PROFESSOR: So in terms of what we're going to be discussing today, it's various aspects of feed-forward loops. So first of all, we will go over this idea of a network motif in more detail. We talked it about a little bit in the context of auto regulation. There is a simple argument there that auto regulation is a network motif. And in order to understand how to detect network motifs, in general we have to look at a little more detail at these subgraphs and the frequency that they'll appear and so forth.

And then after seeing that the feed-forward loop is a network motif, then there's this question-- oh, well what might be the functional significance? And in the chapter that you just read, you found this so-called coherent type one feed-forward loop has a nice attribute-- that it's sign sensitive delay element. The incoherent type one has the feature that it's a pulse generator, and kind of related to that, it can also speed up the response time. So it can make the response time for turning on shorter. So speed up response rate if you'd like.

We'll also maybe say a little bit that there's been later work demonstrating that the incoherent type one can also access a fold detector. So it can sense changes in the fold change of concentrations of proteins. And then finally, we'll say something about how you can extend these ideas of a network motif to larger structures. In particular, how you get useful temporal programs.

So we start out with this network that is kind of-- our base network is this transcription network characterized in E. coli. And we've already talked about it some. So there's going to be some measure the number of nodes and the number of edges. So we have N nodes, and we have E edges. This is from experimental measurements of-- what is it that regulates what? Now from this, we'll have some set of directed edges, because indeed, we know that there's going to be some transcription factor that will regulates some other protein.

And what we want to know is are there patterns that occur more regularly than what you'd expect based on chance. Now auto regulation we found indeed appeared more regularly, or more frequently than you would expect by chance. And there are a limited number of other network motifs that have that property. And in particular, we'll kind of analyze this idea of the feed-forward loop. Now in the network that we kind of talked about a lot, there were around 400 genes or proteins, and then around 500 observed edges.

And this would be interactions or regulation. Now there's this idea of sparseness. Can somebody remind us maybe what this is supposed to tell us about? Yes.

AUDIENCE: There should be roughly N squared in total edges.

PROFESSOR: So there's N squared possible edges.

AUDIENCE: If you just connected everything that you connected, and so N is about the order of E. So roughly, there's only 1 out of 500--

PROFESSOR: And we should be clear, this N squared possible-- we might even want to add directed edges since we're-- and what we see is that the actual number of edges that we observe in this real network is actually around order N. So this sparseness, which is also this probability P-- if we're going to make a network somehow that has some similar property to the observed network, there's some probably P that an actual edge will appear.

And this is given by the observed number of edges divided by the total possible number of edges, which is indeed N squared. And what you see is that, at least in this network, this P is much less than one. So this is what we mean by sparse. Now you can also think about the question of how many edges does a typical gene have emanating from it? Well, you can see that it's around one.

Of course, each edge connects two things, so if you were to say on average, each gene has of one edge going out, and one edge going in. Of course, there might be a reason to believe that these averages are not as-- well they can be misleading. And why might the average be misleading? Yes.

AUDIENCE: The distribution for heavy tails.

PROFESSOR: Right, it's going to be a distribution with heavy tails, in particular on which side? So the average is always the average, but I guess the question is there are going to be some proteins, or some genes with many of these outgoing edges. And more generally, do you expect that-- there's a natural limitation in all this. Part of the value of this approach is that we're abstracting away from a lot of the microscopic or the biological details. But every now and then it's good to go in and think about it a little bit.

So there's a good reason you'd expect many proteins to not have any outgoing edges, and why would that be? Yes.

AUDIENCE: For example, a detector protein, that only evolved one specific function or another.

PROFESSOR: So there are some proteins that have rather specific functions, you might say. And I think that's true. I think that's part of it. But there's maybe something that's maybe even a little bit more general that's worth pointing out here. Yeah.

AUDIENCE: Anything that's not a transcription factor.

PROFESSOR: Anything that's not a transcription factor, right. We say transcription factor, we don't ever really quite specify. But what it means is that it's something that can affect the transcription of other things. So we're talking about the function of it when we say transcription factor. And a majority of the proteins in any genome are not transcription factors. What that means is that they, to first order, cannot, at least directly influence the transcription of other genes.

So this is a reflection of this power law distribution that we observe. And so you could argue well maybe not a surprise that this outgoing edge distribution is power law, because we know that there are some transcription factors that control many things, and there are many proteins they don't control the transcription of any other proteins directly.

But at least it's just useful to know what the properties of this thing are on average even though, for the outgoing edges, the average is a little bit dangerous. Because it's really that most proteins don't have any outgoing edges, and then some have many. All right so this is this probability P. And this is going to be useful, because this P will appear when we're trying to construct these random networks.

So what we're going to do is we're going to ask how frequently or how many of a given subgraph you expect to appear in this larger network that we have here? And we're going to characterize each of these subgraphs by two properties. And in particular, if we're going to analyze some smaller graph, we just have to keep track of how many nodes are in the subgraph, and how many edges are in the subgraph.

So for example, if we have auto regulation, then we're just talking about little n equal to one, little g equal to one. Whereas in the case of this feed-forward loop, what is little n and what's little g? Well we can count now. Here n is equal to three, one, two, three, and g is also equal to three. And we're going to find that actually the fact that these two numbers are equal is somehow very relevant in thinking about the dynamics of these networks later.

In this framework, when I just draw an arrow in the context of a generic subgraph, am I necessarily trying to say that this is up regulation of x up regulating y? No. So this is a bit confusing because depending on the context, sometimes the arrows actually do mean up regulation, sometimes they just mean regulate. And in this case, where we're talking about subgraphs, we're just saying that x regulates y in one way or another.

This is also how we're going to write the so-called coherent type one feed-forward loop, but for right now this is just a generic feed-forward loop. Yes.

AUDIENCE: Regulate means positive regulation?

PROFESSOR: Yes. We say activate. So the way that we think about is we ask, what's the expected number of some subgraph G? And indeed what we're going to be doing for right now is assuming this Erdos-Renyi random network. Now what we're told is that this is going to look something like the following.

could somebody explain one of these three terms? Yes.

AUDIENCE: Well you have to select edges, and you have choose them correctly. So for each one, there's a chance that [INAUDIBLE].

PROFESSOR: So for here what we're going to do is, for example in this context, we'll say, we're going to choose these three. And now the question is, we have to put in three edges as well to connect those nodes, and each one of them has some probability P of actually somehow appearing. Because this is what we're keeping constant from the original network.

So we're assuming that we have this Erdos-Renyi network with the same number, say roughly 400 nodes, and then what we're going to do is grab maybe three of them and ask, all right, what's the probability that we get these three actual edges? So you get P to the g. Now where does this term come from? Yes.

AUDIENCE: Selecting a node.

PROFESSOR: So we're going to be selecting N nodes. We're assuming that this N is much larger. We're assuming that we have a big network, so it's much larger than the size of the subgraph we're looking at, so we don't have to think about n times n minus one, n minus two, and so forth. And then there's this factor a here as well, which, depending on how you do your counting-- this is a little bit tricky, but what was a again? Yes.

AUDIENCE: It's the number of ways to arrange the edges. It's a symmetry factor.

PROFESSOR: Yes, it's a symmetry factor, it's a way of-- there are multiple ways of looking at this. You could think about it as the number of ways of rearranging x, y, and z and having the same subgraph, the exact same one. In this case, there's actually no way to rearrange x, y, and z to have the same one.

Because x occupies a special spot, y is indeed again a special spot. z is special. So there's no permutations that you can do to get the same thing. Whereas if you have this repressilator-- now here I'm just drawing this as an arrow, because again we're just thinking about the generic version of these things. So it's just when we have x, y, and z regulating each other. In this case, you can get the same network by rotating these x's, y's, and z's.

So in this case, you get a equal to three. For pretty much all the conclusions we're going to talk about, these factors of one, two, three don't actually end up being relevant. But it's good to know that they indeed exist. So this is fine, but it's useful to express this in another way. In particular, we can always define this lambda, which is E/N, as the mean number of incoming edges or the mean number of outgoing edges.

And with this, we can express this guy in a way that is surprisingly informative. Nothing happened except that we just plugged this thing in here. But by doing, we see something that's kind of interesting, which is that there's reason to believe that for many of these networks, this lambda, this mean number of incoming edges, that lambda will be roughly similar-- whether you're talking about a network that is 500 nodes large or 5,000 nodes large.

And indeed in this case, it's around one. It's just over one. So that means that when we think about the number of subgraphs that will be in this large network, it scales with the size of the overall network. We had this little n minus little g. And in particular, in cases when you're analyzing a subgraph with the same number of nodes as edges, then you just get n to the 0, and it doesn't scale with a number. So for those, and indeed basically for all those subgraphs where little n is equal to little g-- then you expect of order one of those-- if lambda is around one, then you expect of order one of those to appear in the network.

And so from a very simple standpoint, the networks, like the feed-forward loop that we see that is a network motif, the expectation is that in a random network, you would get around one, maybe two. Whereas if you see many of them, dozens, then it's indeed a network motif. Are there any questions about how that appeared in the chapter or the argument there?

So indeed, we can actually just then say, for the feed-forward loop, we can just go ahead and ask how many were observed in E. coli, this network that was actually observed? And this was 42 I believe. Whereas if you do this analysis for the Erdos-Renyi network, you get 1.7 plus or minus 1.3. Because these things appear randomly, they should be Poisson distributed.

So you expect of order one of them to appear in a random network with this same kind of sparseness, the same number of edges. But we actually observe this much larger number. So then you can say, all right, this is evidence for the feed-forwad loop being a network motif. That for some reason, this subgraph appears more frequently than what you'd expect based on genes.

Of course, we alluded to this on the end of class on Tuesday, that maybe this Erdos-Renyi network is not the proper null model or null network to be using. And maybe we should use one of these degree-preserving networks. So maybe we should try to preserve more the properties of the original network. And So can because somebody say a little bit of what we mean by degree-preserving?

There's an element that our null model here already preserves something about the degree. It preserves the means. So it's not just that we picked up some random null model, some random ER network. So what is it that we want to keep track of in this degree-preserving network? I saw a hand over there, but I'm not trying to call on people randomly. Although I'm going to start in the second half of the semester, just after drop date.

[LAUGHTER]

So we're going to preserve not only the mean of the incoming and outgoing edges, but also the actual degree distribution. In a very concrete way, we can actually just say that each node actually does maintain the exact same number of edges. So in particular, here we say that all nodes maintain their degree distribution or maintain number of incoming and outgoing.

And there was a simple algorithm for doing that. If you recall, what you can do is you can just take two edges randomly and just swap the locations. And you do that many, many times, and you end up maintaining both the incoming and the outgoing number of edges. And if you do this analysis on a degree-preserving random network, you get a seven plus or minus five. So this makes a big difference.

So if, for example, the experimentally-observed network had one of these feed-forward loops, then what you'd see is that actually, comparing to the Erdos-Renyi, you would have said, oh well that that's a network motif. Whereas comparing to this degree-preserving, you would have said it's not. Yes.

AUDIENCE: I don't know why I haven't asked this before, but is it also true that this degree-preserving thing is roughly Poisson in terms of-- I mean, should we expect deviations always?

PROFESSOR: Yeah, it's close. But I think it ends up not being quite. But it is close.

AUDIENCE: So is the entire transcription network for E. coli?

PROFESSOR: So I think that it is, certainly now-- well you'll notice that here there are 400 genes. How many genes does E. coli have, anybody? A few thousand. So the network that we analyzed in that original paper was not the full transcription network of E. coli.

AUDIENCE: How do people-- how do figure out the whole transcription network? It actually sounds pretty hard.

PROFESSOR: Well figuring out and any part of it is actually hard in some ways. The way that they actually annotated this particular network, I'm not sure. I mean what kind of date would you need in order to get at this? Does anybody have any-- I mean, if I asked you to do this in your lab, what would you do?

So you could actually use a computational program to try to actually estimate binding affinities of these proteins to the DNA. And these days actually experimentally you can actually just measure for the entire proteome basically, just the affinity of binding to different promoters. Of course that doesn't prove that it's going to regulate expression, but that at least points you in the right direction.

Then you could actually, if you want, you could just experimentally go and put this protein on an inducible promoter and just see if it does regulate expression. At this stage, for something like E. coli, we have collections of strains where every gene has been removed. We have collections where every gene has been tagged. And of course, when I say every, this means that it was tried to make it for every and then of course if it's an essential gene you can't remove it. And

In some cases it's hard to attack a fluorescent protein, and so forth. But there are collections both E. coli and for budding yeast where this has been done. Any other questions? Can somebody say why it might be-- is this a surprise that this number is larger than this number? And in particular, would the degree-preserving random network have a larger expectation for every subgraph? No.

But in particular, for the feed-forward loop, can hear somebody say why we should have expected that the degree-preserving would have a larger number than the-- yeah.

AUDIENCE: All the measures have one outgoing edge, and so [INAUDIBLE] distribution towards lower numbers of outgoing edges. And so you would expect more from forward loops.

PROFESSOR: I think that the explanation had the right flavor, but I think there were two inversions in there that-- like a not and a not turned into a-- Incidentally, this is a non-sequitur, but this happened to me once. The airport in San Francisco-- I was going to the airport, I got the wrong airline in my head. So I thought I was on the wrong airline, so I went to the wrong terminal, but then it turned out that I also was wrong about which airline was in which terminal, so then I was actually the right terminal even though I had just made two mistakes.

But you can't account on this happening all the time. But I think there were two things that were mixed up in that explanation. Yeah.

AUDIENCE: For a transcription factor, it has many outgoing-- outcoming edges. And you just have to--

PROFESSOR: So in the actual network, There are some nodes with many outgoing edges. And then--

AUDIENCE: You just need to have another line between.

PROFESSOR: That's right. You somehow just have to add one more edge. Because x actually has two outgoing edges. So there's a sense of the feed-forward loop here-- you can think about x being some transcription factor. And then what you need is just to get-- x might have many, many outgoing edges. And so to get a feed-forward loop, what you need is you need for one of those genes that are targeted to just target another one that's in that network. So if you have x that's regulating one00, then that actually that presents many opportunities to generate feed-forward loops. Yes.

AUDIENCE: At the same time z, it's just going in. So wouldn't that make the [INAUDIBLE]? Is that why there's more ingoing edges than outgoing edges?

PROFESSOR: Now this is a problem with verbal arguments-- you can construct anything. And indeed, I would say that this is an example. What we said is that the distribution of incoming edges is roughly kind of similar to an ER network in the sense that if the mean is one, then sometimes you get 0, sometimes one, sometimes two. And they're all kind of reasonable. So in that sense, I'd say this z node is not so unusual from the standpoint of degree-preserving network.

If z had one00 incoming edges, then it's certainly true what you're saying-- that the degree-preserving would then have fewer.

All right. So this is the basic argument for why you might go and look at what the function of the feed-forward loop might be. I just want to say a few things about this original paper that Uri published. So it's in *Science* in 2002.

"Network motifs: Simple Building Blocks of Complex Networks." All right. So the authors did indeed analyze both the E. coli and the yeast transcription network. But they also analyzed a number of other networks to look at these networks motifs. So they also analyzed neurons from C. elegans, the worm, where the connectome has been known for several decades now.

And again, they found that feed-forward loops appeared more frequently than what would be expected, based on the known model of degree preserving. And that's encouraging is that saying, oh maybe feed-forward loops really are somehow preserving some-- they're performing some useful information-processing task.

Of course, you always have to worry-- there's also the spatial arrangement. You can worry about a lot of things. But that's encouraging. He also analyzed food webs, where in that case feed-forward loops were not a network motif, but other things were, So that's interesting.

He analyzed the design of electronic circuits, a forward logic chip. I don't know that is. But then also the worldwide web, it's another network people love to analyze. And indeed, he saw some other patterns. And the idea is that in each of these contexts, the network motifs are different, depending upon the microscopic structure that's leading to it, or the function that it's maybe evolving towards, or whatnot. So it's a way of getting insight into the properties of these complex networks.

Are there any questions about these network-- the global network structures, before we get into the feed-forward loop in particular?

So first I want to just go ahead and do a few of our little concept questions. Just because you have the cards, and I think that the chapter is actually pretty nice in the sense of you can read it, and get a clear sense of what's going on.

But let's start by just considering this feed-forward loop, which is this coherent type 1. So now the arrows actually mean activating. So we have X going to Y. Now it's going to be going to a Z. But we have to remember that now that there are two inputs, we do have to specify how the inputs are going to be combined.

And for now what we'll do is we'll assume that it's an AND gate. And that goes to Z. As always, we're going to have to think. There's some signal x and signal y that come in here. In many, many cases, these transcription factors in addition to being regulated by another, say transcription factor, may also be responsive to some signal.

And there was a nice example of this in Uri's book which was how E. Coli decide whether to make the suite of proteins that are required to digest the carbon source arabinose, the sugar arabinose. But for now, let's just think about this. And we want to just make sure that we remember. And once again, it's not that you should necessarily memorize these things. But after having seen the argument once or twice, you should be able to reconstruct all of these things.

So I claimed that somewhere in here there's a sign sensitive delay. Now the question is, in which direction is there a delay. And so it's going to be some combination of on and off perhaps.

And check means that it's a delay in that direction. Well actually we should just from nothing there-- and D is don't know.

AUDIENCE: In that direction you mean?

PROFESSOR: That there's a-- this means that there's a sign. That's right. So this would be going from off to on, so turning on. So this is turning on, as compared to turning off. And we're looking at this is delay. We're talking about, this is in response to Sx changing concentration of Z.

Any questions about what I'm referring to in this?

AUDIENCE: Is there any Sy?

PROFESSOR: Yes. Good question I like that right Sy is present.

AUDIENCE: [INAUDIBLE]?

PROFESSOR: Right. So this is compared to simple regulation, or i.e. does the concentration of Z immediately start to change after this Sx changes? It goes from either 0 to 1, or 1 to 0.

AUDIENCE: So simple regulation in this case would be erase that line between X and Y?

PROFESSOR: Yes. And make the AND gate a--

AUDIENCE: Not an AND gate?

PROFESSOR: Not an AND gate, exactly, yeah. We're comparing to just if X is just directly regulating Z. Because what we want to know is, I mean what might a function of the feed-forward loop be. So I'll give you 15 seconds to think through this. Once again, it's not that you should have memorized it.

I don't want anybody saying that they just couldn't read my handwriting.

AUDIENCE: So this is for the second case we're asking?

PROFESSOR: I'm sorry. So this is-- I'm asking about for this feed-forward loop, the coherent type 1 with an AND gate. And I'm comparing, I'm asking is there a delay in either turning on or turning off, as compared to the simple regulation of X regulating Z. And of course, this is my [INAUDIBLE] Sx.

And we assume that X is already present. Do you need more time? All right, ready? Three, two, one. OK. We got a clear majority of the group actually is saying C. And so we can get at this kind of visually, graphically. I really like graphs. I think they're much nicer than equations. Different people can agree or disagree. but that's my--

The idea is that we have Sx. It starts out off and say turns on. So if we think about the X star, X was always around. So means that X star immediately-- so the signal immediately changes X into X star, the active version.

Now Y, and this is why we can even say Y star, because the signal Y is always there. It starts our here. It immediately gets the signal. So it starts coming up. But of course, this is an AND gate. Which means that you need to have both active Y and active X in order to start getting expression of Z.

So we have if we look at Z, there's something threshold at which is Y starts allowing for expression of Z. So just because we have active X, doesn't mean that we immediately start getting expression of Z. We need Y as well. So this comes up.

However, when the signal here goes away, X star immediately goes away. This is the separation of timescale idea. This is just a binding of a small molecule or so. What that means is that Y star-- is there a delay on Y star?

We're going to do a verbal. Is there a delay before Y star just starts coming down? So the question, it's going to decay exponentially once it starts going. Does it a start going immediately, or is there a delay? So the question is, is there a delay before the exponential fall off of Y star. You're going to say yes or no, ready, three, two, one.

AUDIENCE: No.

PROFESSOR: No delay. Great. And indeed over here it's the same thing, no delay because of the AND gate. Expression of Z requires both X star and Y star. So although Y star still there, since it's an AND gate, Z goes down.

That means that in this case we have a sign sensitive delay for turning on Z, but not for turning off Z. And of course, this can be useful, depending upon the costs and benefits of having false positives and false negatives in the signal.

If this AND gate were switched to an OR gate, how does this thing change? I'm going to give you 10 seconds. All right. So, question is, if I convert this to an OR gate, does it change anything or not. Do you need more time? Ready, three two one.

Now we got a lot of B's. Great. So in this case it's coherent type 1, feed-forward loop with an OR gate. I'm not going to go over the logic. But I encourage you, if you're confused by this, just make sure that you can reconstruct the argument.

From my standpoint, these equations I mean, it's good to do equations. But it's more important to be able to understand the logic here. Did you have a question?

AUDIENCE: Yes. So how is it easy to prove experimentally what kind of gate there is?

PROFESSOR: Yes. So the idea is that in many cases you can put X on an inducible promoter. You could put Y on an inducible promoter. So you can-- just some small molecule will allow you to control these. And then you can measure, say fluorescence, on Z. And that's the most direct things. It's experimentally doing it yourself.

Of course, much of the data that you see the chapter is kind of just looking at the fluorescent Z as a function of the signals that you put in. And that's certainly an argument for it. And then ultimately what you'd like is to measure things in multiple different ways, confirm that it's all consistent.

Any other questions about this idea of sign sensitive delay element? Yeah?

AUDIENCE: Sorry. So among these 42-- these that are this type, is it possible to look at the actual genes, and see if that interrelationship actually makes sense?

PROFESSOR: That's a good question. So if you look at across both E. coli and yeast, what you see is that of the feed-forward loops, about half of them are coherent type 1, which is one of the eight possible kinds of feed-forward loops. And so what you're asking is, in this case, so let's say there are 20 coherent type 1, how many of them, what fraction of them does this all makes sense? And it's a good question. I don't know.

I haven't looked at it. Because it's always dangerous, of course, that we find one example of the 20 where it kind of makes sense conceptually. And then we go and we test it experimentally, and see that it all works. And then we're convinced. But you're pointing out that maybe we shouldn't be convinced yet.

AUDIENCE: But I'm just curious. If you're proposing a functional kind of explanation, and if know what the genes are.

PROFESSOR: That's right. You should be able to go and see whether it somehow make sense. And of course makes sense is always a slippery concept. Because we can always-- it's not that this radically changes the logic. And then in any given circumstance you may be say, oh well. You can kind of wave your arms and make up a story where it kind of makes sense. But then the only way to really feel comfortable with it or not, is for you yourself to go on look at them, and see how comfortable you are with each of those arguments. And I haven't actually done that.

So one more question in this regard. All right. So let's imagine that instead of thinking about changes in Sx, with Sy present, let's now flip things. Let's assume that Sx is present and ask about Sy turning on and off.

Again with the AND gate, I want to know in which direction is there a delay when turning either on or off. Now we're talking about with Sy turning on or off. Does everybody understand the question? I'll give you 10 seconds to make sure.

Do need more time? Let's go ahead and vote. Ready, three, two, one. All right. OK, so I'd say now it's pretty overwhelming that the group again agrees that now it'll be A. So if we have Sx equal to 1, and Sy is changing, then in this case we don't get any of these delays. So there's sort of immediate changes in Z as Sy changes. And that's because this is an AND gate. If X is already there that means that we've already satisfied this half of it. So then we're just reduced to simple regulation. This is just equivalent to Y regulating Z. So there are no delays either turning on or turning off.

So what you read about from Uri's book is what is that the coherent type 1 is perhaps the most common of the feed-forward loops observed in these transportation networks. The other of the feed-forward loops that is distinctly overrepresented is this incoherent type 1. So it's very similar, with the exception that now what we have is X activating Y, but now Y is going to be repressing Z. And we have X again activating Z.

And we're going to use an AND gate again. Its edge going to Z. For me, I find it sometimes a little bit confusing to think about a repression and an AND gate. So it is useful to make sure that we kind of understand the logic of all these things. So if we have say X star, Y star, and we can just make sure this is absence or presence, digital approximation of each of these things.

And the question is, if we have expression of Z. Now the way to just think about this is that this guy is equivalent to kind of inverting the sign of Y star. And then we have an AND gate. So this is a 0,1. That's not an AND. Well 0 is enough to give us a 0. So here we get activation. Here we don't.

And so we can do a similar kind of story of what we did here. Except that now instead of Y being an activator, it's now a repressor. And again, we're going to think about what happens with the signal coming in.

Is any difference up to this point? Let's think about it. Everything but this for a second. All right. So this is a case where we already have signal Y that's allowing, say, the Y repressor to bind. Then we make Sx appear. I want to know verbally, yes or no. Do I have to draw something new up to this point here? Ready, so what do I want to say? Is there a change from this drawing up to this point? Ready, three, two, one.

AUDIENCE: Yes.

PROFESSOR: Yes. All right. And that's because actually Z starts coming up at this point. It's very, very nerve-wracking, these quizzes, I know.

The idea is that here Y is now a repressor. So it's not that you need Y in order to get expression of Z. Is that once you have Y star, then you stop getting expression of Z. So it looks like maybe I'll make a-- so in this case everything's the same here. Except that in this case you start getting Z coming up. And then once Y gets up to a sufficiently high level, it starts repressing. In that case it might do something like this.

Now, depending upon the strength of that repression, this curve might look different. Because it could come all the way down to 0. Depending on if it's a very effective repressor. And depending upon whether it's fully repressed or only partially repressed, you might think about it as either being a pulse generator, so you get some Z, and then it goes away. Or you could think about it as a way of increasing the rate at which you're able to turn this gene on.

Because it's sort of like this negative autoregulation idea that initially you get lots of expression. And then later you stop getting as much. Of course, here you would get an overshoot. But maybe that's not all bad.

So in this what you might say is this is a pulse generator. And this here is a way of making t on go down. Are there any questions about the logic of what happens in this incoherent type 1? Yes?

AUDIENCE: Can you explain the t on?

PROFESSOR: Sure. So maybe let's zoom in. And we can look at Z. So it kind of gets expressed. And then it represses like this. And then you always have to ask, well what do you mean by t on? And then we have is working definition, which is that we say t on is defined as there's some equilibrium concentration. So this is Z equilibrium. And we often define t on as the time in which you get half of that concentration.

So what we do is we take half of that. And we say, where does this cross? It crosses right here. I maybe overdid my drawing. It's too good of a-- So t on, in this case, would be that time right there.

Now of course you have to say, well what should we compare that to? And the comparison should be the t on that you would have had with just simple regulation. So that's based on the generation time. And you can actually see what the generation time is here. Because this thing in the absence of the repression would have done something like that. So this tells us that the simple regulation would have lead to something that looks like this.

So you see the t on here, this is t on simple, is much larger than the t on that you actually get here.

So from this argument there are-- we can now try to recapitulate or recall for ourselves the various strategies for increasing the rate of response to signals. So we have, we can think about, you want to decrease t on, and you want to decrease t off. And we want to think about maybe different strategies that we've encountered over the last few weeks. What were some of the strategies?

Yes?

AUDIENCE: Just having a higher degradation?

PROFESSOR: All right, higher degradation. Right. So increase degradation, well maybe I'll just say decrease lifetime. Well it's the same thing. So decrease the protein lifetime. And what does that do for us? Does that decrease t on, or does it decrease t off? Both.

All right. So this decreases t on, and it decreases t off. What were some of the other strategies? Yes?

AUDIENCE: Negative autoregulation.

PROFESSOR: Negative autoregulation. Great. And what does that do for us?

AUDIENCE: I think it decreases t off and on also.

PROFESSOR: All right. So let's do a vote. This is a good opportunity. This is like review for exams for you guys right now. All right. So negative autoregulation, does it decrease t on, t off, both, neither, or something or another? All right, eight seconds. Ready, three, two, one.

All right. So we got many C's, but many other things as well. Right. But indeed it's only going to decrease t on, actually. And that's because remember, the negative autoregulation what it does, is initially, it makes a lot of this protein. And then once you hit the repression threshold, then it sort of represses itself. And so you are able to kind of rapidly get up to that level, and then you clamp it there.

Whereas turning off that just means that you just tell to stop making it. So it's always going to then decay at the rate that is dictated by its effective lifetime, which is kind of either from the protein or from the actual degradation, or from the growth of cell. So this only decreases t on.

And then indeed, we just learned about another one here, which was the incoherent, incoherent type 1 feed-forward loop, with kind of modest amounts of repression, incomplete repression or so. And what does that do here? Did we actually even figure out what it did here? Oh, we maybe didn't say. Five seconds, again over here. What does it-- we've-- well I told you one half of it already. But what about the other half? All right.

All right, ready? Incoherent type 1 feed-forward loop, what does it do? Ready, three, two, one. OK. We've got again, a bunch of C's. Indeed this is-- and of course, I'm using the same chart for the sign sensitive delay, and for the time to turn on turn off. So I hope that that doesn't confuse you. If it does, I'm sorry.

All right. So this is, it decreases t on, but not t off. This is an AND gate. So you need both active X star and Y star. So the moment that you make Sx go away, then it's going to start-- oh wait. I'm explaining a different one. OK. Sorry. This is the time. So we're at Z. But yeah, it immediately starts going down. But at the same normal rate of effective lifetime.

Many more ways to make a protein quickly, than to get rid of it quickly. Are there any questions about what we've said here?

So while we're on the topic of kind of response times and so forth, it's important to remember that the characteristic time for all these things is kind of the generation time, or the protein lifetime. So these are ways of processing information over time scales that are kind of like minutes, maybe even tens of minutes, maybe hours. So it's rather slow.

Now is that, in general. Going to be good enough for everything that a cell needs to do? No. So it's important to highlight that transcription is slow. So that means that transcription networks are going to be slow. And that's both because you actually have to do transcription and translation, and so forth. But also you just have to change the concentrations of proteins.

So if you need to kind of respond to things more rapidly, what is it then you need to do?

AUDIENCE: Phosphorylate.

PROFESSOR: Phosphorylate, yeah. It's all about phosphorylation, yes. Indeed, phosphorylate-- so you need, if you want to be fast, you can't be changing overall protein concentrations. You have to change protein state. Right, and phosphorylation is kind of the classic way of doing that. so I just want to highlight that for speed you need just to do kind of protein networks.

So we're not actually going to be reading the chapter analyzing these map kinase cascades, and so forth. But if you're interested in such things, I very much encourage you do so. It's also nice chapters, maybe not as nice as the first four, which is part of why we're not reading them. But it's a very important insight, in the sense that we've spent a lot of time talking about transcription networks, just because there's a lot of, I think, simple, beautiful things that you can say about them. Whereas they are intrinsically limited in terms of speed.

So for much of what a cell needs to do, it has to already have the proteins there. And then you can take advantage of these rapid processes. So we talked about Sx rapidly binding, changing the state of X. From the standpoint of transcription networks, we just draw this as a straight line. It's rapid. But what that's saying is that if you just change states of proteins, then you can do a lot of information processing rather rapidly. And you don't have to do just a simple thing of Sx binding X. You can also have proteins regulating each other, and performing logic functions at the protein-only level.

What I want to do for the last 20 minutes is say something about temporal programs that can be implemented with sort of larger network motifs. And in particular this is material basically from chapter five of the book, which again we're not to be reading. I think it's again, it's beautiful but it's really simple. So I think that in 20 minutes we can cover it just fine.

So for many cases, for example, in the context of metabolic pathways, it might be the case that you have some protein we'll call them Z's. So you'll have some protein Z1 that does something. So that catalyzes-- so we might have some molecule one that's converted into module two, by Z1, converted into molecules three by Z2.

So many metabolic pathways have this structure where there are a series of enzymes that are doing something to the product of the previous enzyme. Now the question is, let's say that this is some carbon source and we didn't before. But now it's appeared in our environment. So what we would like is we'd like to start digesting that carbon source. Or in the flip side, maybe we have to make some complex molecule or an amino acid or so. And so then what we're doing is we're building something up, coming down.

Now in either case, if before you weren't making these Z proteins, but now you want them, a question is, maybe you could just make them all at the same time. But maybe it would be better to make some of them first, and some of them later. What do you think?

Let's say for the sake of argument that you would want to have some first, and some later, which ones would you want first?

AUDIENCE: The ones that you use first?

PROFESSOR: Yeah, the ones you need first. So you'd maybe want to first have Z1, then Z2, et cetera. It's a trivial statement, but you might not actually think about it. And the question is how might we be able to do this.

Well there's a very simple thing called a single input module. The idea here is there's some transcription factor X, which actually does this fabulous thing where it creates all these guys. Interestingly, this also often has autoregulation in order to, for example, stabilize the concentration of it.

But the reason it's called a single input module is because the network motif is saying not just that X makes many Z's. That actually you can't actually argue that that's a network motif, from the standpoint of a degree preserving network. Because of course, this is just saying well some nodes activate many other nodes. And in a degree preserving network that's always still going to be true, right?

But you can say that such a thing is a network motif, when you say that these Z's are only regulated by X. So that happens somehow more frequently than what you would expect.

Although now that I just said that, I'm a little bit worried that even the degree preserving would-- I think you have to be a little more subtle in defining your null model in that case. But I'll just say that this happens more frequently than you might expect, which is that you have one transcription factor, say activating many, many different proteins. And this makes sense.

Because if all these Z's are involved in the same metabolic program, then when you want Z1, you also want Z2, and you also want Z3. So this makes a lot of sense. But what is a little bit less obvious perhaps, is that it's possible to do this such that you first make one, and then you make the other.

And the way that you can do this is just by, you have different activation thresholds, K1, K2, et cetera, up to Kn for each of these. So then if X is turned on, so let's say that you first see something. So this you actually have to have X start at 0, and then grow over time. But then if you just have different thresholds, the question is where should I draw K1, and where should I draw-- Do I draw K1 the low position or the high position?

Low. Perfect. All right. So we say K1. Here's K2. Here is Kn. And then there might be some others in between. So the idea is that X grows over time. Then you first activate expression of gene one, and then gene two, and so forth. And then the proteins will naturally appear in the proper order. And there's actually beautiful data in chapter five illustrating this in the context of our gene biosynthesis.

So it's quite neat to see that it's not just-- of course, it's easy to think up this idea. And say, oh yeah. Maybe the cell might want to do this. But then it's quite cool when you see that actually in some cases, the cell actually really does do this. And you can actually see that they're expressed sequentially, in the same order as they appear in the biosynthetic pathway.

So this kind of gives you a warm, fuzzy feeling inside. I'm not going to make you vote on whether you have a warm fuzzy feeling. But, yeah?

AUDIENCE: Could you also have just some sort of Z1 required for transcription?

PROFESSOR: Ah, yes. So you could have a cascade in that way. And that's a really good question. That would work. But there's a problem, which is that it's super slow. Because there's a characteristic timescale for each of these things, which is this cell generation time. And here when I say you want it after the other, what I'm saying is you might want it a few minutes after the other. So in the context of development, in some cases some things really are very slow. Then that is actually what happens. There's a long cascade of one activating two, activiating--

But in the context of this, you really want something that's just delayed by five minutes each, or maybe even just a couple minutes each. And in that case, because really the range over which you can have this sort of delay like this, from the beginning to the end, I mean this is still-- it might be one to two, say cell generations/lifetimes. So you just can't get much more of a dynamic range. Otherwise you're going to be in trouble.

Because you can't have this be too close to the top. Otherwise if you're a little bit off, you're off. So we really maybe we should just even say one generation. So that's kind of how much of a delay you might reasonably be able to get from this mechanism. And indeed, and that's as much as you would want for something like this.

So this is great. When I first read this, I was like, oh. I was feeling it. Now the question is, after this carbon source, or the need to make our gene or whatnot, after it goes away, then we'll stop making these proteins. And the question is, is this what we call a FIFO queue, or a LIFO queue?

LIFO, LIFO? It's been a long time. I have a lot of faces that are like, what are talking about? So this is a First in, first out. Have you guys really not-- you never took any of these computer science classes where they talked about this? And this is a Last in, first out.

So just to be clear, what this means at the grocery store is that a first in, first out, or a last in, in first out.

AUDIENCE: First in, first out.

PROFESSOR: It's a first in, first out. Right? If you get in line first, you get out first, hopefully. And when that doesn't happen, you get very annoyed, and so forth. But last in, first out, this is for example what happens in your inbox. So you have a stack of paper. People are giving you things you're supposed to sign or fill out. And the pile kind of comes up, and then you handle things on top of the pile first. So the things that get stuck at the bottom you never get to them. And that's because that's a last in, first out queue.

And so different, depending on-- well and in computer science, then they can choose these things. And it's relevant and so forth. But there are many situations where this sort of idea appears. And so the question is, in the single input module, do we have a first in, first out, or a last in, first out queue? I'll give you 15 seconds to think about this. Yeah, question?

AUDIENCE: What do you mean by in and out?

PROFESSOR: So in and out, what I mean is that the concentration of each of these Z's, these guys they were produced in some order. Right? So first we started producing Z1, then Z2's and then so forth up to Zn. And what I want to know is what order will the concentration kind of go away in?

And you might want to look at this figure. Because it's going to be super useful. Any other questions about what I-- all right, so in and out refers to concentrations going up, and then going down.

Let's go ahead and vote. Ready, three, two, one. See, I mean people learned what FIFO and LIFO queues were. And now already we can use it. So this is a last in, first out queue. And in general, which kind of queue do we like? We like LIFO queues more. Well, OK, you could argue. But in this actuation would we like a FIFO or LIFO queue? I'll give you 10 seconds. In a biosynthetic pathway, would you want a LIFO or a FIFO queue?

Yeah, and if you're totally confused by everything I'm saying, you can do the-- flash all the letters, numbers, whatever. All right. Ready, three, two, one. All right. So there's some disagreement we got. But now a majority of people are saying that although the single input module gives us a LIFO queue, what we might really like is a FIFO queue. And can somebody say why that would be?

AUDIENCE: We don't want that much intermediates.

PROFESSOR: Right. We don't want to pile up those intermediates. So just for the same reason that we wanted to start with Z1, and then get Z2 and so forth. And the reason that was because there's no point in having Z2 until after we have Z1. Because there's nothing for Z2 to do. It would be wasted energy to make it.

In the same way, when we're getting rid of these proteins, we would actually like to get rid of them first this one and then later.

AUDIENCE: Why? There's no point in having Z2 if you don't have Z1. You technically want to get rid of them in the other way. Because then if the carbon source shows up again, you want to have--

PROFESSOR: Well yeah, carbon source showing up again. I think that's a separate argument. So the reason that we-- the statement is really just that if we first get rid of Zn. Then we are just going to pile up molecule n minus 1.

AUDIENCE: So the metabolism is down.

PROFESSOR: That's right. That's right. So you can just think about the flow of those of the metabolites and the molecules in there. And you kind of want to first get rid of this. So then we stop making this. And then we kind of travel on down. So there are many contexts in which you would perhaps really like to have a FIFO queue. And indeed, one of the things that had been studied previously was the flagellar biosynthesis pathway. So E. coli and many other bacteria, they make these flagella that allows them to swim.

We're going to talk a lot about that in coming weeks. But in this context what had been found actually is that it is indeed a FIFO queue. So when they first start to make these little flagella, they make. And it's a big complicated, machine. Right? But they make it in the order that it's transported and put in the membrane. But then when it is taken away, when you stop making the flagella components, then it's again in the same order as they were made in, which kind of makes sense.

Now the question is, how might we be able to do that. So let me explain it. The basic answer is via an extension of these feed-forward loops. So it's what we call a multi-output feed-forward loop.

What we have here is we have some X, and it is going to come to Y. And then Y again is going to do this thing to Z1 and Z2, and all the others. But it's a feed-forward loop, because we also have X coming in here as so. And I think that we do want to have these be AND gates. Let me just make sure I'm not-- no here's it's an OR gate.

So we're going to assume that all of these inputs are combined via an OR gate. And what we have is we have some K1, K2, et cetera, up to Kn. Then we have another set of K's which are K1 primes, K2 prime, Kn prime. So we have a set of K's corresponding to how X interacts with the promoter at the Z. We have different set of K primes that tells us how Y interacts with the promoter at Z.

And the question is how can we get a FIFO order.

I'm going to illustrate some options. And you can think about it while I do it.

AUDIENCE: Are those associations or dissociations?

PROFESSOR: These are dissociation constants. So these are again, this can be thought of as the concentration of the active protein which it starts being effective in sending a signal to Z. Question?

AUDIENCE: Would you not like [INAUDIBLE]?

PROFESSOR: I hope not. Because otherwise I'm going to be in trouble. Maybe for now let's, assume that I've written the right thing. And then we'll find out soon enough. Does everyone understand the question here? I'll just give you another 15 seconds to think about it.

All right. Do you need more time? All right. Just a little bit more then.

All right. Let's go ahead and give it a go. Ready, three, two, one. OK. So we got a majority B, but some C's. All right. So let's try to figure this out. So what we assume is that X comes and it's going to do this. And then it's going to do this. And we can just have two values for now, K1 and K2. So this is X. Now we're going to have Y.

And we can talk about-- this is also X star Y. We'll assume that we always have these things. So Y is also going to come. It'll be activated here at some time, which we don't-- it's going to be delayed by a little bit, right? Maybe.

So that's what Y is going to do. Now we want to know about Z1 and Z2, right? Well, the idea here is that it's going to be the K, the regular K1 and K2 that determine the order that it appears. We have an OR gate. Y is going to be delayed. So it's really, the important thing is what the normal K's do in terms of turning on Z's. Yes?

And this is especially true because Y is again delayed. Because it has its own threshold for turning on. So since Y is delayed, it won't really have a chance to influence the behavior of the X's. And actually I should have drawn this delayed too, as well.

So it's the order of the K's that determine how Z is turned on. But it's the order of the K primes that tell us how the Z's are turned off. And that's again because Y is delayed relative to X. And we have an OR gate.

So indeed, in this case, if we have like this, and we want things to be in the opposite order with respect to Y. So if we wanted K1 to be less than K2, then we actually want say Kn to be-- I'm sorry. We want K1 here, and then we want Kn down below.

And the heart of this is really because of the fact that X is also regulating Y with some other constant Ky. And this is going to tell us how much Y is delayed relative to X. But the heart of this is that since Y is delayed, it's really the dynamics of X at the beginning that tell us how the Z's are turned on. But it's the dynamic of the Y and the K primes, K1 prime, Kn prime, that tell us the order at which those Z's are going to be turned off.

So with the proper choice of orderings of K's and K primes is you can then get a FIFO queue. With that I think we should quit. Please read Sunney Xie's paper very carefully, because it's going to be focused on a lot over the next lecture. All right. Have a good weekend.