Introduction

Motifs are interaction patterns that occur more often in real biological networks than one would expect by chance. Under the reasonable assumption that interactions are generated randomly by mutation, and maintained over long timescales only if they are valuable, enrichment indicates that a motif fulfills a functional need which the biological system has encountered not just once, but many times.

You have already seen two examples of motifs: mutual repression and positive autoregulation (both direct and indirect). Unbiased searches for such motifs detect others, which we can then examine to understand what functionality they may provide.

How are connections identified?

The old method: hand curation (Shen-Orr 2002 and others)

- Combing the primary literature
- Relying on other authors to curate a database, e.g. RegulonDB or BioGRID
- Shen-Orr et al. got to 577 interactions with 116 transcription factors and their targets.

Around the same time, others were already starting to use more high-throughput approaches:

1. Lee et al., 2002 perform chromatin immunopreciptiation (ChIP) to identify binding sites of 106 transcription factors in budding yeast.
   (a) Approach is to add a "tag" – an extension of the protein that folds into a unique and detectable shape – to one transcription factor. They used a ten amino acid sequence called a Myc tag (after the protein from which the sequence was taken). Unfortunately this cloning step remains challenging to automate.
   (b) The researchers then attempted to confirm that the Myc tag did not interfere with the transcription factor's normal functioning. Iron-clad proof of non-interference would be very time-consuming to obtain, so the authors simply check that the Myc-tagged protein is present, which they do via antibody staining. (Of course the Myc tag could still prevent the transcription factor from binding to DNA, but confirming DNA binding would be more challenging.)
      i. Antibodies are generated randomly by the vertebrate immune system. They are four-subunit proteins that have a region which contains random sequence and may form a shape that perfectly complements another molecule. The immune system is able to identify functionally-relevant antibodies and amp up their production.
      ii. Antibodies against the Myc tag can be collected by injecting the Myc tag into a vertebrate and allowing its immune system to respond. Just as when you receive an immunization, the vertebrate begins to produce more antibodies against Myc which can be detected in and collected from serum. (Alternatively, to get a pure sample of a single type of antibody, researchers can immortalize the animal's B cells and identify a single clone that produces the desired antibody.)
      iii. The antibody alone is not visible, but it can be modified to add radioactivity, fluorescence, or enzyme activity that allow the antibody to be detected.

iv. To check whether the Myc-tagged transcription factor is expressed, the yeast cells are fixed and permeabilized (so that the antibody can get in), then a solution containing the antibody is added and given time to find and bind its target, and finally excess unbound antibody is washed away. Any remaining fluorescence signals that that the Myc-tagged transcription factor is present and has bound the antibody.

(c) After confirming that the Myc-tagged transcription factor is expressed, experimenters attempt to determine which DNA sequences the protein binds to.

i. Cells are fixed with formaldehyde to preserve the connection between the protein and whatever DNA it may be bound to.

ii. The cell is lysed (broken apart) and the DNA sheared into tiny fragments. Some fragments will have our Myc-tagged transcription factor still attached.

iii. An anti-Myc antibody is attached to beads, which are placed in a column. The cell lysate is poured over the beads and rinsed. Only the Myc-tagged transcription factors (and any DNA attached to them) remain on the column.

iv. The transcription factors/DNAs are eluted off the column by adding free Myc tags, which compete for binding to the beads. The DNA fragments can then be sequenced (or, back in those days, identified by microarray). If the sequence is near a gene, then that gene is considered a "target" of the transcription factor.

(d) This approach does not discern positive vs. negative regulation. Indeed, DNA binding might have no effect on expression level at all.

(e) About one-third of all yeast genes (2300) were bound by at least one of the 106 transcription factors Lee et al. studied. Found nearly 4000 interactions.

2. A related technique for identifying protein-protein interactions (rather than transcription factor-DNA interactions) is to enrich a sample for all proteins that bind to one protein of interest. This can be done by co-immunoprecipitation (Co-IP) if an antibody is available, or by pull-down if the protein of interest is tagged. The proteins present in the enriched sample can be identified by mass spectroscopy.

3. Most studies of this type focus on interactions between transcription factors and their binding sites. There are also means to identify interactions between proteins but unfortunately these have been especially fraught with reproducibility issues.

(a) The most common method is the yeast two-hybrid. It dates to 1989, but requires two libraries of fusion proteins; this process was not fast/automated/comprehensive until the early 2000s.

(b) The assay is called a yeast two-hybrid because it involves a yeast transcription factor called GAL4 (which you have encountered in Melen et al., 2005).

(c) The assay is carried out in bacteria, which do not contain GAL4. These bacteria have been modified, however, to contain a *reporter construct* consisting of a UAS (GAL4's binding site) upstream of an antibiotic resistance gene.

(d) The GAL4 protein is divided into two halves: a DNA-binding domain and a transcriptional activation domain. Each half folds and functions independently. If a bacterium can bring the two halves together to assemble a functional GAL4 protein, then the antibiotic resistance gene is expressed and the cell lives.

(e) The DNA binding domain is fused to a protein of interest referred to as the "bait." The bait protein is therefore recruited to a UAS binding site in the DNA, but (unless it is a transcription factor) this does not cause antibiotic resistance.

(f) The transcriptional activation domain of GAL4 is fused to another protein, called the prey. The TA domain can bind RNA polymerase, but since this fusion protein can't bind DNA on its own, this is not enough to cause antibiotic resistance.

(g) The screen is conducted on bacteria that all contain the same bait plasmid but contain different prey plasmids.

(h) If the prey binds the bait, then the transcriptional activation domain is recruited to the antibiotic resistance gene promoter, and that bacterium lives. The identity of the prey can be determined later by sequencing.

(i) Mammalian protein-protein interaction trap (MAPPIT) is an analogous method but is based on restoring a signaling pathway.

(j) Unfortunately two studies that each applied this same technique to find interactions between two proteins only had 10% (of 1500) hits in overlap.

(k) A recent attempt to duplicate randomly-chosen singly-annotated interactions (Rolland 2014) found only slightly higher success rates than for negative controls.

(l) Many predictions from Y2H/MAPPIT have even been shown to be inconsistent with structural data.

(m) Number of interactions to check is n choose 2: number of genes in human genome is ≈20,000 which is 200 million.

More recently (Rolland 2014), researchers studied a network of 11,000 interactions that had been annotated multiple times (this is roughly an order of magnitude below the expected total number, but represents reliable data). Highly-studied proteins like retinoblastoma 1 form a dense network as we would expect from investigation bias.

- Systematically studied lesser-known parts of the proteome by Y2H in quadruplicate and in multiple backgrounds.
- Confirmed each interaction in mammalian systems using MAPPIT, fluorescent protein reconstruction in Chinese hamster ovary cell lines, etc.
- Structural data much more consistent with interaction predictions, where available. Also more likely to find reported interactions between proteins that are known to be members of complexes.
- Disease-associated variants are more likely to disrupt protein-protein interactions than synonymous (likely, neutral) variants.

Detection of motifs

- We will assume for now that we have a set of regulatory interactions between transcription factors. We can regard each transcription factor as a nose, and each interaction as a directed edge, in a directed graph. We allow nodes to have self edges.

- To detect enrichment (more of a certain motif than we would expect by chance), we must compare our real graph to our expectations either analytically or through simulation. In either case, we must decide what properties of our network to regard as fixed and which are considered to be random.
- For example, suppose our graph has M edges and n nodes.
  - Our null hypothesis might be that our graph is chosen at random from all graphs where the probability of an edge existing between any two nodes is

$$p = \frac{M}{\binom{n}{2}}$$

    This is called the $G(n, p)$ model. Notice that the number of edges is not fixed: e.g. it is possible, but highly improbable, to have no edges at all. The probability that a given node has k edges (ignoring directionality) is

$$P(k \text{ edges }) = \binom{n-1}{k} p^k (1-p)^{n-k} \approx \frac{(np)^k e^{-np}}{k!} = \frac{\lambda^k e^{-\lambda}}{k!} \tag{1}$$

    which is a Poisson distribution with parameter $\lambda = np$.
  - Alternatively, H_0 might be that our graph is chosen at random from all graphs where the number of nodes and edges are n and M, respectively. In the limit of very large n/M, this model behaves very similarly to the one above.
  - Unfortunately random, false positive interactions tend to make distributions look more Poisson-like.
  - Examination of many high-quality, real-world networks reveals that the distribution of the number of edges per node is often significantly different from Poisson. For example, many networks are *scale-free*, that is,

$$P(k \text{ edges }) \sim k^{-\gamma} \tag{2}$$

  - Distributions 1 and 2 differ in the fraction of nodes expected to have unusually high numbers of edges. A few general consequences of this are:
    * Small-world phenomenon: in scale-free networks, the presence of highly-connected nodes reduces the average distance between any two nodes ($\log n$ vs. $\log \log n$, for $2 < g < 3$). Examples from epidemiology and social interactions.
    * Fault tolerance: random loss of nodes in scale-free networks is less likely to disconnect portions of the graph.
  - Scale-free properties can caused by network growth with preferential attachment.
- In acknowledgment that the distribution of edge numbers within a network may be important, we may wish to compare graphs based on real data to random graphs where the edge distribution is the same, i.e., exactly s_i edges start and e_i edges end at node i.

- This requirement is easily implemented by visualizing the real graph as an adjacency matrix and tallying the rows and columns to determine the number of inputs and outputs to/from each node. New graphs are then generated by choosing a possible input at random and a possible output at random until all inputs and outputs have been used. The resulting random graph has the same number of nodes of each degree as the original, but with a different connectivity.
- Alternatively, we can start from the original graph and simply swap two inputs or two outputs at random until the graph is sufficiently randomized.
- We generate many random graphs to compare to the original. Then, we quantify the number of each motif that we find in our real graph and the average/variance in the random graphs. We can then determine whether our real graph is statistically distinct from the random ones in this way.
- Issues with multiple hypothesis testing and significance.

Results of enrichment analysis

Consider the *E. coli* gene regulatory network (424 nodes, 519 edges in the data set at the time).

- Autoregulation
  - We'd expect this network to have 519/424 = 1.2 self-edges if its edge counts were Poisson-distributed. The real network has 40 self-edges.
  - Roughly two-thirds of transcription factors in *E. coli* are activators, yet 85% of these autoregulatory connections are negative. In other words, negative autoregulation occurs much more frequently than we would expect by chance.
- Feed-forward loop
  - Eight varieties (not including types of promoter logic at the target gene Z.
  - Highly-enriched overall, even relative to degree-preserving random networks.
  - In particular, FFLs consisting only of positive interactions, and incoherent FFLs where the target is initially activated, then repressed, are dramatically enriched relative to all others.
- Interestingly, different motifs are often found in other types of networks (food chains, electronic circuits, the World Wide Web)

Introduction to negative feedback

Many biological processes have evolved within the context of stable internal conditions. The zero-order ultrasensitive responses we studied earlier assumed, for example, that there would be a generous supply of ATP with which to phosphorylate each substrate, that the pH would permit the proteins to stay folded, and many other implicit requirements which could only be considered reasonable because of the cell's active efforts to maintain consistent internal conditions despite a variable environment – a process called homeostasis. Biological systems designed to maintain some variable at a constant value are thus essential for permitting all of the other qualities of life.

Most such systems share a quality in common: the system's output is monitored and used to regulate the system's activity through negative feedback. (Draw example block diagram, introducing the idea

of an open-loop vs. closed-loop control system.) More generally, some processing might be done to the output signal before it is fed back [draw another box]. For example, we will soon compare the value of responding proportionally to the output, to the derivative of the output, and to the integral of the output.

We do not yet have a mathematical framework for understanding feedback of this type. In the remainder of this lecture we will introduce Laplace transforms, which will greatly aid our analysis of systems with feedback. For practice, you'll use Laplace transforms to analyze an open-loop system – activation of photoreceptor neurons by photons – on problem set four.

Introduction to LTI systems

Let's return to our simple open-loop system, which we envision as a black box to which we can supply an input $x(t)$ and measure an output, $y(t)$. [Draw a block diagram.] To describe how the system works, we would like to find a mathematical representation for the relationship between $x(t)$ and $y(t)$. You can imagine how we might approach this experimentally, by supplying simple inputs like pulses, step functions, or ramps and measuring the resulting output timecourse.

It would make our life much easier if we could make a few simplifying assumptions about the system. For example, it would be great if we got the same output every time we supply the same input: this system property is called *time invariance* and its utility is probably clear. It would also be handy if the system is *linear*, i.e., once we have determined the outputs $y_1(t)$ and $y_2(t)$ for two inputs $x_1(t)$ and $x_2(t)$, then we should be able to predict that the output for $ax_1(t) + bx_2(t)$ is $ay_1(t) + by_2(t)$. Why is this second property so useful?

Recall that the Dirac delta distribution $d(t)$ is defined to have integral unity and the properties:

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

Any input function $x(t)$ can be represented as a time-offset combination of (Dirac) delta distributions. So if we know the system's response to a delta distribution input, and the system is linear, then we can predict the system's output for any input! The response to the Dirac delta distribution input deserves its own name – the *impulse response function* – and can be used to fully describe a linear, time-invariant (LTI) system.

Convolution

Consider first a system with no feedback and suppose that we have determine the impulse response function, $h(t)$ (recall that this is the system's response to a DDD input). What is the system's output $y(t)$ for an arbitrary input $x(t)$?

Suppose as an example that we would like to calculate $y(5)$, i.e. the output at time $t = 5$ seconds. Imagine breaking the input down into small"bins" in the range 0 to 5 seconds. We will calculate how input from each bin contributes to the output at $t = 5$.
Consider the bin centered at τ ($0 \leq \tau \leq t$). Since the bins are small, we can think of the input in each bin as instantaneous, and represent it by a multiple of the delta function centered a τ: $x(\tau)\delta(t - \tau)$. The impulse response function tells us that the contribution from this bin to the output at time t will be $x(\tau)h(t - \tau)$. If we sum up the contributions from all bins,

$$y(t) = \int_0^t x(\tau)\, h(t-\tau)\, d\tau = x(t) * h(t)$$

This type of integral is a convolution of x and h (represented by the $*$ symbol here). Similarly, for our system with (unmodified) negative feedback:

$$y(t) = \int_0^t \left[x(\tau) - y(\tau)\right]\, h(t-\tau)\, d\tau = \left[x(t) - y(t)\right] * h(t)$$

Notice that $y(t)$ occurs on both sides of the equation – it could be challenging to compute if not for a method we now introduce, called the Laplace transform.

Laplace transforms

The Laplace transform of a real function $f(t)$ defined for $t > 0$ is:

$$\mathscr{L}\left[f(t)\right] \triangleq \hat{f}(s) = \int_0^\infty \exp\left(-st\right)\, f(t)\, dt$$

The result of the transform is a function of s, a variable in frequency space (inverse time). An inverse Laplace transform also exists, but is difficult to implement; usually we just look up $f(t)$ for $\hat{f}(s)$, and even $\hat{f}(s)$ for $f(t)$, in a reference table to save time However, here is one example of how the calculation is performed. Suppose the original function is:

$$f(t) = \begin{cases} e^{-\alpha t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

In this case we can plug in to the definition and integrate:

$$\begin{aligned} \mathscr{L}\left[f(t)\right] &= \int_0^\infty e^{-st} e^{-\alpha t}\, dt \\ &= -\frac{1}{s+\alpha}\, e^{-(s+\alpha)t}\Big|_0^\infty \\ &= \frac{1}{s+\alpha} \end{aligned}$$

This particular Laplace transform is worth memorizing because exponentially-decaying inputs are commmon in systems analysis.

Laplace transforms convert derivatives and integrals into algebraic expressions. For example, suppose we would like to calculate the Laplace transform of df/dt. We make use of integration by parts:

$$\int_0^\infty u\, dv = uv\Big|_0^\infty - \int_0^\infty v\, du$$

Plugging in with $dv = \frac{df}{dt}\, dt$ and $u = e^{-st}$,

$$\begin{aligned}\mathscr{L}\left[\frac{df}{dt}\right] &= \int_0^\infty e^{-st}\frac{df}{dt}\,dt \\ &= e^{-st}f(t)\Big|_0^\infty + s\int_0^\infty f(t)e^{-st}\,dt \\ &= -f(0) + s\hat{f}(s)\end{aligned}$$

Likewise, it is easy to show that:

$$\mathscr{L}\left[\int_0^\infty f(t)\,dt\right] = \frac{\hat{f}(s)}{s}$$

This makes it easy to find solutions to systems of differential equations, since we can "eliminate" variables in frequency space through substitution. The Laplace transform also works miracles on convolutions:

$$\mathscr{L}\left[x(t) * h(t)\right] = \hat{x}(s)\hat{h}(s)$$

which will turn out to be very helpful when calculating the system's output. When we begin on Friday, we'll use these properties to examine how simple systems with negative feedback behave.