**Lab 5: Gene Circuits**

# The Dynamic Process of Gene Expression

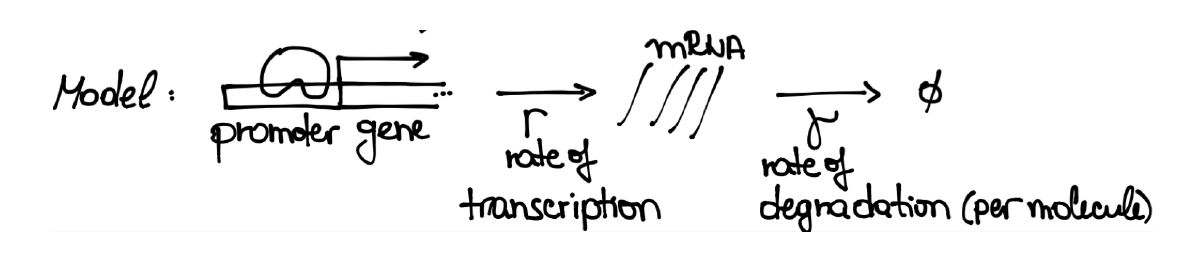
There’s a lot more to genetics than just whether or not you possess a certain gene! From your introductory biology courses, you know that genes are information-storage devices—the function they correspond to is carried out by a specific protein that they produce. These proteins are manufactured by ribosomes, but first an intermediate is created, which is called a *messenger RNA* (mRNA). A cell might produce mRNA from a particular gene at a certain rate, *r*, and this mRNA is destroyed at another rate, *γ*. In many cases, proteins called transcription factors bind to DNA regions called *promoters* just upstream of the gene to regulate the production rate (see Fig. 11).

Figure 1: Genes are transcribed (mRNA is produced) at a rate *r*. These mRNA molecules then break down at a rate *γ*. The transcription rate can increase when a transcription factor binds to the promoter.

Naturally, a quantitative biologist would want to have an equation that can predict the average level of mRNA produced by a given gene. The way that the number of mRNA molecules changes in time (its *dynamics*) can be modeled by the following simple equation:

where *m* is the average number of mRNAs at a given time. Why is *γ* multiplied by *m*, while *r* is not?

R doesn’t depend on the number of mRNA transcripts, since it’s what is doing the transcribing. does depend on the number of transcripts, since they’re being degraded.

In this model, what is the value of *m* when *dm/dt = 0*? This is referred to as the *steady-state solution*.

If dm/dt = r-m =0, then r = m and m=r/

1Kondev J, “Gene Circuits.”

1. **Simple Gene Regulation in *E. coli***

*E. coli* is a bacterial organism that serves as a *model* for many basic biological questions. For *E. coli* typical values for the parameters in the previous section are

*r* = 0*.*3 = *γ*

In this section, you are going to use the simple differential equation introduced in the previous section to model *E. coli* transcriptional dynamics with the help of ode45.

Open the MATLAB m-file “transcription.m” and fill in the differential equation for the transcriptional model in Block 1. Paste your code below.

dy = @(t,y) [r - g\*y];

Fill in the code to define the values of the parameters *r* and *γ* (we will use g for *γ*). Then add a line to define the initial condition

*m* = 0*.*5 at time zero

Solve the differential equation. The plot that you produced will level off (approach an *asymptotic* value). Does it match your expected steady-state value?

Steady-state value = r / = 0.3/0.3 = 1

Now, simulate transcription for two new initial conditions, *m*(0) = 1*.*5 and *m*(0) = 0*.*1. Based on your answer to the previous question, what do you expect to happen in both cases?

All approach the steady-state value of 1, since r and haven’t changed.

In addition to the steady state value, we often care about how quickly the system reaches steady state. Before running the simulations, which parameter(s) do you think will control how quickly the system gets to steady state?

Repressor and/or activator genes will control transcription rate r, which will in turn affect the degradation rate .

1. **Measuring timing**

A standard way to quantify how fast the system gets to steady state is by measuring the time that it takes to get halfway there. In Block 2, we will do this as we vary the transcription rate *r*.

First, read the block of code. What will be stored in the variable *tPastHalfMax*? (You can run the code first, and then inspect what is stored in this variable if that helps you figure it out).

tPastHalfMax is a subset of timepoints within all timepoints for which the average mRNA transcript number is half the maximum number.

Now execute the block of code and note the result. Does *tHalf* depend on the transcription rate *r*?

No, it’s independent.

Working based on Block 2, fill in Block 3 to determine how *tHalf* depends on the degradation rate. Paste your code from inside the for loop below. (Hint: you can copy and paste the code from inside the loop in Block 2. Then you will need to make one key change so that the degradation rate is properly adjusted each time through the loop.)

for i=1:length(gvals)

g=gvals(i);

dy = @(t,y) r - g\*y;

Does *tHalf* depend on the degradation rate? Describe the relationship between *tHalf* and the degradation rate.

tHalf does depend on degradation rate.

At which degradation rate (high or low) does the system approach steady state the fastest?

High mRNA degradation rate minimizes time to reach steady state (implies the steady state is low).

# A simple gene circuit

We will now consider a simple gene circuit. This circuit will have two genes, each of which encodes a transcription factor that represses the other gene. To model this system, we will need to include additional processes – *translation* and *binding* of the transcription factors to the promoters. We will assume that the binding and unbinding reactions occur on a much faster timescale than transcription and translation. Under that assumption, we can model the binding reactions with equilibrium equations (rather than using differential equations). We will still use differential equations to model the mRNA and protein levels.

What do you think will be the steady state output of this system? (this is just a thought question – will one gene dominate? Or will the two genes balance each other?)

It could depend on the relative translation and binding speeds of the two gene, or if they’re about equal, we might expect oscillation.

To build our model, first we will make a *function* to compute the fraction of promoters bound at each possible concentration of repressor protein. In this case, the concentration of DNA promoter sites (2 per cell) will typically be much lower than the concentration of repressor molecules, so we can make the approximation that [R] = [RTotal]. Doing some algebra working from the Kd equation we saw in lecture, you could then find the following expression for the fraction of promoters bound:

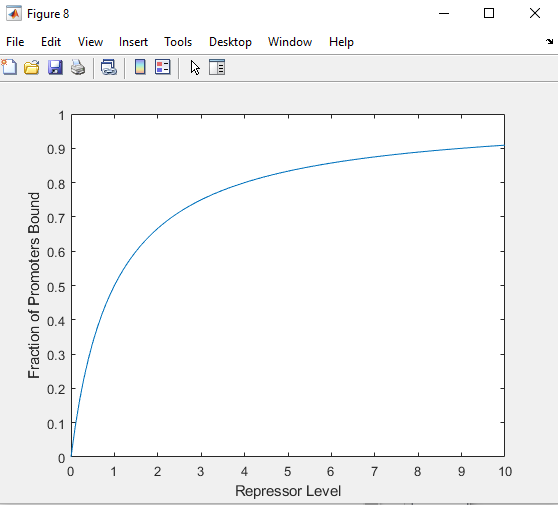
Complete the line below to define a MATLAB function using the @ symbol to compute the fraction bound if *x* is the concentration of repressor protein. Your function should depend on *x* and *Kd*. You should also use “./” instead of “/” so that the function can work on vector inputs. (That could save you some headaches when you try to run the block.)

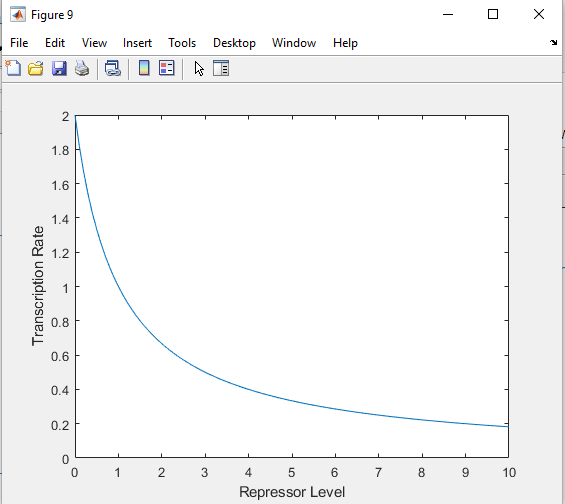
fractionBound = @(x) [x./(Kd+x)]

Assuming that the transcription rate in the basal (unrepressed state) is *r0* and the transcription rate in the repressed state is *r1*(which is less than r0), complete the line below to define a function to compute the transcription rate (*rInst*) as a function of repressor concentration *x*. You should use the function *fractionBound* that you just defined above in your answer. (Hint: what fraction of the promoters should be doing transcription at the rate r1?)

rInst = @(x) r0\*(1-fracbound(x)) + r1\*(fracbound(x));

Edit Block 5 to insert the code for the two functions you just defined here. Run the block and view the two plots. Paste in a screen shot of each plot.





We can write the differential equations for the two mRNA and two protein species, as we have done in class and above. Most of the pieces will be the same. However, now our positive terms for the mRNA equations will be different. Instead of r, we should now have *rInst*, but you need to choose the right input for *rInst* for each equation. What should be the positive term (production rate) for the derivative of the level of mRNA 1?

RInst (y(1))

# Plotting the results

Fill in the missing terms in the differential equations in Block 6 and execute the block. Why does the plot command use *y(:,[2 4])* as an input? What is being plotted in the graph?

Columns 2 & 4 are repressors 1 and 2. Plot is of final level of protein 1 (column 2) and final level of protein 2 (column 4).

Look at the resulting plot. Does it match what you expected?

I didn’t know what to expect, but the two genes converge in their final protein product levels.

Try adjusting the *Kd* for binding between repressors and promoters. Can you make the system end with different protein levels for the two repressors? (If it looks like the two proteins end up at different final values, try increasing the length of tspan to simulate a longer time period.)

They still converge eventually.

Describe why you think the system always reaches this steady state.

The rates of repression between the two genes are roughly equal.