1 Protein Interaction Networks : Structure

1.1 What are proteins and what do they do?

Proteins are large molecules, created by cells, that carry out specific cellular functions. They are the machinery of the cell. Every protein is encoded in an organism's genome by a specific gene. Each proteins is built from some specific string of amino acids, which then folds up into a 3d shape. The folding process itself can be simple or complicated (sometimes requiring the presence of additional proteins or other molecules to fold properly). Once folded, we often say that proteins are composed of different "domains."[1] Some of these domains create structures like a pocket or other binding-ready shape, and it is through these structures that proteins mainly interact.

There are two main types of binding: (i) stable binding, in which the two proteins stay stuck together after the binding event, and (ii) transient binding, in which they do not. Transient binding can be short, or long, and is the type of binding used in information relays as in a signaling pathway.

When we visualize proteins, there are three ways to shown them (see Fig. 1 below)[2] : (i) cartoons, (ii) helices and sheets (also called "secondary structure"),[3] and (iii) atomic structures (visualizing individual atoms). For this class, the "cartoon" visualization is sufficient, and we will generally not care about secondary structures or protein domains. The cartoon is sufficient to capture the idea that a protein is a thing that binds to something else.

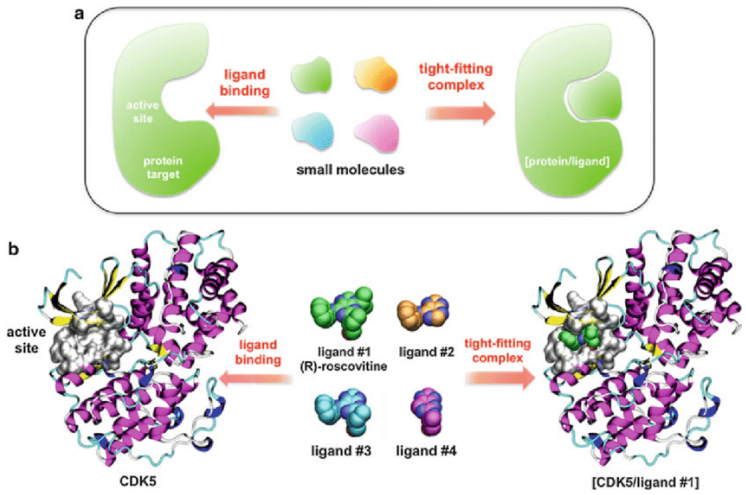


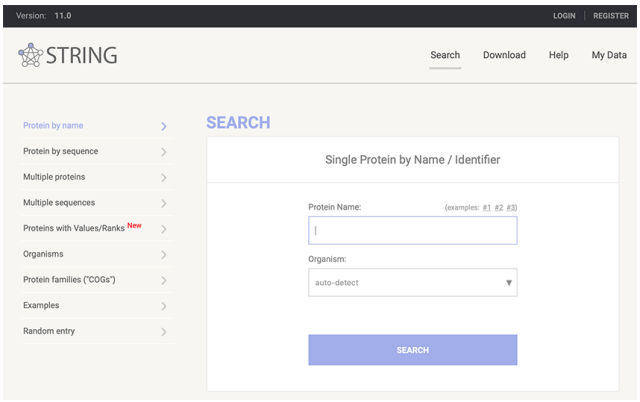
Figure 1: Visualizing proteins as (a) cartoons and (b) helices and sheets, and an atomic structure.

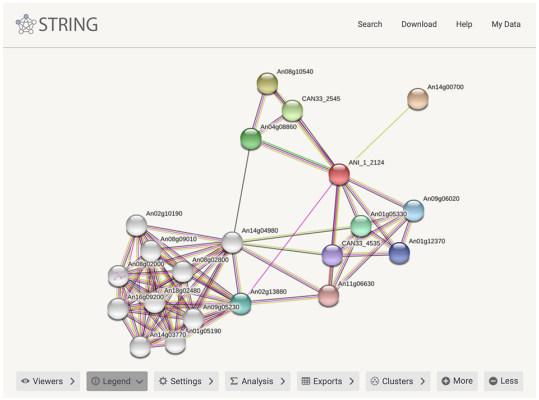
[1] See `https://proteinstructures.com/Structure/Structure/protein-domains.html`
[2] Source: `https://bit.ly/2v6xoPp`
[3] See `https://proteinstructures.com/Structure/Structure/secondary-sructure.html`

1.2 Protein interaction databases

There are many databases that list protein interaction data. The STRING database is one that includes both known interactions (meaning: those with evidence from experiments) and predicted interactions (meaning: predicted based on protein domain structure, co-expression data, etc.). In the "v11" version, STRING covers nearly 10,000,000 proteins across more than 2000 organisms, making it one of the largest available.

Here is the basic search box looks for interfacing with STRING at `https://string-db.org/`, along with one result obtained via the "Random entry" option:

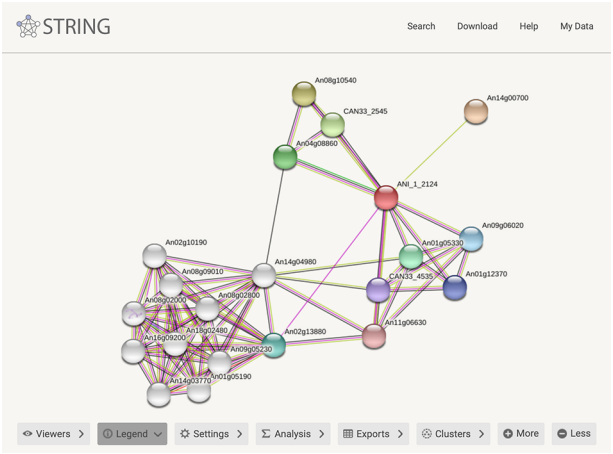


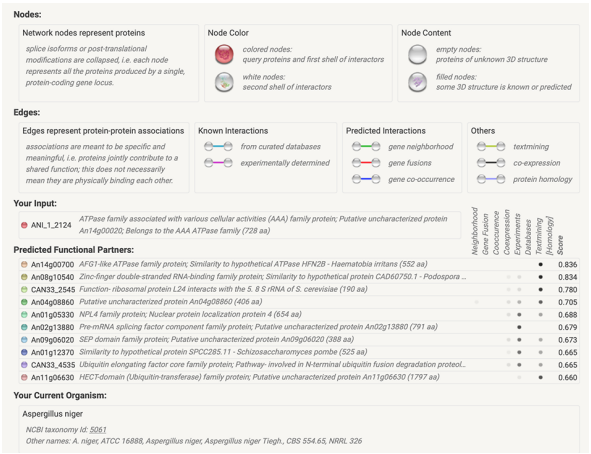


In this case, STRING picked the protein `ANI_1_2124` (proteins generally have names like this; sometimes they're meaningful, but usually they're just a unique string), and then shows us its local neighborhood in the PPIN. Initially, this view into the network shows just immediate neighbors, but we can use the `+More` button to show nodes +1 hop away.[4] The red node in the middle is `ANI_1_2124`.

Below this visualization is a Legend that illustrates the rich kinds of network data that STRING stores, including different types of edges, edge weights, edge signs, node metadata, etc. It is a remarkable amount of information. And yet, overall, we know remarkably little about the set of proteins, their properties, and their interactions. Such is the scale of the task of just describing all the pieces of a PPIN.

[4] Source `https://version-11-0.string-db.org/cgi/network.pl?networkId=pNkiZSN3VyXo`





1.3 Signaling pathways

If we zoom in on certain parts of the full PPIN, we can find what are called "signaling pathways," which are a set of nodes and edges (usually directed) that represent a kind of cellular sensor. Some of these are extremely well studied because when they break, they lead to diseases or other issues of particular interest. For instance, the MAPK pathway is used by cells to detect the presence of a protein called EGF (epidermal growth factor) outside their cell membrane, and, if it is, change what genes are expressed in the nucleus (contributing to cell proliferation). Changes in the MAPK pathway are implicated in diseases including cancer, inflammatory diseases, obesity, and diabetes.[5]

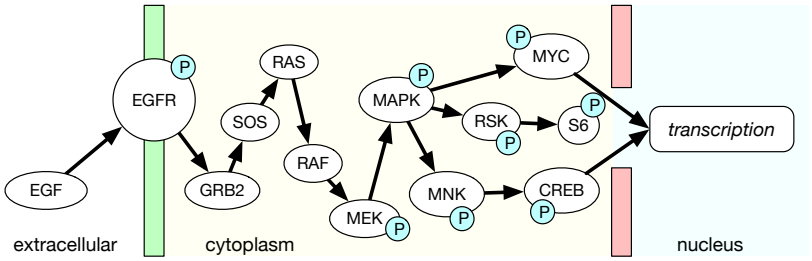


Figure 2: The MAPK pathway, which detects the presence of the EGF protein outside a cell.

But, the MAPK pathway is itself embedded within a larger signaling network, composed of many more proteins. The visualization below shows this larger network, which also contains the WNT

[5]See Lawrence et al., "The roles of MAPKs in disease." *Cell Research* **18**, 436–442 (2008). `https://www.nature.com/articles/cr200837`

pathway, another well-studied pathway implicated in a variety of diseases,[6] and a variety of intracellular proteins that interact with these pathways. Within this larger network, we may observe that there are two different kinds of protein interactions shown: *activation edges* (an arrow) and *inhibition edges* (a line with a blunt end). For instance, the structure around the JNK node (in the middle) indicates it has five input interactions: two activation inputs and three inhibition inputs. Hence, JNK works like an elaborate conditional function in a programming language: only if all its inputs are satisfied, will it be able to activate itself and continue the information cascade toward the nucleus.

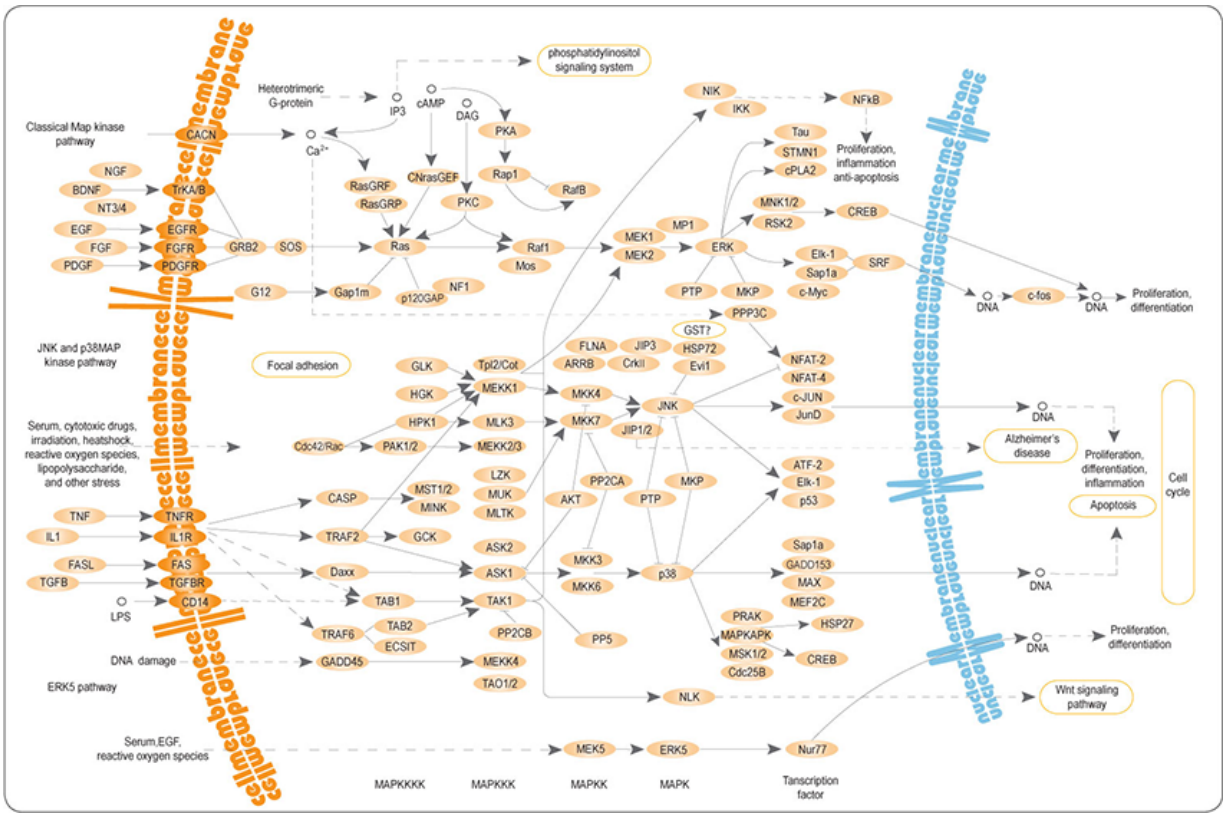


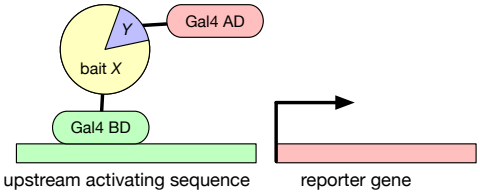
Figure 3: The MAPK and WNT signaling pathways, along with a variety of intracellular proteins that interact with them. EGF is located in the upper-left corner.

[6]Source: `https://en.wikipedia.org/wiki/File:MAPKpathway.png`

1.4 Where do the data come from?

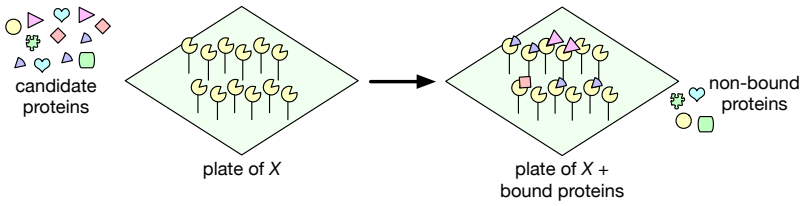
Data in a PPIN come from one of four main sources:

1. **yeast two-hybrid (Y2H)**
   Y2H is an *edge sampling* experiment, which tests for binding between a specific pair of proteins X (the "bait") and Y (the "prey"). Y2H does this by inserting the potential interaction between the binding and activation domains (BD and AD below) of the Gal4 transcription factor so that the downstream gene is only expressed if X and Y successfully bind.



   High-throughput versions of Y2H allow testing for many such pairings in parallel, but it also has many limitations. In particular, the Y2H method works best for nuclear proteins, and it can have a high false positive rate (activated gene is expressed even when X and Y don't bind) and a high false negative rate (no expression even when X and Y do bind). Various improvements to Y2H have aimed to mitigate some of these issues.

2. **affinity purification and mass spectrometry (AP-MS)**
   AP-MS is a *node sampling* method, which takes a protein of interest X and make an array of them on a plate. We then wash the plate with a mix of other proteins and use mass spec. to identify what stuck to the Xs.



   AP-MS can also have a high false positive and a high false negative rate, but for different reasons than Y2H. The false positives come in part from the fact that we mass spec. everything on the plate, meaning we cannot tell the difference between all proteins $\{Y, Z, A\}$ binding to X from a "chain" of bound sequentially bound proteins, e.g., the edges $(X, Y), (Y, Z), (Z, A)$, of which only Y binds directly with X. False negatives from from weak binding events where none of the protein Y remains bound to X after washing the plate. Various improvements to AP-MS have aimed to mitigate some of these issues, as well.

3. **text mining**
   We can also use natural language processing methods to trawl the scientific literature to identify interactions that were reported via various (typically low-throughput) experimental approaches. This approach also has a non-trivial false positive and false negative rate, because it can be tricky to correctly identify an interaction from a paper's language alone.

4. **prediction from protein domain analysis, etc.**
   Based on the secondary and tertiary structures of two proteins, e.g., the shape and location of their binding domains, we may also be able to predict that proteins X and Y interact by identifying that X has a domain that could bind to one of Y's domains.

It is important that each of the above methods has non-trivial false positive and false negative rates. Because real-world networks are typically sparse, and PPINs are no different, there will be $\Theta(n^2)$ non-edges, each of which could be misclassified as an edge (a false positive), and $\Theta(n)$ true interactions, each of which could be misclassified as a non-edge (a false negative). It is likely that in the real PPIN data we have, a great many of the observed interactions (derived using the above methods) are in fact false positives, and a great many of the true interactions are missing (false negatives). Basically, PPIN data is messy, and it is important to bear that in mind as we learn about what we think we know about their structure.

In addition, PPIN data typically omits other types of biologically useful information about proteins:

- binary interactions only, meaning we don't have edge weights (binding affinity) or edge sign (activation or inhibition);
- no data on *where* in the cell the protein occurs, or *when* in the cell cycle the protein is expressed, meaning two proteins that bind in an experiment may never interact in the cell;
- no data on protein *concentrations* in the cell, i.e., is it a rare or abundant protein; and
- many proteins are *missing functional labels*, meaning we know they exist, but we do not have much sense, or a complete sense, or what cellular functions they are involved in.

1.5 What do people do with PPINs?

1. **fill in the details**
   There is value in just having as complete a description of the PPIN as possible, having a complete list of proteins, and a complete list of their interactions, and a complete list of annotations on the edges and nodes. On the computational side of things, this often means
   • predict missing node attributes (functional labels, etc.)
   • predict missing (and spurious) links

2. **find modules** ("the building blocks of complexity")
   • these may be small, as in "complexes"
   • these may be large, as in communities (the way we've defined them in this class)

3. **understand disease**
   The PPIN is different from social networks in that it is functional, meaning it carries out specific functions and its correct functionality is essential to life. When it breaks, it can create disease.
   • understand the pathways and partners of proteins associated with specific diseases, e.g., Alzheimers, Parkinson's, Schziophrenia (see below), various cancers, etc.
   • understand what specific proteins regulate, and what regulates them, both of which might be targets for therapies
   • what modules and pathways are associated with specific diseases, which ones are shared across many diseases
   • modeling the interaction dynamics of these pathways to understand how they lead to disease
   • do specific (local) network structures induce greater or lesser resilience to disease states?

4. **systems biology**
   • what can the architecture of the entire PPIN tell us about how cells work?
   • are there general principles that structure it?
   • the PPIN may look like spaghetti, but it is evidently both highly functional and highly robust. How is that possible?

The tools researchers use in these efforts are familiar:

- **motifs** (small subgraphs) and their frequencies relative to null models. Motifs are believed to act like little logical circuits, and hence hooking them together in a network should (we imagine) create a kind of dynamic computational program. Finding the motifs that are more abundant than we would expect (under the null) may offer hints about how it works.

- **modules**, meaning communities, which provide a coarse graining of the network that allows us to think about it as a lower-dimensional object

- **pathways**, especially in dynamic models, where we model the abundance of individual proteins and their sequential activation

- **hubs** ("high" degree nodes), for example, whether they interact with many other proteins at the same time ("party hubs"), or over time ("date hubs"). There is often great emphasis placed on proteins with "high" degrees (aka, hubs), but it's not clear how much of their apparent importance is driven entirely by the fact that they simply have more connections (and thus more chances for being connected to other things).

1.6 From structure to dynamics: computing with protein interaction networks

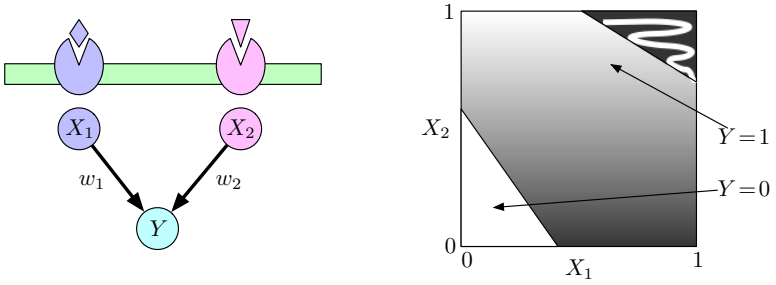
What makes protein interaction networks interesting is that they are functional, meaning their dynamics produces useful phenotypic behavior for the cell. There are three general approaches to modeling how the structure of a PPIN produces biological function:

1. **rate (differential) equation models**, which aim to capture the temporal dynamics of protein concentrations and rely on detailed knowledge of the biochemical kinematics of the associated interactions,
2. **linear models**, which approximate the rate equations with simple linear functions, and
3. **boolean network (logical) models**, which approximate the linear models with simply binary logic.

These models are typically deterministic, but more realistic versions can incorporate stochasticity. In all three models, the crucial beginning assumption is that the network structure G is known. For simplicity, we start with linear models.

1.6.1 A simple, linear circuit

In signaling, protein interactions can operate like little circuits. For instance, consider a pair of membrane-bound proteins (the pacmans below) that activate molecules X_1 and X_2, respectively, when they detect their respective preferred molecule outside the cell membrane (green band). Both of these molecules can then bind to Y, but do so with different weights w_1 and w_2, respectively.



In a linear model of this system, we ignore the underlying rate equations that govern how the dynamics actually work. Let X_1 and X_2 denote concentrations ranging over $[0, 1]$, and let w_1 and w_2 denote each molecule's "binding strength" with Y. We then say that Y becomes activated if the linear combination of its inputs exceeds a threshold:

$$Y = \begin{cases} 1 & \text{if } X_1 w_1 + X_2 w_2 > 1 \\ 0 & \text{otherwise.} \end{cases}$$

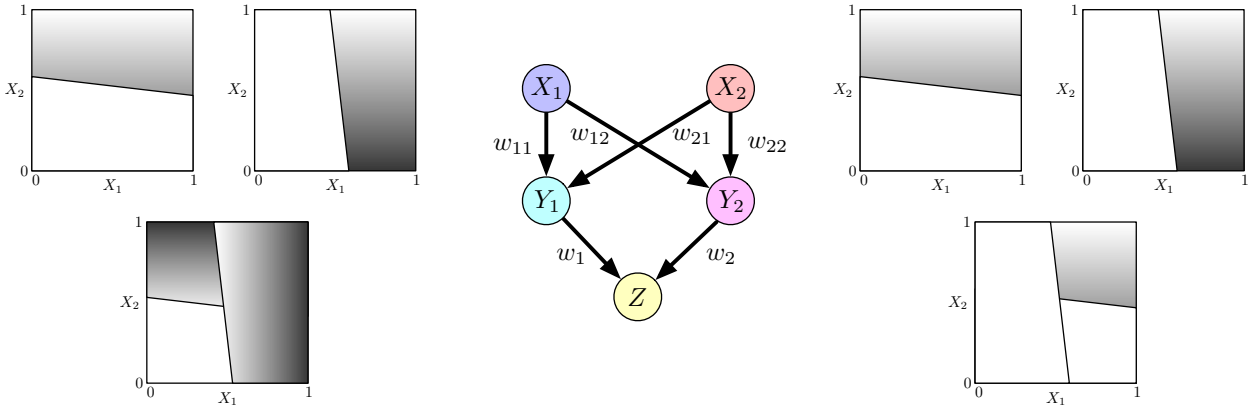
The right-hand plot above maps out the response of Y to variations in X_1 and X_2: the two shaded regions represent the joint concentrations X_1 and X_2 that activate Y, for two different choices of

weights w_1, w_2. (The values of w_1, w_2 determine the slope and intercept of the line that divides the input space into $Y = 1$ and $Y = 0$. Do you see why?) In this sense, we can say that this motif, i.e., this subgraph of the larger PPIN, computes this function.

1.6.2 Combining linear circuits

Now consider putting two of the above circuits in parallel with each other, so that X_1 and X_2 both feed into molecules Y_1 and Y_2, and these two in turn feed into a new molecule Z. Now we have six free parameters, and three corresponding "activation plots," one for the activation of each of Y_1, Y_2, and Z.

But, the interesting part comes in the way the two input plots interact to produce the plot for Z: if the weights w_1 and w_2, which feed into Z are both very strong, then activating either Y_1 or Y_2 is sufficient to activate Z. But if they are individually too weak to activate Z, but are strong enough together, then we must activate both Y_1 and Y_2 in order to activate Z. That is, depending on how we set the weights (w_1, w_2), this little circuit is either an OR gate or an AND gate for X_1 and X_2. (In this example, all the weights are positive, representing excitatory interactions, but negative weights are possible, which represent inhibitory interactions.)



$w_1 > 1, w_2 > 1$

$w_1 < 1, w_2 < 1$
$w_1 + w_2 > 1$

Hence, simple circuits can be combined to compute more complicated functions over the input space, and this is what we imagine that signaling pathways do; the more complicated the pathway, the more complicated a function the circuit can compute.

However, there are several large caveats for applying these ideas more broadly to analyzing a PPIN:

1. Generally, we only know the weights (binding strengths, etc.) for a small subset of very well studied protein interactions, even in crucial pathways like MAPK. Hence, we cannot perform

large-scale analysis of the kind of functions that the PPIN computes, because changing the weights necessarily changes the function (as the example above shows).

2. Drawing small circuits is easy in isolation, but real PPINs, and even just real signaling pathways, are a mess of cross-talking, interacting, overlapping circuits, and it's a big stretch to go from understanding how to build an OR or AND gate to understanding the architecture of an entire computer (or cell).

For these reasons, developing a realistic linear model of how PPINs compute often still remains outside our reach. These constraints also tend to constrain the use of rate equation models to very simple circuits, like the well-studied Lac operon in bacteria that controls the expression of the lactase enzyme.

1.7 Boolean circuits

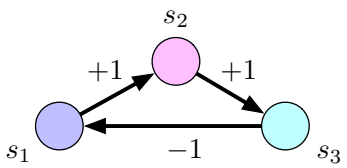
In a boolean network model, we let each node have a binary state $s_i \in \{0,1\}$. The particular assignment of state values $\mathbf{s} \in \{0,1\}^n$ defines the system's state, and the following simple update rules define the way the network traverses this space:

$$s_i(t+1) = \begin{cases} 1 & \text{if } \sum_{j=1}^{n} A_{ij} s_j(t) + c > 0 \\ 0 & \text{otherwise,} \end{cases}$$

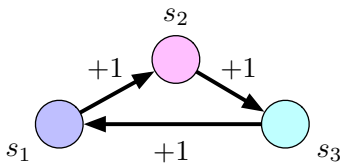
where A is the adjacency matrix,c is a threshold, as in the linear model case, and we update all states from $s_i(t)$ to $s_i(t+1)$ synchronously. Letting the network be a **signed** graph, in which $A_{ij} \in \{-1, 0, 1\}$ allows this model to capture both activation ($A_{ij} = 1$) and inhibition ($A_{ij} = -1$) interactions. (Do you see how this model is a simplification of the linear model in Section 1.6.1?)

1.7.1 A simple circuit

To see how a boolean network works, consider two very simple circuits: a feedback loop (FBL) with inhibition, and a feedback loop with activation. Let's see how these systems evolve when we initialize them to a few different states, with $c = 0$ in the update equation.



feedback loop (inhibition)



feedback loop (activation)

- In both cases, if all nodes are inactive, then $\mathbf{s}(0) = 000$. Applying our update equation, we find that $\mathbf{s}(1) = \mathbf{s}(0)$, meaning this state is an "attractor," and specifically a "fixed point," meaning that once the system reaches this state, it states in it forever.
- For the inhibition FBL, if we initialize the system as $\mathbf{s}(0) = 100$, we then visit $\mathbf{s}(1) = 010$, $\mathbf{s}(2) = 001$, and finally $\mathbf{s}(3) = 000$, as before. Qualitatively, this circuit runs once, and then turns off. If there were an active input to s_1, and an output at s_3, this circuit would produce a "pulse" signal that emerges from s_3 exactly once every three time steps, but only so long as the input were active.
- For the activation FBL, if we initialize the system as $\mathbf{s}(0) = 100$, we then visit $\mathbf{s}(1) = 010$, $\mathbf{s}(2) = 001$, and then back to $\mathbf{s}(3) = \mathbf{s}(0)$, and so on forever. This sequence of states is a "limit cycle," the remains active forever, once turned on. If there were an output attached to s_3, this circuit would generate the same "pulse" signal as the other FBL, but would do so even after s_1 was deactivated.

1.7.2 More complicated circuits

Boolean networks can be used to model more complicated systems. For example, they have been used to illustrate the robustness of the regulatory network structure that governs the cell cycle in the budding yeast *Saccharomyces cerevisiae*, a single-cell model eukaryotic organism.[7]

The yeast cell cycle consists of four phases: G1 $\rightarrow$ S $\rightarrow$ G2 $\rightarrow$ M $\rightarrow$ G1, which correspond to cell growth, DNA synthesis, a "gap" phase, and finally the division phase, after which each daughter cell returns to the G1 state and awaits a signal to move to initiate another cycle by moving to S. Although more than 800 genes are involved in the overall cell cycle, a simple model of 11 key proteins, plus one "check-point" state, provides a minimal representation of the regulatory network (Fig. 4).

With 11 nodes, there are $2^{11} = 2048$ possible states for this system, which is small enough to enumerate every state transition $\mathbf{s}(t) \rightarrow \mathbf{s}(t+1)$, for all possible choices of $\mathbf{s}(t)$. (The exponential size of the state space means this approach quickly becomes computationally intractable for even modest-sized boolean networks.) These pairs are themselves edges in a graph where nodes are entire states of the system, and analyzing the structure of that graph lets us understand the kind of dynamics the boolean network produces. Because each state "points" to a single next state, this graph is a forest, and each component is a "basin of attraction," in which starting the system at any node in that tree will lead it to converge on a fixed point or a limit cycle. The larger the tree, the more states in the basin.

[7] Li et al., "The yeast cell-cycle network is robustly designed." *Proc. Natl. Acad. Sci. USA* **101**(14), 4781–4786 (2004). `https://www.pnas.org/content/101/14/4781`

In this model of the yeast cell-cycle network, there are 1764 states (86%) are connected in a single tree, and a line of 13 of those states maps precise onto the four phases of the cell cycle: a START state, three G1 states, an S state, five M states, and then two G1 states (Fig. 4). The remaining 14% of states group into smaller basins, composed of 151, 109, 9, 7, 7, and 1 state.

The large size of the largest component in this "state space" graph means that yeast's cell-cycle network is very robust. That is, for a very large number of perturbations to the system's state vector, i.e., changes to $\mathbf{s}$, the network will "relax" back to the main sequence of states representing the four cell-cycle phases.

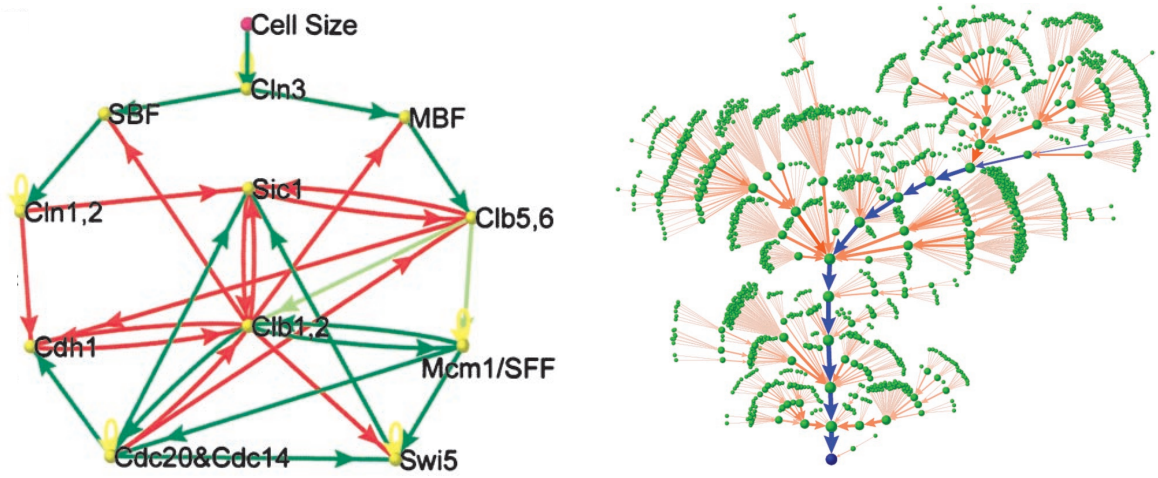


Figure 4: (left) A simplified model of 11 key factors that regulate cell division in yeast, which span four classes: 4 cyclins (Cln1,2; Cln3; Clb1,2; Clb5,6, which bind to the kinase Cdc28); 3 inhibitors, degraders, and competitors of the cyclin/Cdc28 complexes (Sic1, Cdh1, Cdc20&Cdc14); 4 transcription factors (SBF, MBF, Mcm1/SFF, Swi5); and 1 check-point (Cell Size). Green arrows represent activation, and red inhibition. The cell cycle begins by activating Cell Size, which activates Cln3. (right) The largest component of state transitions, covering 1764 (86%) states total, with the main sequence of states corresponding to the four cell-cycle phases shown in blue. (Reproduced from Li et al. (2004).)

1.8 The structural compromise: Counting motifs

As a compromise to the computational complexity of modeling biological circuit dynamics, researchers often resort to a structural approach: identify and count subgraphs that are computational building blocks. When doing this, we typically draw a distinction between a subgraph (a

specific, and usually small set of nodes and edges) and a motif:

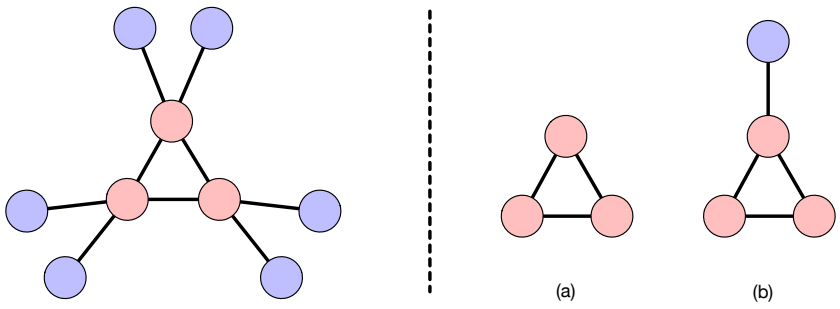
A motif is a subgraph that appears more often than expected.

The latter part of this definition implies a choice of a null model, i.e., some $\Pr(G)$. We have already seen several such possibilities (see Lectures 3 and 5). Usually, researchers choose the configuration model as $\Pr(G)$, meaning they are measuring the abundance of a given motif of interest relative to a random graph with the same degree sequence.

Before we can count motifs, however, we must choose what counts as a new subgraph. There are two choices:

1. non-identical subgraphs, meaning subgraphs that different in at least 1 node or edge
2. edge-disjoint subgraphs, meaning two subgraphs that do not share any edges, i.e., we cannot reuse a node or edge in a second subgraph, once it's been included in another.

To illustrate the distinction, consider the following example. On the left is a graph G, and on the right are two possible motifs. Under the edge disjoint definition, there's 1 subgraph of type (a) or 1 subgraph of type (b). (Do you see why?) In contrast, under the non-identical definition, there's still only 1 subgraph of type (a), but now we can find 6 subgraphs of type (b). Which of these is the right way to count? The answer is: it depends. It depends on the type of scientific question we want to answer.



The maximal motif G_s is defined as a subgraph of G that is not contained in any other subgraph of G that is a motif. That is, it's the largest subgraph that is also a motif. If G_s is contained within another motif, then it's not the maximal one.

Suppose we have an algorithm that can count how many of a given motif G_k are within a graph G, which we can denote $f(G_k)$. To determine whether or not that count $f(G_k)$ is more than we expect, we often calculate a z-score under the given null model:

$$z\text{-score} = \frac{f(G_k) - \mathrm{E}[f(G_k)]_{\text{null}}}{\sqrt{\mathrm{V}[f(G_k)]_{\text{null}}}} \tag{1}$$

Basically, this calculation imagines that $f(G_k)$ is drawn from a Normal distribution, and converts it into units of the standard deviation of that distribution. There are basically three possibilities:

1. If $z = 0$ (or close to it), then $f(G_k)$ is about what we would expect under the null.
2. If $z > 0$ (by some margin, often 2), then $f(G_k)$ is higher than we expected under the null.
3. If $z < 0$ (by some margin, often 2), then $f(G_k)$ is lower than we expected under the null.

Nearly any null model of graphs that we might use, e.g., the configuration model or the Chung-Lu model, will generate a distribution of motif frequencies $\Pr(G_k)$ that is not Normal, meaning a z-score should only be interpreted as a (rough) descriptive statistic, rather than as a test statistic.[8]

1.8.1 Algorithms for counting

There are basically two kinds of algorithms for counting motifs. The key difficulty that all of these algorithm face is accounting for what's called subgraph isomorphism. Two subgraphs are isomorphic if we could permute the names of the nodes, and still have the same motif. In the above example, motif (a) is a triangle, and there's one of them in the left-hand graph. But, we could write down the names of the three nodes that compose it in $3! = 6$ different ways, and hence any algorithm would need to avoid over counting these isomorphisms. (The larger the subgraph, the more isomorphisms it will have.)

1. **exhaustive search**, which is usually very slow, and works mainly for $k < 5$. These algorithms often enumerate all possible listings of the k vertices, and then checks if they have the target set of edges among them. These algorithms run in time $O(n^k)$. They can be made somewhat more efficient if we are smart about recognizing how motifs can be nested, e.g., the motif (b) above contains (a) as a subgraph, and so if there are no (a) motifs in a graph, there can be no (b) motifs either, and we don't need to check for them.
2. **sampling**, which is faster than exhaustive search, but still not scalable; it works mainly for $k < 20$. There are various strategies for sampling, one of which is simply to sample the exhaustive search approach. A different approach is to find some subgraph in G, repeatedly, and to track which ones are found. In this approach, a problem arises, which is that some motifs are easier to find (sample) than others, depending on the method of grabbing a subgraph out of the larger graph G. Fixing this problem requires re-weighting the estimated frequencies in order to produce an unbiased estimate.

[8] A better statistic can be found in Picard et al., "Assessing the Exceptionality of Network Motifs." *Journal of Computational Biology* **15**(1), 1–20 (2008). `http://doi.org/10.1089/cmb.2007.0137`

Supplemental readings

Protein-protein interaction networks:

1. Szklarczyk et al., "STRING v10: proteinprotein interaction networks, integrated over the tree of life." *Nucleic Acids Research* **43**, D447D452 (2015).
   `https://www.ncbi.nlm.nih.gov/pubmed/25352553`

2. Giot et al., "A Protein Interaction Map of *Drosophila melanogaster*." *Science* **302**, 1727 (2003).
   `https://www.ncbi.nlm.nih.gov/pubmed/14605208`

3. Chautard et al., "Interaction Networks: From Protein Functions To Drug Discovery. A review." *Pathologie. Biologie.* **57**, 324-333 (2009).
   `https://www.ncbi.nlm.nih.gov/pubmed/19070972`

Boolean network models:

1. Bornholdt, "Boolean network models of cellular regulation: prospects and limitations." *J. R. Soc. Interface* **5**, S85–S94 (2008).
   `https://dx.doi.org/10.1098/rsif.2008.0132.focus`

2. Grieco et al., "Integrative Modelling of the Influence of MAPK Network on Cancer Cell Fate Decision." *PLoS Comput. Biol.* **9**(10), e1003286 (2013).
   `https://dx.doi.org/10.1371/journal.pcbi.1003286`

Motif discovery:

1. Ciriello and Guerra, "A review on models and algorithms for motif discovery in protein-protein interaction networks." *Briefings in Functional Genomics and Proteomics* **7**, 147–156 (2008).
   `https://www.ncbi.nlm.nih.gov/pubmed/18443014`

2. Ahmed al., "Efficient Graphlet Counting for Large Networks." *IEEE International Conference on Data Mining* (ICDM) (2015).
   `http://nesreenahmed.com/publications/ahmed-et-al-icdm2015.pdf`

3. Grochow and Kellis, "Network Motif Discovery Using Subgraph Enumeration and Symmetry-Breaking." *RECOMB* (2007).
   `http://compbio.mit.edu/publications/C04_Grochow_RECOMB_07.pdf`