- This section has focused on feedback as a method for system control. In the last few lectures we've described different forms of negative feedback and their potential advantages and shortcomings.

- Earlier in the course, we used a very different approach to analyze positive feedback systems and their ability to generate switch-like behaviors.

- Soon we will move on to discussions of oscillators and their roles in biological clocks, but first, we briefly discuss an entirely different form of control important for living systems: feedforward loops.

Feedforward loops and derivative control

- How can we detetermine and respond to the rate of change in the input? We have seen that negative feedback is one possible route in the example of perfect adaptation in chemotaxis. Increase in receptor activity increases [CheY-P] and [CheB-P]; the latter demethylates receptors to lower receptor actvity, forming a feedback loop. The result is a pulse of unusually high or low tumbling frequency after a change in ligand concentration.

- Another option for detecting changes in the input is with feedforward control. Although any system with two or more branches leading from input to output could be considered a feedforward system, we will talk about a simple type of FFL made of three components: an input X that regulates both Y and Z, where Y in turn regulates Z. FFLs of this type can detect changes in input directionally, and in some cases are sensitive to the magnitude of the change.

- While the three players could be any biological molecules and activity could be regulated in many ways, we'll assume that X and Y are transcription factors, and that regulation within the FFL occurs at the level of gene expression[1].

- We will consider transcription factors that act as activators, repressors, or even both (in the case of the input, which has two targets). We will also consider that the promoter of the output can use "AND" or "OR" logic with its two inputs. All told, this makes for sixteen different feed-forward loops (FFLs).

Expected vs. observed frequency of FFLs

- One of the primary arguments for the fact that these feed-forward loops perform a biologically-useful function is that there are more of them in natural gene regulatory networks than we would expect by chance.

- As we saw back in lecture 12, this argument hinges on the assumption that regulatory connections between transcription factors appear randomly through mutation (oversimplifying somewhat, a new binding site for TF1 appears in the promoter of TF2), and if the new regulatory connection somehow increases the fitness of the organism, then that organism will likely leave more offspring and eventually the whole population will have that connection; is new mutations that remove the connection appear, they will be selected against. On the other hand, if the new connection is non-functional, then there will be nothing to help the connection spread; there is a small chance it would reach high frequency by drift, and if it somehow did, there would be nothing to prevent its loss through a subsequent mutation. So if we see

[1] This choice is not just for the sake of simplicity: gene regulation introduces time delays that are easy to measure experimentally and thus may be relevant biologically.

more FFLs than we expect by chance, it seems likely that selection on their utility is helping them to accumulate.

- We saw then that there were some challenges in selecting the "random" networks against which the real one is compared. You grappled with this issue on question one of problem set five. In particular, for many biological networks it is a poor assumption that the number of edges per node is Poisson-distributed (as predicted by the Erdös-Rényi model). More commonly, the distribution of edges per node follows a power law (i.e. the network is scale-free).

- We now describe in a little more detail why this is so. We have seen already that for large n and small, equal probability p of an edge occurring,

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

- Suppose instead we begin with a small network with m_0 vertices and no edges, and at every time step add a new vertex with $m \leq m_0$ edges to existing vertices. The new edges are chosen to preferentially attach the new vertex to vertices that already have more edges than usual. This is done by setting

$$P(\text{ new vertex connected to vertex with degree } k_i \text{ in time step } t) = \frac{k_i}{\sum_j k_j} = \frac{k_i}{2t}$$

(During the first time step, we assign the edges with equal probability.)

- After t time steps, we have a network with $t + m_0$ vertices and mt edges.

- Under this model, the rate at which a vertex acquires edges is

$$\frac{dk_i}{dt} = \frac{k_i}{2t} \implies k_i(t) = ct^{0.5}$$

- Our initial condition requires that the connectivity of the node is m at the time it is added (t_i), so:

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{0.5}$$

- To find the distribution of the k_is, consider the probability that $k_i(t)$ is less than some value k:

$$\begin{aligned} P(k_i < k) &= P\left(t_i > \frac{m^2 t}{k^2}\right) \\ &= 1 - P\left(t_i \leq \frac{m^2 t}{k^2}\right) \\ &= 1 - \frac{m^2 t}{k^2 (t + m_0)} \end{aligned}$$

- This gives the cumulative probability distribution. To calculate the probability density, we take the partial derivative with respect to k:

$$P(k) = \frac{\partial P(k_i < k)}{dk} = \frac{2m^2 t}{k^3 (t + m_0)} \approx 2m^2 k^{-3} \text{ at long times}$$

  which is of the form $P(k) \sim k^{-\gamma}$ which you were promised for scale-free networks.

- For this reason (and because many networks don't follow perfect power laws, either), a good approach to generating random networks comparable to a real one is to reorder the edges while keeping the distribution of degrees the same (e.g. swapping rows and columns in an adjacency matrix).

- This fact notwithstanding, we can gain some intuition for how frequently we would expect to see a feed-forward loop (FFL) by chance under the assumption that all edges are equally likely.

- Suppose that each transcription factor gene in the organism represents a node, and the regulatory connections between them are edges (directed edges, since each connection has an agent and a target – which might be the same!).

- In *E. coli*, for example, there are about $N = 400$ transcription factors and $E = 500$ connections between them. If all edges are equally likely, what is the probability in terms of N and E that any two transcription factors chosen at random share an edge? [Audience response – may require reminder that self-connections are possible]

- There are N ways to choose the first node, and N ways to choose the second, so N^2 possible edges. Of these, only E actually exist, so

$$p = \frac{E}{N^2} \approx 0.003 (\text{for } \textit{E. coli})$$

- If our network has N transcription factors, how many ways can we choose three nodes to be the input, intermediate, and output (specifically) of a FFL? [Note that we no longer allow the same node to be used twice, as this would not fit out motif.]

  Number of ways to choose input, intermediate, and output: $N(N-1)(N-2) \approx N^3 = 6.4\times10^7$

- Notice that there is no combinatoric term here, since we assume that the order of choice is important. The probability that three randomly-chosen nodes have exactly three edges between them is given by the binomial distribution (the same one you would use to calculate the probability of getting five heads in ten coin flips):

$$P(\text{exactly three edges}) = \binom{6}{3} p^3 (1-p)^3 = 120 \cdot 3 \times 10^{-8} \times 1 = 3.6 \times 10^{-6}$$

- What is the probability that the edges are hooked up properly for the input, intermediate, and output we've defined?

$$P(\text{hooked up correctly}) = \frac{3}{6} \cdot \frac{2}{5} \cdot \frac{1}{4} = \frac{1}{20}$$

- Altogether, this means that the expected number of FFLs in the *E. coli* gene regulatory network is[2]:

$$6.4 \times 10^7 \cdot 3.6 \times 10^{-6} \cdot \frac{1}{20} = 11.5$$

It turns out that for degree-preserving randomly-generated networks, the predicted number is 7.

- The number of annotated FFLs in *E. coli* is significantly greater: 42. This suggests that the FFLs have a general function useful enough to be selected for in many different contexts.

Some FFLs are more common than others

- Every directed edge in the feed-forward loop can be either a repressive or an activating interaction. Four of the choices result in the change in output activity being the same through both loops: these are called *coherent* loops while others are *incoherent*.

- You can see that some FFLs require the same transcription factor to be both an activator and a repressor. Is this possible? [Audience responds.] Yes: for example, a transcription factor that normally activates nearby genes can inhibit RNA polymerase by steric hindrance (sitting in its way). So all of these FFLs are possible, and the probability of each type of FFL occurring will depend on the likelihood of an inhibitory vs. activating interaction (approx. 40% vs. 60%).

- Even so, some FFLs are much more enriched than we would expect by chance. The two we will describe today are the coherent feed-forward loop in which all interactions are activating (C1-FFL) and the incoherent feed-forward loop in which the input directly activates the the output but also activates a repressive intermediate (I1-FFL). Together they make up more than 70% of FFLs in the *E. coli* gene regulatory network.

Function of the I1-FFL

- These represent about 30-40% of known FFLs. The input directly activates the output, but also activates an intermediate which will ultimately inhibit the output. Suppose for now that the promoter of the output gene uses AND logic: in order for the output to be expressed, the input must be "on" and the intermediate must be "off."

- How should we think of on/off in the context of a gene regulatory network? For simplicity's sake we assume there is some threshold concentration above which the transcription factor is usually bound to the target's promoter, and below which it is usually not.

- A good choice for this threshold concentration might be the binding constant, K_m, of the transcription factor's binding site: if the transcription factor binds to its target promoter cooperatively, as many do, then the binding curve will be sigmoidal and our approximation to switch-like activity will be reasonable.

- This leads to the following differential equations for the rate of change in Y and Z.

$$\begin{aligned} \frac{dY}{dt} &= k_Y \Theta\left(I - K_X\right) - \gamma_Y Y \\ \frac{dZ}{dt} &= k_Z \Theta\left(I - K_X\right) \Theta\left(K_Y - Y\right) - \gamma_Z Z \end{aligned}$$

[2] This differs from the value in the reading because we have kept some O(1) terms neglected in the book chapter. I also made a math error in the numerical solutions on the board in lecture; I had divided by 30 instead of 20.

where $\Theta(x)$ represents the Heaviside function:

$$\Theta(x) = \begin{cases} 0 & : x \leq 0 \\ 1 & : x > 0 \end{cases}$$

- Suppose the input is off and has been for a long time, so that the system has reached steady-state. What are the concentrations of the two other transcription factors? [Audience response.]
- Now suppose that from this state, we turn on X and leave it on. These equations then become:

$$\begin{aligned} \frac{dY}{dt} &= k_Y - \gamma_Y Y \\ \frac{dZ}{dt} &= k_Z \Theta\left(K_Y - Y\right) - \gamma_Z Z \end{aligned}$$

- At first, Y and Z both rise. But when Y crosses its threshold for activity, production of Z turns off, and degradation eventually brings it back down to zero. Meanwhile Y just approaches its steady-state value $k_Y/\gamma_Y > K_Y$. We have generated a pulse in Z in response to a sudden rise in X.
- Now imagine we did the opposite, switching X from "on" to "off". Would this generate a pulse in Z? [Audience response.]
- We therefore say this pulse generator is sign-sensitive. If we relax our simplifying assumptions involving thresholds, the height of the pulse could scale with the rate of change in X: a necessary feature for a differentiator.

Incomplete repression

- What if repression in this system were incomplete?

$$\frac{dZ}{dt} = k_0 + k_1 \Theta\left(X - K_X\right) - k_2 \Theta\left(Y - K_Y\right) - \gamma_Z Z$$

- In this case the expression level would pulse high, then fall back to a non-zero steady-state value. This could be useful for helping a gene turn on faster while keeping the steady-state value the same.
- This activity is seen in the galactose response system in *E. coli*. Here, CRP detects the accumulation of cAMP (a sign that glucose has been depleted), then activates both GalS and other parts of the Gal operon. This allows the galactose response genes to turn on fast while maintaining a reasonable steady-state expression level. (Simple negative autoregulation can achieve the same benefit.)

What about I4-FFL?

- I4-FFL, like I1-FFL, serves as a pulse generator and a response accelerator. So why is it so much less common?
- We have claimed that it is not necessarily because X must function as both an activator and a repressor in this circuit.

- Alon points out that the I1-FFL can be generalized to incorporate additional signals at the level of the intermediate. For example, we might require that Y be phosphorylated before it can affect gene transcription, then make that phosphorylation contingent on another signaling pathway to shape the response.
- By contrast, in the I4-FFL, Y is being diluted so it may be difficult to use post-translational modifications to modulate the activity of Y.

Function of the C1-FFL

- What about the C1-FFL, which makes up roughly 40% of all feed-forward loops? Suppose first that the output's promoter uses AND logic. Then the dynamics are described by:

$$\begin{aligned}\frac{dY}{dt} &= k_Y \Theta\left(X - K_X\right) - \gamma_Y Y \\ \frac{dZ}{dt} &= k_Z \Theta\left(X - K_X\right)\Theta\left(Y - K_Y\right) - \gamma_Z Z\end{aligned}$$

- If the system has been off for a while, the intermediate and output will start at zero. The output begins to accumulate only after a delay that corresponds to the intermediate reaching its functional concentration. How long is this delay?

$$\begin{aligned}\int \frac{dY}{k_Y - \gamma_Y Y} &= \int dt \\ -\frac{1}{\gamma_Y}\log\left(k_Y - \gamma_Y Y\right) &= t + c_0 \\ k_Y - \gamma_Y Y &= c_1 e^{-\gamma_Y t} \\ Y(0) = 0 &\implies c_1 = k_Y \\ Y(t) &= \frac{k_Y}{\gamma_Y}\left(1 - e^{-\gamma_Y t}\right) \\ Y(\tau) = K_Y &= \frac{k_Y}{\gamma_Y}\left(1 - e^{-\gamma_Y \tau}\right) \\ e^{-\gamma_Y \tau} &= \frac{k_Y - K_Y \gamma_Y}{k_Y} \\ \tau &= \frac{1}{\gamma_Y}\log\left(\frac{k_Y}{k_Y - K_Y \gamma_Y}\right)\end{aligned}$$

- However, if the input is suddenly shut off, the output will start to decay immediately. This is called a *sign-sensitive delay*. Similarly, if the output promoter used OR logic, there would be a delay in turning the output off but not in turning the output on.
- Why would a time-sensitive delay be useful in a living organism? [Audience response.] Noise can generate both "on-to-off" and "off-to-on" responses. We can easily imagine that some of these decisions will be more costly than others.
- For example, a dormant seed should not germinate in February in response to one uncharacteristically warm day because it would almost certainly freeze and die; by comparison delaying

the growing season slightly because of one cold day in April would have relatively minor disadvantages.

- Examples of both AND and OR logic are found in biological systems. An example of AND logic is found in the *E. coli* arabinose response pathway. *E. coli* prefer the sugar glucose to the sugar arabinose: when glucose is present, they do not waste energy making the machinery needed to degrade arabinose, even if both sugars are present at the same time.

- When glucose is depleted, an intracellular second messenger called cAMP begins to accumulate and acts as the input signal to the C1-FFL that turns on arabinose. (The intermediate in this feed-forward loop requires arabinose to be fully activated – that way, the system does not make arabinose response genes if no arabinose is present.)

- The delay in beginning to express arabinose response genes prevents wasted energy if the cAMP signal is noisily on when glucose is still present. However, as soon as the prefered glucose appears, arabinose response genes stop being made.

- What about OR logic? There is a very interesting case, slightly more complicated, in the *E. coli* flagellar pathway. We have seen that *E. coli* use their motility through chemotaxis to seek out better places to live. Once they have found a very hospitable environment, they stop making flagella, which are very costly in terms of production resources (each is about ten times the length of the cell) and operating costs (making them spin). It takes a long time to start making a flagellum again so it makes sense to place the longer delay at the "on-to-off" transition, and so it is in this natural system.

- This is not the only contribution of the FFL to the efficiency of flagellar synthesis in *E. coli*. About six operons are required to make the flagella: all of these "outputs" respond to the same input and intermediate in a generalization of the FFL called a multi-output FFL [draw it: X=FlhDC, Y=FliA and class 2 operons, Z=class 3 operons].

- By tuning the kinetics at each promoter, the cell has managed to control the *order* at which each component's concentration rises to a (presumably) functional concentration. [But N.B. – this effect was reported by Kalir et al. for average expression levels within a population and bursting is the reality within individual cells – Mark Kim, p.c.]

- Proteins which will form the base of the flagellum appear on average before the proteins that will form the shaft, and so forth. When the flagellar genes are turned off, the base proteins turn off first and the shaft genes later. This is called *First-In, First-Out* or FIFO ordering and would not be possible to attain through direct regulation. By contrast, simple regulation would have required that the genes which turn on last would turn off first (LIFO).

- How does the flagellar system achieve this FIFO ordering? The order at which the class 3 operons are turned on is expected to be set by the kinetics of activation by the inputs FlhDC, which may be different at each promoter (for example, there could be more binding sites, or they could have imperfect recognition sequences). The order at which the genes are turned off can instead be governed by the kinetics of activation by the intermediate, FliA.