, only for the most probable. Determining this seems simple at this scale, but can become really difficult with complex networks. The computational algorithms that do that, however, are exceedingly good. That's why we use them.

Experimental Techniques

So we have this framework. How do we get the weights for our vertices? Let's look at some techniques, starting with Mass Spec.

1. Bind a protein to a surface. We'll call it bait.
2. Wash cell lysate over it, and bait will catch all sorts of things. Some will be specific interactions, some will be non-specific.
3. Use Mass-spec to determine which proteins are stuck there.

Turns out that the non-specific interactions are a huge problem, though. We can't get rid of them, and they give us all sorts of MS red herrings. Some proteins are just sticky, and will stick to everything. One way to solve this is with a **TAP-tag**, where TAP stands for Tandem Affinity Purification.



1. Start with the above combination of proteins.
2. Wash with lysate
3. Use TEV protease to cut everything away from the wall, where most of the non-specific interactions will remain
    a. Your stuff is now in solution
4. Use fixed calmodulin to capture these complexes again elsewhere
    a. Your stuff is now fixed
    b. This interaction requires Calcium
5. Use EGTA, a mild calcium chelator, to sequester the calcium and release the CBD, along with your Bait and all the stuff stuck to it.

The net result of all this convoluted capture is that we dramatically reduce the number of non-specific protein-protein interactions that we're capturing in Mass spec. If the above list of steps was not clear (and I don't think it is), then I suggest more reading elsewhere.
http://en.wikipedia.org/wiki/Tandem_Affinity_Purification

Computational Techniques

You have two proteins and PyRosetta. How do we get the ΔG of binding? Say we start with the crystal structures of the two proteins. We might try to start with our Docking method, moving the proteins around each other and looking for a good docking site. This is an extreme method. This would not work, because proteins undergo significant conformational changes when binding to each other, and there's just too many ways that could change. The combinatorial search is astronomical, and completely unreasonable. This method is slow.

At the other end of the spectrum is our lesson from CASP: search for homology. Are there other proteins that interact with each other and are similar to this? The problem is that this has very low coverage. You need to have seen this before, if you're going to make a prediction. This method is limited.

This is where the field stood for a while, but people realized after a while that for homology searches you don't really need the whole protein to be the same. Interfaces are classified thus:

Type I clusters	Similar interface architectures	Similar global folds
Type II clusters	Similar interface architectures	Different global folds
Type III clusters	Interface architecture similar only on one side	Different global folds

Thus, even when global folds are very different there can still be structurally conserved binding regions. We've identified interaction **hotspots**, which are conserved between very different proteins. And so, people have made databases of the *interfaces*, and matching algorithms that decide whether a query matches the known interfaces and avoids steric hindrances from elsewhere in the protein.

So what now?

What have we accomplished during this second half of the class? Starting from molecular structure, we can calculate energetics for a protein. We can predict the effects of mutations, design altered specificity, and we understand how these tools could be used to make large-scale predictions. The key to all this has been to take an insurmountable problem and divide it into smaller, more manageable problems. Our tools cover both the details within a single protein and vast networks of interacting proteins.

We've coupled analysis and design. Such is the gist of biological engineering. Remove either, and you are no longer doing engineering. You're either doing just science or taking stabs in the dark. You are now capable of choosing the right level of abstraction to analyze a problem, and you realize the implications of all of them. The pep talk was longer, but that's as much as I got. Good luck with the final!

20.320 Analysis of Biomolecular and Cellular Systems
Fall 2012