Biological Physics - NOTES
modules F,G (lectures 19-26)

Pietro Cicuta and Diana Fusco

Experimental and Theoretical Physics
Part III
Michaelmas 2021

Notes version: v0.05
Release name: Evolving Emergence

Physics of regulation: Reactions and Stat Mech of promoters

1

1.1 Modeling protein production with ODE

Ordinary differential equations can be used to describe chemical reactions inside the cell.[1] Molecular interactions between protein molecules, other small molecules, and DNA binding sites can turn on and off the activities of proteins and genes. These regulatory interactions combine into complex regulatory networks that ultimately control how cells behave. Here, we will use ordinary differential equations (ODEs) to describe how these regulatory networks work. ODEs provide a powerful tool for predicting how a regulatory network that is wired in a particular way will behave inside the cell. We will consider in this section two rather simple but very important examples (an unregulated gene and a negatively autoregulated gene), but the same methods are used to analyse much more complicated networks with many tens of genes and proteins.

The interior of cells is complicated: eukaryotic cells contain different cell compartments (e.g. the nucleus), and the contents of these compartments can also be organised in complicated ways. Prokaryotic cells, such as bacteria, don't have compartments but they are highly packed with proteins and DNA, and some proteins tend to occupy specific regions of the cell.

Although this spatial structure probably plays an important role in the ways in which cells function, we can understand many aspects of cell regulation without taking it into account. Here, we will make the important assumption that the interior of the cell (or a particular cellular compartment) is "well mixed" (this will not always be the case!)

[1] credit for lectures on ODE and noise modeling: Dr Rosalind Allen, Univ. Edinburgh, through IoP Biological Physics online teaching material

General intro to reaction ODE

Suppose that when an A molecule collides with a B molecule, the two can react to produce a molecule of type C. Starting from a mixture of A and B, we would like to know how many C molecules will have been produced at time t. We suppose that in a small

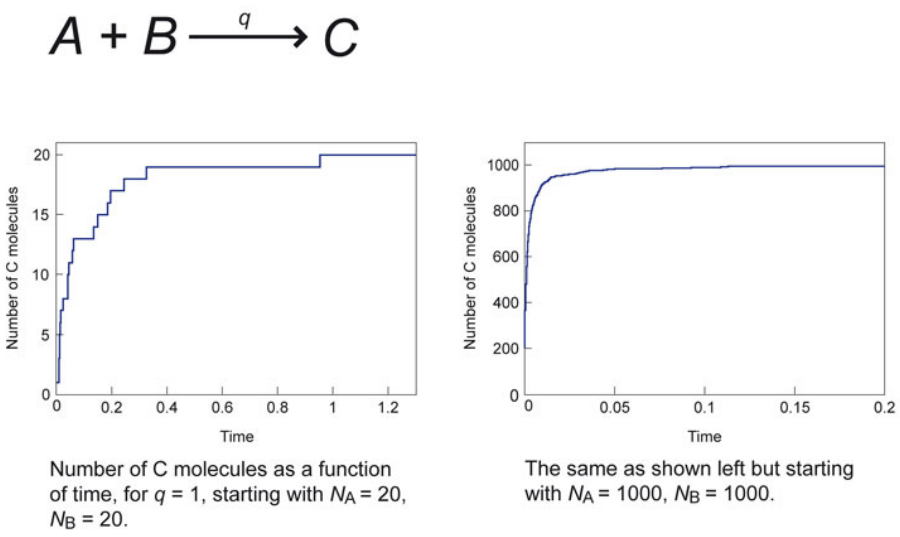

Fig. 1.1 Simulations of simple chemical reaction.

interval of time dt, the probability of a C molecule being produced is qN_AN_B/V, where V is the volume of the system, N_A is the number of A molecules and N_B is the number of B molecules (the probability scales with $1/V$ since a pair of A and B molecules will be less likely to meet each other in a larger volume). We can then write

$$\text{source} \xrightarrow{\text{q}} \text{P}$$

The constant q is called the rate constant.[2]

Figure 1.1 shows the output of a numerical simulation of the reaction

$$\text{A} + \text{B} \longrightarrow \text{C}$$

In these simulations, we have assumed that the volume, V, is set to 1. In the left-hand plot, we can see that the number of C molecules (N_C) increases over time, and that the C molecules are produced at random points in time, whenever an A and a B molecule happen to collide. This randomness can be important if there are only a small number of A and B molecules, and we will return to this later. The right-hand plot shows the same reaction, but with many more A and B molecules. In this case, many collisions happen in a small time interval and the plot for the number of C molecules versus time is much smoother. In fact, we can assume that the number of A, B and C molecules (per unit volume) change continuously with time. This is an important assumption because it allows us to write ODEs to describe how the system changes with time. The variables in these ODEs are the concentrations (number per unit volume) of the chemical species,

[2]The usual symbol for a rate constant is k, but there are several rate constants in this lecture, so we are using q for this one to avoid having several different constants all called k.

in this case A, B and C, which we denote c_A, c_B and c_C (e.g. $c_A = N_A/V$). For example, the set of ODEs that represents the reaction of A and B to produce C
$\mathrm{A} + \mathrm{B} \xrightarrow{\mathrm{q}} \mathrm{C}$
is

$$\begin{aligned} \frac{dc_A}{dt} &= \frac{dc_B}{dt} = -qc_a c_B \\ \frac{dc_C}{dt} &= qc_a c_B. \end{aligned}$$

It's important to note that because this is a second order or bimolecular reaction (it involves two reacting molecules), the dimensions of the rate constant are (concentration^{-1})(time^{-1}). We also need to specify initial conditions, e.g. $c_A(0) = c_B(0) = c_0$ and $c_C(0) = 0$.

Application to protein production

We can use the same ordinary differential equation methods to understand how cells control the production of protein molecules from their genes. Here, we are interested in how the concentration, c_P, of a specific protein molecule, P, changes with time inside the cell. Protein P is produced from its gene, gP, by transcription (to make messenger RNA) followed by translation (to make an amino-acid chain) and protein folding. We could model all of these processes in detail but for now let's just suppose that protein P is produced at a constant rate, k, as long as the gene, gP, is active. This reaction is zeroth order: the protein P is created at a constant rate that does not depend on any other variables in the model. The dimensions of the rate constant for this reaction are therefore (concentration)(time^{-1}).

We write this as a chemical reaction,
$\mathrm{source} \xrightarrow{\mathrm{k}} \mathrm{P}$
In this reaction, the "source" is actually the gene, gP, plus the whole machinery of transcription and translation. Here we just put this into a 'black box' and assume that protein P is produced at a constant rate.

Protein molecules are also removed from the cell; This could be because another protein molecule actively degrades them or because the cell is growing and dividing into daughter cells (and every time the cell divides, a given protein molecule has a chance of being lost). For now, let's just assume that there is a fixed probability per unit time, μ, that any given molecule of P is removed. We can also write this as a chemical reaction,
$\mathrm{P} \xrightarrow{\mu} \mathrm{sink}$
This is a first-order or unimolecular reaction: a single molecule of P reacts. For unimolecular reactions, the dimensions of the rate constant are (time)$^{-1}$. The "sink" here is another black box; P

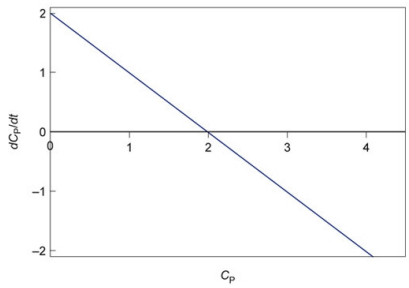

Fig. 1.2 Rate of change of protein concentration.

might have been removed into a daughter cell or it might have degraded into unspecified products.

Combining the constant rate of production, k, and the constant rate, per molecule, of loss, μ, we can write a differential equation for the rate of change of the concentration $c_{\rm P}$ of P molecules:

$$\frac{dc_P(t)}{dt} = k - \mu c_P(t). \tag{1.1}$$

We can tell a lot about the system without actually solving this equation. Figure 1.2 shows the rate of change, $dc_{\rm P}/dt$, plotted for different concentrations of protein, $c_{\rm P}$, for parameter values $k = 2$ and $\mu = 1$. When the concentration of protein is small ($c_{\rm P} < k/\mu$), the rate is positive. This means that there will be net protein production ($c_{\rm P}$ will increase). However, when the concentration of protein is large ($c_{\rm P} > k/\mu$), $dc_{\rm P}/dt$ is negative. This means there will be a net loss of protein. We can also see that for $c_{\rm P} = k/\mu$, $dc_{\rm P}/dt$ is zero. When the protein concentration reaches this value, there will be no net change: production balances removal. This is the steady-state protein concentration, $c_{\rm P}^{(ss)}$.

Steady-state concentrations are a very important property of regulatory networks, and quite often this is all that people focus on when they study a model for a particular regulatory network.

The value of $c_{\rm P}^{(ss)}$ depends on both k and μ. If protein removal is due to cell division and if the average time between cell divisions (the cell cycle time) is τ, then

$$\mu = \frac{\ln 2}{\tau}. \tag{1.2}$$

For the bacterium *E. coli* on a good food source, τ is about 30 min, so μ is about 0.02/min. Protein production rates, k ,vary greatly, from virtually zero to about 50/min. So the number of protein molecules in a cell (assuming that there is only one copy of the gene) can vary from zero to several thousand.

For the simple model discussed here, we also solve the model for the time-dependent protein concentration, $c_{\rm P}(t)$. This is important because genes can be turned on or off in response to signals, and we'd like to know how fast the cell can respond to a given signal. The time-dependent solution for protein concentration in this model can be found by simple integration,

$$c_{\rm P}(t) = \frac{k}{\mu}(1 - {\rm e}^{-\mu t}) + c_{\rm P}(0){\rm e}^{-\mu t}. \tag{1.3}$$

To work out how fast the cell can respond to a signal, let's suppose that protein P is an enzyme that allows the cell to metabolise lactose. Initially, the gene, gP, is repressed because a repressor protein is bound to its promoter. We assume that initially no protein P is present: $c_{\rm P}(0) = 0$. At time zero, the cell detects some lactose and the repressor leaves the promoter, so the gene

becomes activated. How quickly can the cell produce protein P and start metabolising lactose? If $c_{\mathrm{P}}(0) = 0$, then the dynamics is given by

$$c_{\mathrm{P}}(t) \;=\; \frac{k}{\mu}(1 - \mathrm{e}^{-\mu t}). \tag{1.4}$$

We define the rise time, t_{rise}, as the time it takes for protein P to reach half of its steady-state value. Setting $c_{\mathrm{P}}(t)$ to $c_{\mathrm{P}}^{(ss)}/2$ and solving for t_{rise}, we obtain

$$t_{rise} \;=\; -\frac{1}{\mu} \ln \left[1 - \frac{\mu c_{\mathrm{P}}^{(ss)}}{2k} \right]. \tag{1.5}$$

which becomes, when we substitute in $c_{\mathrm{P}}^{(ss)} = k/\mu$,

$$t_{rise} \;=\; \ln(2)/\mu. \tag{1.6}$$

This result tells us that the response time of this simple network is determined only by the protein-removal rate. For bacteria, protein removal is usually due to cell growth and division. As we saw earlier, the removal rate, μ, is typically $\ln(2)/\tau$, where τ is the cell cycle time. So the response time for bacterial gene networks is typically of the order of the cell cycle time, which is at least 30 min.

ODE for negatively autoregulated gene

Genes can be turned on and off by the binding of specific proteins to the DNA in the promoter region. In many cases, proteins actually turn off their own production (i.e. the protein product of a gene is a repressor that binds to its own gene and turns off protein production). This is an example of negative feedback and is called negative autoregulation. It turns out that for *E.coli*, and probably for other organisms too, negative autoregulation happens much more often than one would expect if the regulatory "connections" between genes were chosen at random. Why has negative autoregulation been selected by evolution as a favoured regulatory motif? To try to understand this, let's write down the equivalent differential equation model for a protein that represses its own production. We recall that for a protein binding to a DNA binding site, the probability that the binding site is occupied is:

$$p_{bound} \;=\; \frac{\left(\frac{c}{c_0}\right) \exp -\beta\Delta\epsilon}{1 \;+\; \left(\frac{c}{c_0}\right) \exp -\beta\Delta\epsilon}, \tag{1.7}$$

where c/c_0 is the concentration of protein (relative to some standard value, c_0) and $\Delta\epsilon$ is the change in energy when the protein binds. We can define a dissociation constant, K_d, as

$$K_d \;=\; c_0 \mathrm{e}^{\beta\Delta\epsilon}. \tag{1.8}$$

For low concentrations (where c/c_0 is very small), we can see that the probability p_{bound} that the binding site is bound becomes proportional to the inverse of the dissociation constant: $p_{bound} \to c/K_d$. This shows us that K_d is actually just the equilibrium constant for the dissociation of the protein from its binding site. The reason why this proportionality does not hold at higher concentrations is that the binding site becomes saturated with protein.

The more strongly the protein binds to its DNA binding site, the more negative $\Delta\epsilon$ will be. Strong negative autoregulation (large negative $\Delta\epsilon$ therefore corresponds to a small value of K_d.

Combining the equations above, we get

$$p_{bound} = \frac{\left(\frac{c_P}{K_d}\right)}{1 + \left(\frac{c_P}{K_d}\right)}, \tag{1.9}$$

and the probability that the binding site is unoccupied is given by

$$p_{unbound} = 1 - p_{bound} = \frac{1}{1 + \left(\frac{c_P}{K_d}\right)}. \tag{1.10}$$

Returning to our differential equation for the production and degradation of protein, the production rate is now proportional to the probability that the promoter binding site is not occupied by protein:

$$\frac{dc_P(t)}{dt} = kp_{unbound} - \mu c_P = \frac{k}{1 + \left(\frac{c_P}{K_d}\right)} - \mu c_P. \tag{1.11}$$

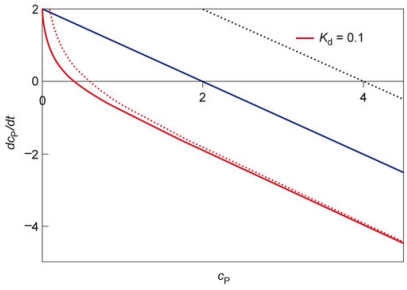

Fig. 1.3 Rate of change of protein concentration with negative autoregulation. The solid lines are for $k = 2$ and $m = 1$, and the dotted lines are for $k = 4$ and $m = 1$. The blue lines show the result without negative feedback (for the same k and m).

We now have a nonlinear differential equation for the concentration of protein, $c_P(t)$. Let's find out what the steady-state protein concentration is. Figure 1.3 shows a plot of the rate of change of c_P versus c_P, for two values of the production rate k. Also plotted are the results for a gene without negative autoregulation. We see that as in the non-regulated case, when the protein concentration c_P is low production dominates, while when the protein concentration is high protein degradation dominates over production. Again for one particular value of protein concentration production and degradation are balanced ($dc_P/dt = 0$), and this is the steady-state protein concentration.

We can see from Figure 1.3 that negative autoregulation affects the steady-state protein concentration in two important ways. First, the steady-state protein concentration is lower for the negatively autoregulated gene (shown in red) than for the unregulated gene (shown in blue). Second, when we compare the results for two different values of the production rate, k (solid and dotted lines), we can see that for the unregulated gene the steady-state protein concentration depends strongly on k (in fact, we know from our calculations above that it is proportional to k); while for

the negatively autoregulated gene, $c_{\mathrm{P}}^{(ss)}$ changes only a little when k is changed by a factor of two. Both of these effects have important implications for the performance of the gene, as we shall see.

To get an expression for the steady-state protein concentration $c_{\mathrm{P}}^{(ss)}$ for the negatively autoregulated gene, we set the rate of change of $c_{\mathrm{P}}(t)$ to zero:

$$\frac{dc_{\mathrm{P}}(t)}{dt} = \frac{k}{1 + \left(\frac{c_{\mathrm{P}}}{K_d}\right)} - \mu c_{\mathrm{P}} = 0, \tag{1.12}$$

obtaining

$$c_{\mathrm{P}}^{(ss)} = \frac{K_d}{2}\left[-1 + \sqrt{1 + \frac{k}{\mu K_d}}\right]. \tag{1.13}$$

For very strong autoregulation (where K_d is very small), the result reduces to:

$$c_{\mathrm{P}}^{(ss)} = \frac{K_d}{2}\left[-1 + 2\sqrt{\frac{k}{\mu K_d}}\right] \simeq \sqrt{\frac{kK_d}{\mu}}. \tag{1.14}$$

Figure 1.4 shows $c_{\mathrm{P}}^{(ss)}$ as a function of the protein production rate, k, for several values of the dissociation constant, K_d. As the negative autoregulation gets stronger (as K_d decreases), the curves become flatter: the steady-state protein concentration becomes less dependent on the protein production rate.

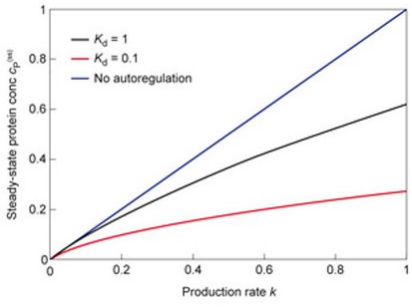

Fig. 1.4 Steady-state protein concentration for a negatively autoregulated gene.

In the cell, the protein production rate depends on the concentration of RNA polymerase, as well as the concentration of ribosomes, mRNA degradation enzymes, etc. All of these factors vary from cell to cell and over time inside any given cell. We therefore expect the protein production rate to fluctuate within and between cells. For a gene without negative autoregulation, this will cause the protein concentration to fluctuate, since $c_{\mathrm{P}}^{(ss)}$ is proportional to the production rate k. This **fluctuation problem can be avoided using negative autoregulation**. Because the curve of $c_{\mathrm{P}}^{(ss)}$ versus k is much flatter in the case of negative autoregulation, the steady-state protein concentration will remain stable even if the intracellular environment (i.e. the protein production rate) fluctuates. In other words, negative autoregulation can make the performance of a gene robust to changes in protein production rate.

You may have noticed that for negative autoregulation $c_{\mathrm{P}}^{(ss)}$ does depend on the dissociation constant, K_d. Is this a problem for robustness? Probably not: we expect K_d to fluctuate much less than k because K_d depends only on how strongly the protein binds to its DNA binding site, which is determined by the structure of the protein and the sequence of the binding site.

Negative autoregulation also has an important **effect on the rise time**, t_{rise}: the time the cell needs to turn the gene on (to the

half-maximal protein level). We saw that for the unregulated gene this time was fixed by the protein-removal rate, $t_{rise} = \ln(2)/\mu$. What happens for a negatively autoregulated gene? To calculate t_{rise}, in principle, we should solve the full version of eq. 1.12, but this is tricky analytically. If we look at early times, when $c_{\rm P}$ is small, we can approximate $c_{\rm P}(t)/K_d < 1$ then

$$t_{rise} \;=\; -\frac{1}{\mu}\ln\left[1-\frac{\mu c_{\rm P}^{(ss)}}{2k}\right], \tag{1.15}$$

and if we also assume that autoregulation is strong we can substitute the previous result for $c_{\rm P}^{(ss)}$, obtaining

$$t_{rise} \;=\; -\frac{1}{\mu}\ln\left[1-\frac{\mu}{2k}\sqrt{\frac{kK_d}{\mu}}\right] \;=\; \frac{1}{\mu}\ln\left[\frac{2}{2-\sqrt{kK_d\mu}}\right]. \tag{1.16}$$

This result is plotted in Figure 1.5: As K_d decreases (i.e. as the negative autoregulation becomes stronger), t_{rise} decreases. This important result shows that negative autoregulation can help cells to respond more rapidly to changes in their environmental conditions than they would be able to without regulation. The units chosen in Figure 1.5 are rather arbitrary. To get a feeling for some real numbers, we have already seen that a typical protein-removal rate μ in a bacterial cell would be 0.02/min, so the rise time for a typical protein without negative autoregulation would be $\ln(2)/\mu$ (~ 35 min). While protein production rates and protein-DNA dissociation constants can vary enormously, a realistic value for k might be 0.2 molecules/min per cell volume and K_d might be 0.02 molecules per cell volume (for a protein that binds very strongly to its DNA binding site). The value of t_{rise} for a negatively autoregulated gene, assuming these parameter values, would then be 12.7 min: almost a factor of three faster than the gene without negative autoregulation.

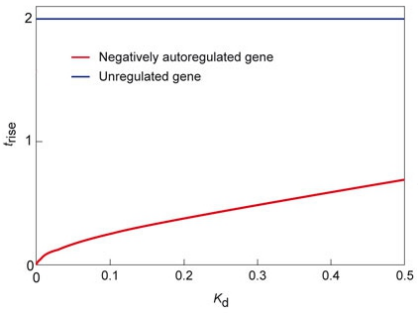

Fig. 1.5 Negatively autoregulation has strong effect on dynamics.

How are these measurements done, at population level?

Aside from noise and fluctuations, which we address below, how is the type of mRNA present in a sample (a population) of cells measured? DNA microarray chips can be used. These are large arrays (tens of thousands) of pixels (dots). Each pixel represents part of a gene, by having of the order of $10^6 - 10^9$ single-stranded DNA-mers, that are identical copies from the DNA of the gene. The chip size is of the order of 1 cm^2. The analysis consists of taking a cell sample, extracting all mRNA in this (hopefully) homogeneous sample, and translating it to cDNA (DNA that is complementary to the RNA, and thus identical to one of the strands on the original DNA). The cDNA is labeled with a fluorescent marker. The solution of many cDNAs is now flushed over the DNA chip, and the

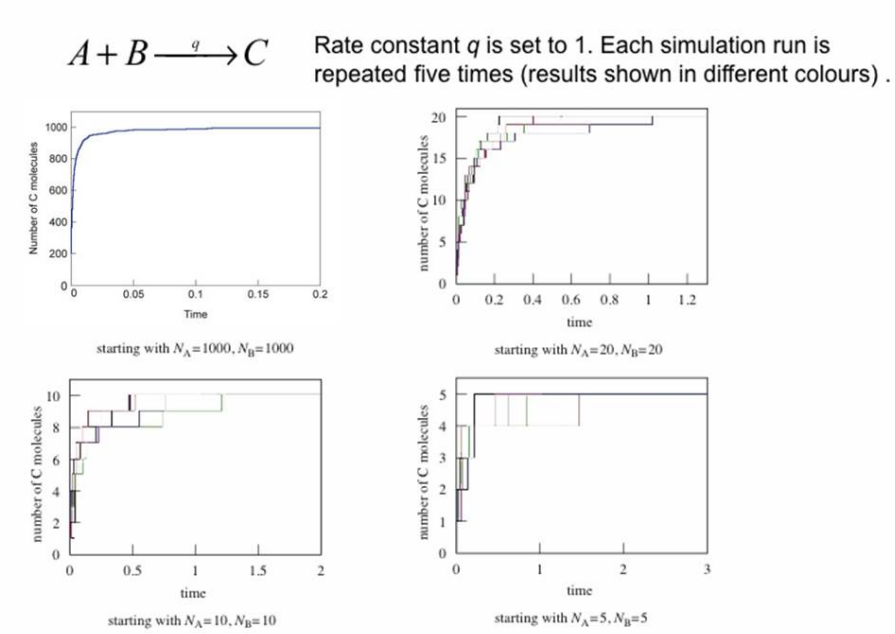

Fig. 1.6 Noise from low number of molecules can lead to different outcomes. Computer simulation results for reaction $A + B \to C$, starting with different numbers of A and B molecules.

cDNAs that are complementary to the attached single-stranded DNA-mers will bind to them. The DNA chip is washed and images (with pixel resolution) and the fluorescent light intensity thus measures the effective mRNA concentration. In its basic implementation, this technique gives one "snapshot" in time, and an average over many cells.

1.2 Biochemical noise

Cells with identical genes and environmental factors can differ chemically: we will see one way in which this can come about, using ideas about probability to model the processes mathematically.

Consider as before the reaction $A + B \to C$. Figure 1.6 shows how the number of C molecules increases in time, if we start with a 50:50 mixture of A and B. These results were obtained via computer simulations. Simulations were carried out, starting with 1000 molecules each of A and B, then with 20, 10 and 5 molecules each, with the rate constant, q, set numerically equal to 1 (to keep things simple). In each case, the simulation was repeated five times. When the total number of molecules is large, the number of C molecules rises smoothly and the repeat runs all give the same results. In this case, we can model the system with deterministic ordinary differential equations, as discussed in the

previous section.

However, if the total number of molecules is small, the system becomes very "noisy": the number of C molecules does not rise smoothly and repeat simulation runs give different results. Using standard methods from statistics, we can quantify what we mean by the number of molecules, N, being "small". It is convenient to define s as the ratio of the standard deviation in the mean to the standard deviation itself, $s = 1/\sqrt{N}$; this tends to unity for small N, and equivalently $N \simeq \sqrt{N}$. It turns out that "small molecule number" effects become important when the number of molecules becomes small enough that it is similar to its own square root.

Putting in the starting numbers of molecules for the simulations in Figure 1.6, when $N = 2000$, $s = 0.022$, but when $N = 5$, $s = 0.44$. Although Figure 1.6 shows computer simulation results, the same effect would happen in an experiment, if we could build an experimental system so small that it contained only a few molecules each of types A and B.

What is going on here? Why is our chemical reaction "noisy" when the number of molecules is small? The reason is that chemical reactions are stochastic, or random. That is, the outcome is governed by probabilities, and there are sufficiently few molecules that there is no single overwhelmingly favoured outcome. In our box of A and B molecules (the cell), we do not know the exact positions and velocities of all of the molecules and so we do not know the exact time when a pair of A and B molecules will meet and react. The exact times when reactions happen and the exact sequence of reactions that happen can be different in repeat runs of the same experiment. This may all be very interesting but why is it relevant? Even in something as small as a bacterial cell, there are many billions of molecules, so why would these stochastic effects be important? In fact, stochastic effects can be very important in cells, because even though the total number of molecules in a cell is large, the number of molecules involved in a particular biochemical reaction network can be very small. For example, in slow-growing cells, there is only one copy of the DNA (so the number of molecules of a particular gene may actually be only one). The number of messenger RNA molecules in the cell corresponding to a particular gene can also be very small for weakly expressed genes, and some proteins are only present in small numbers. Biochemical reaction networks involving genes, mRNA or proteins that are present in small numbers per cell are likely to be dramatically affected by small-molecule number fluctuations. We call these stochastic fluctuations "biochemical noise".

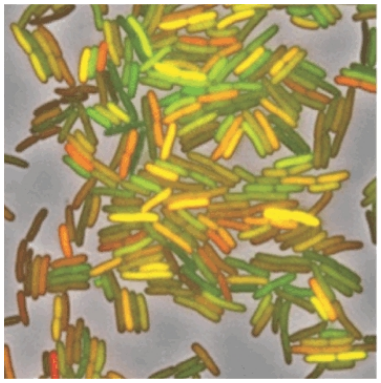

Fig. 1.7 Cells with identical genes in identical environments can behave differently. This can be explained in terms of biochemical noise. Cover image from Science Vol 297 issue 5584 (2002)

Individual cells are not identical

The fact that biochemical noise really is significant for biological cells was illustrated in an important experiment by Michael

Elowitz *et al.* in 2002. They engineered Escherichia coli bacteria carrying two different-coloured fluorescent reporter genes. These genes encode proteins that do not interfere with any cellular functions but when excited by UV light of the right wavelength they fluoresce (i.e. they emit light of a longer wavelength). This can be detected in an epifluorescence microscope. Elowitz *et al.* were therefore able to measure the relative amounts of the two fluorescent proteins in individual bacterial cells. The question that they wanted to answer was: if two cells are genetically identical and experience the same environmental conditions, will they produce the same amount of the two fluorescent proteins?

Figure 1.7 shows the results of one of their experiments. This is an overlay of micrographs of a group of *E. coli* cells growing on a semi-solid gel under the microscope. These cells all grew from a single "ancestor" at the start of the experiment so they are genetically identical. The colours show the relative amounts of the two fluorescent proteins present in each cell: green represents protein 1 and red represents protein 2. Cells that are coloured yellow contain approximately equal amounts of proteins 1 and 2. It is clear from this image that these "identical" cells are different colours, showing that they are very far from identical in their levels of production of the fluorescent proteins. Elowitz *et al.* also showed that cells that produce the reporter proteins at low levels (small number of molecules) have much more "noisy" levels of expression than cells that produce the proteins at high levels (a large number of molecules). This is what we would expect if differences between cells are caused by small molecule number noise since $s = 1/\sqrt{N}$ is larger for small N.

Concept of intrinsic and extrinsic noise. Are the differences between cells shown in Figure 1.7 really caused by small molecule number noise in the chemical reactions involved in protein production (transcription and translation)? Or are the different colours caused by differences between the cells? For example, we can see in Figure 1.7 that some cells are short because they have just been generated, while others are much longer and are about to divide. Perhaps this affects the level of protein expression? Cells could also contain different concentrations of RNA polymerase or ribosomes, which would cause them to produce more or less fluorescent protein.

To explore the origins of the different amounts of the proteins, Elowitz *et al.* used two fluorescent proteins (in different colours) instead of just one. Within each cell, the genes encoding the two proteins should experience the same cell volume, RNA polymerase, ribosome concentration, etc. So, if the differences in protein expression are caused by differences between cells, the levels of the two colours should be correlated: cells with a lot of protein 1 should also have a lot of protein 2. However, if chemical reaction stochasticity is responsible for the differences in protein

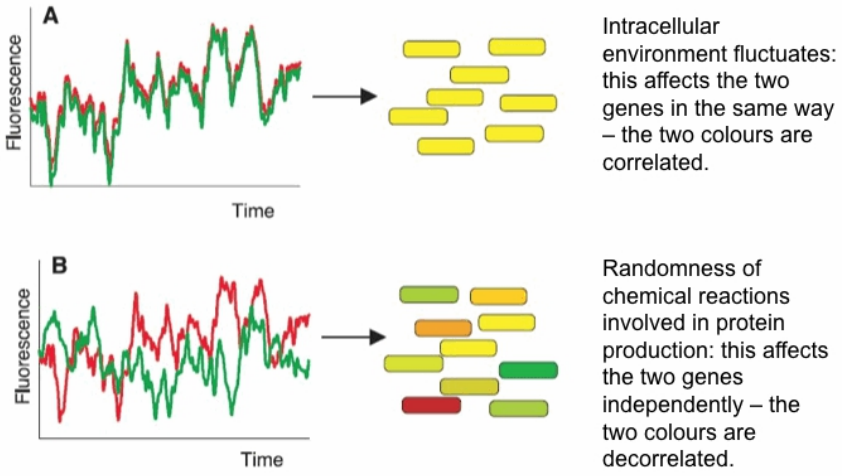

Fig. 1.8 Use of two "reporters" allows to distinguish intrinsic versus extrinsic noise. Protein levels vary because of fluctuations in the intracellular environment and of biochemical noise in transcription and translation.

expression, we would not expect the levels of protein 1 and protein 2 in individual cells to be correlated. This is illustrated in Figure 1.8.

In fact, by measuring the amount of correlation between the levels of proteins 1 and 2 in individual cells in their experiments, Elowitz *et al.* could measure how much of the cell-to-cell variation is caused by differences between cells (which they called extrinsic noise) and how much is caused by chemical reaction stochasticity (which they called intrinsic noise). In their experiments, both sources of noise played a significant role.

Why does it matter that genetically identical cells can have different levels of protein expression? One reason is that biochemical noise limits how precisely cells can control their own behaviour. If a cell needs to control precisely the concentration of a particular protein, either it must produce a large number of molecules (which is expensive) or it must use a biochemical control circuit (such as a negative feedback loop) to reduce the noise.

On a more positive note, biochemical noise may actually be useful for cells in some cases. For example, bacterial populations are often exposed to environmental stress (attack by antibiotics, changes in food availability, etc). If all of the cells in the population are identical in their protein composition, the stress may wipe them all out; but if there is large variability in protein composition among cells, it is possible that a few cells will happen to

have the right protein levels to survive the stress. The population can then regrow from these cells once the stress is over.

Theory of noise

For stochastic chemical reactions, we cannot predict exactly which reaction will happen when, or which cell in a population will contain which exact numbers of molecules of proteins, mRNA, etc. However, we can make predictions about probability distributions. For example, we might predict the probability that a randomly selected cell in a population will have 100 molecules of a particular protein, even though we cannot predict which cell this will be. The quantity we are interested in is therefore $p(N,t)$: the probability that our system contains N molecules of protein P at time t.

"Birth-death" model for gene expression. We can write down an equation for $p(N,t)$ for the simple "one-step model" of gene expression that we discussed above, in which we include chemical reactions for protein production and degradation:
source $\xrightarrow{\text{k}}$ P
P $\xrightarrow{\mu}$ sink
We assume that these reactions are "Poisson processes". This means that if we observe the system for a very short time interval from time t to time $t+dt$, the probability that the first reaction (production) happens will be

$$Prob(produce) \;=\; kdt,$$

while the probability that the second reaction (degradation) happens in this same time interval will be

$$Prob(degrade) \;=\; \mu N dt,$$

where N is the number of molecules of protein P, since the more P molecules there are, the more likely it is that this reaction will happen somewhere in the system during the time interval $t \;\rightarrow\; t+dt$.

How does the probability, $p(N)$, of having N molecules change during the time interval $t \;\rightarrow\; t+dt$? To determine this, we need to think about how the system can enter and leave the state of 'having N molecules'. To get N molecules, the system could have (a) previously had $(N-1)$ molecules and gained one more in a production reaction, or (b) previously had $(N+1)$ and lost one in a degradation reaction. These are the only ways in which the system can enter the 'state of having N molecules'. However, it can also leave this state if it already has N molecules and either (a) another one is produced (then it will have $N+1$), or (b) one is degraded (then it will have $N-1$), see Figure 1.9.

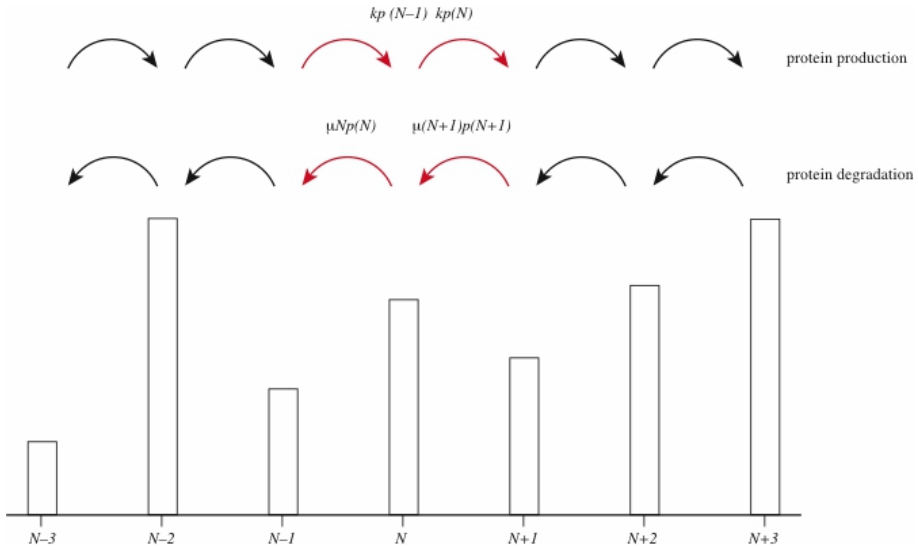

Fig. 1.9 Considering individual steps in a chemical reaction. Here, the vertical bars represent the probability of having a particular number of molecules and the arrows represent how the number of molecules is changed by the protein production and degradation reactions. In a very small time interval, $t \rightarrow t + dt$, the probability $p(N, t)$ increases due to the possibility of reactions happening from states $(N-1)$ or $(N+1)$ to N, and it decreases due to the possibility of reactions from state N to $(N-1)$ or $(N+1)$.

By summing all of the probabilities we can generate an equation called the chemical master equation:

$$\frac{dp(N,t)}{dt} = kp(N-1) + \mu(N+1)p(N+1) - kp(N) - \mu N p(N) \tag{1.17}$$

Let us suppose that we are only interested in the probability distribution $p(N)$ after a long time, once the system has reached its steady state. In that case, we have

$$\frac{dp(N,t)}{dt} = 0. \tag{1.18}$$

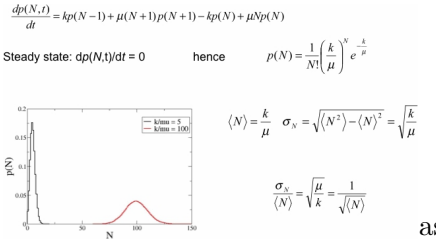

Fig. 1.10 Solution of chemical master equation for the simple one-step model of protein expression.

Solution to this is:

$$p(N) = \frac{1}{N!}\left(\frac{k}{\mu}\right)^N e^{-\frac{k}{\mu}}. \tag{1.19}$$

as you can check by substitution, noting that $p(N-1) = N(\mu/k)p(N)$ and that $p(N+1) = (k/\mu)(1/(N+1))p(N)$.
Equation 1.19 is the well known Poisson distribution.
Figure 1.10 shows the probability distribution $p(N)$ plotted for different values of (k/μ). We can see that as (k/μ) increases, the average number of molecules increases. The mean and standard

deviation σ_N of the distribution $p(N)$ are given by:

$$\begin{aligned} \langle N \rangle &= \frac{k}{\mu} \\ \sigma_N &= \sqrt{\langle N^2 \rangle - \langle N \rangle^2} = \sqrt{\frac{k}{\mu}}, \end{aligned}$$

We can estimate the importance of stochastic effects looking at the ratio of the standard deviation to the mean:

$$\frac{\sigma_N}{\langle N \rangle} = \sqrt{\frac{\mu}{k}} = \frac{1}{\sqrt{\langle N \rangle}}, \tag{1.20}$$

this explains why earlier we stated that small molecule number noise becomes important when the inverse square root of the number of molecules is close to one.

A two-step model for protein production

The model that we have just been considering may be too simple. In reality, the production of protein from a gene does not happen in a single step. We can make our model slightly more realistic by making a two-step model that includes both transcription and translation. The reaction scheme for this model would be

source $\longrightarrow$ M

M $\longrightarrow$ sink

M $\longrightarrow$ M + P

P $\xrightarrow{\mu}$ sink

Here, M represents mRNA and P represents protein. It is possible to write down a chemical master equation also for this model, and to solve it for the steady state probability distribution. In this case, there is a probability distribution for the number of messenger RNA molecules as well as for the number of protein molecules. For mRNA we only need to consider the top two reactions (since the bottom two reactions do not change the number of mRNA molecules), which are identical to our previous simpler model. So we expect the probability distribution for the number of mRNA molecules to be a Poisson distribution. However the bottom two reactions, which control the production and degradation of protein, are now different from our simple model. This means that the probability distribution of protein may be different from a Poisson distribution in this model.

Figure 1.11 shows the protein number probability distribution for this model. We set the parameters (translation rate/mRNA decay rate) so that five proteins are made on average per mRNA molecule (although some mRNA molecules will produce more and some less). We can compare this with the previous one-step model by fixing the transcription rate so that the average protein number is the same in both models. The results are shown in Figure 1.11: we can see immediately that the distribution is broader

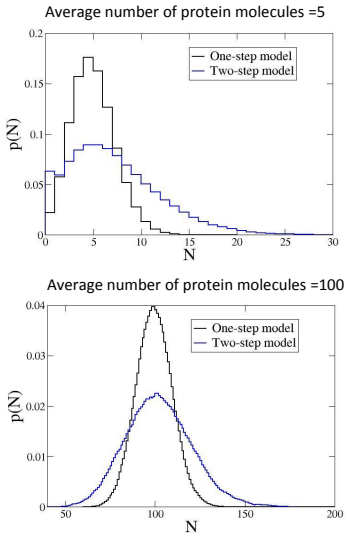

Fig. 1.11 Chemical master equation solutions for the one- and two-step models of protein expression. For the two-step process we assume that on average an mRNA produces five proteins, and we fix the transcription rate to get the same average number of proteins as in the one-step model

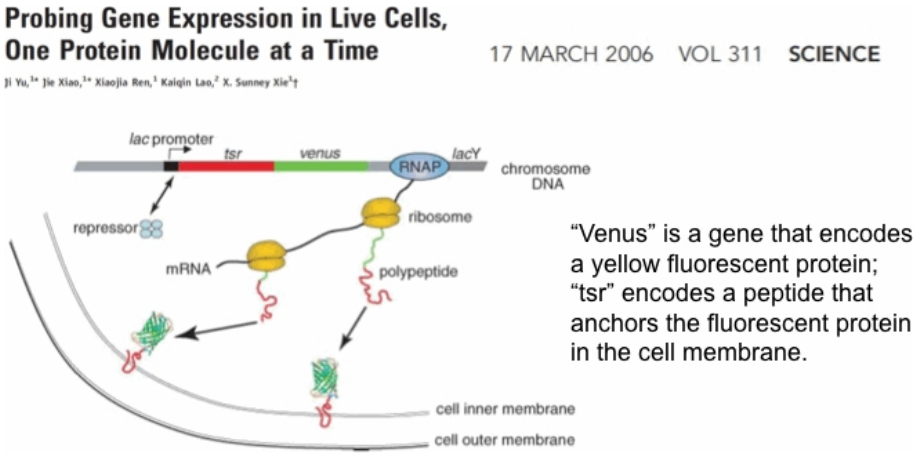

Fig. 1.12 Direct imaging of noise in gene expression. This experimental system was constructed by Yu *et al.* in 2006 to visualise in real time the production of a single protein molecule in a cell. From Yu *et al.* 2006, 'Probing gene expression in live cells, one protein molecule at a time', Science 311 1600.

in the two-step model. This model predicts more noisy protein expression than the one-step model. The reason for this is that the extra chemical reaction step amplifies the noise: the number of mRNA molecules is itself noisy, and then on top of this each mRNA molecule can produce a variable number of proteins.

Visualising noise in gene expression

How can we test whether these are good models for noisy gene expression in real cells? One way to do this is actually to carry out single molecule experiments, in other words to watch, under the microscope, the production of single protein molecules in individual cells. Since protein molecules are very small, this is a very challenging task. However, in 2006, Yu *et al.* managed to design an appropriate experiment (Figure 1.12). They made a strain of *E. coli* that produced a yellow fluorescent protein attached to a polypeptide (a chain of amino acid molecules), which could anchor this complex in the cell's lipid membrane. When the fluorescent protein is anchored in the membrane, it diffuses around much less, making it easier to see single molecules under the microscope. In this system, using advanced fluorescent microscopy, it is possible to see individual fluorescent protein molecules as dots within the cell membrane. Yu *et al.* could then grow cells under the microscope and track the moments when individual dots appeared in the membrane. In this way, they could see the production of individual protein molecules in real time. To keep the protein numbers low, the researchers included a binding site for the Lac repressor protein (see Section 1.3) When this repressor protein is bound to the operator site in front of the gene that encodes the fluorescent protein, no protein will be produced.

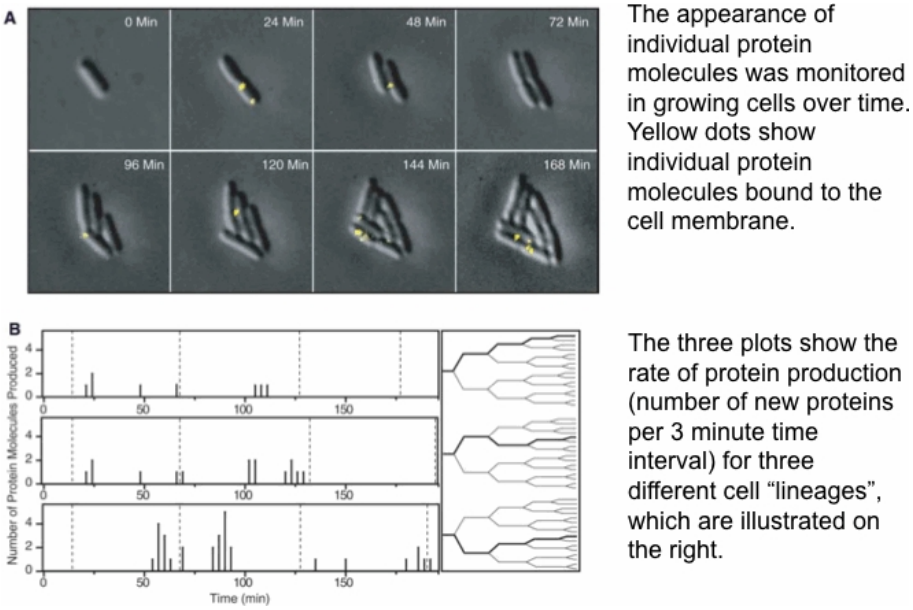

Fig. 1.13 Experiments can identify production of individual proteins. Some of Yu *et al.*'s results, showing the moment when individual protein molecules are produced in growing bacterial cells. From Yu *et al.* 2006.

Figure 1.13 shows some of Yu *et al.*'s results. The bacterial cells in the series of images grow from a single cell during the experiment. The yellow dots show individual protein molecules bound to the cell membrane. By tracking the appearance of these dots, Yu *et al.* were able to monitor the moments when protein molecules appeared in the membrane. This was done for different cell lineages, as shown in the plot, which indicates the number of protein molecules that were produced in a 3 min interval. The dotted vertical lines show the moments when the cell divided into two daughter cells.

What's really striking about Yu *et al.*'s results is that for *most of the time, no protein molecules are being produced.* Protein production occurs in short bursts, with long intervals where nothing happens. This is probably because most of the time the Lac repressor protein is bound to the DNA, thereby preventing protein expression. The bursts of expression take place during the rare moments when a stochastic fluctuation causes the repressor to fall off its DNA binding site. Yu *et al.*'s setup therefore allows us to see stochastic chemical reactions happening inside biological cells, in real time and at single-molecule resolution.

We have focused here on noise in gene expression, but the stochasticity of chemical reactions is also important in many other cell functions. Single-molecule experiments have revealed the effects of biochemical noise in the molecular machines that drive the flagellar motor that allows cells to swim and in the bacterial membrane receptors that sense environmental gradients. Other

experiments have found important effects of biochemical noise in the development of fruit-fly embryos and the mechanisms that control whether or not cells proliferate. It seems that noise is everywhere.

1.3 From a molecular to a stat mech description of regulation

We develop here a physics-based view of how gene expression is regulated, following closely the text of (Phillips et al., 2013).

RNA polymerase binding to a specific site

Following from page 242 (Phillips et al., 2013).
L ligands. Prob that 1 ligand is bound to receptor:

$$\text{weight when receptor occupied } = \mathrm{e}^{-\beta\epsilon_b} \times \sum_{solution} \mathrm{e}^{-\beta(L-1)\epsilon_{sol}},$$

where the summation is the sum over all ways of arranging the $L-1$ ligands in solution. Imagine Ω 'lattice sites' in solution. Then

$$\sum_{solution} \mathrm{e}^{-\beta(L-1)\epsilon_{sol}} = \frac{\Omega!}{(L-1)![\Omega-(L-1)]!}.$$

The partition function is

$$Z(L,\Omega) = \sum_{solution} \mathrm{e}^{-\beta L\epsilon_{sol}} + \mathrm{e}^{-\beta\epsilon_b \sum_{solution} \mathrm{e}^{-\beta(L-1)\epsilon_{sol}}}.$$

The sum in the second term has already been evaluated. The first term is

$$\sum_{solution} \mathrm{e}^{-\beta L\epsilon_{sol}} = \mathrm{e}^{-\beta L\epsilon_{sol}} \frac{\Omega!}{L!(\Omega-L)!}.$$

Bringing both together,

$$Z(L,\Omega) = \mathrm{e}^{-\beta L\epsilon_{sol}} \frac{\Omega!}{L!(\Omega-L)!} + \mathrm{e}^{-\beta\epsilon_b}\mathrm{e}^{-\beta(L-1)\epsilon_{sol}} \frac{\Omega!}{(L-1)![\Omega-(L-1)]!}.$$

If we simplify considering

$$\frac{\Omega!}{L!(\Omega-L)!} \simeq \frac{\Omega^L}{L!},$$

which is ok provided $\Omega >> L$, then we can write the probability of being bound as:

$$p_{bound} = \frac{\mathrm{e}^{-\beta\epsilon_b}\frac{\Omega^{L-1}}{(L-1)!}\mathrm{e}^{-\beta(L-1)\epsilon_{sol}}}{\frac{\Omega^L}{L!}\mathrm{e}^{-\beta L\epsilon_{sol}} + \mathrm{e}^{-\beta\epsilon_b}\frac{\Omega^{L-1}}{(L-1)!}\mathrm{e}^{-\beta(L-1)\epsilon_{sol}}}.$$

Now defining $\Delta\epsilon = \epsilon_b - \epsilon_{sol}$, we can simplify to:

$$p_{bound} = \frac{(L/\Omega)\mathrm{e}^{-\beta\Delta\epsilon}}{1 + (L/\Omega)\mathrm{e}^{-\beta\Delta\epsilon}},$$

which we can write in terms of a concentration c:

$$p_{bound} = \frac{(c/c_0)\mathrm{e}^{-\beta\Delta\epsilon}}{1 + (c/c_0)\mathrm{e}^{-\beta\Delta\epsilon}},$$

where c_0 is a reference state of full occupation. For example, if we assume our molecules to be of volume $1\,\mathrm{nm}^3$, then $c_0 = 0.6\mathrm{M}$

This, obtained here in the language of ligand/receptor binding, is a classical result known as 'Hill function', or also as a 'Langmuir adsorption isotherm' from the Part II Stat Phys course.

RNA polymerase binding: competition between specific and non specific site

This extends the calculation above. Let's assume the non-specific sites on the DNA are N_{NS} 'boxes'. Then the partition function associated with these states is:

$$Z_{NS}(P, N_{NS}) = \frac{N_{NS}!}{P!(N_{NS}-P)!} \times \mathrm{e}^{-\beta P \epsilon_{pd}^{NS}},$$

where ϵ_{pd}^{NS} is the energy of binding the polymerase to a non-specific site (and ϵ_{pd}^{S} will be later the energy of binding the polymerase to the specific site).

Now we can write the total partition function. We need to sum over the states in which the promoter is occupied (hence P-1 polymerase molecules in the non-specific sites), and those where it is not:

$$Z(P, N_{NS}) = Z_{NS}(P, N_{NS}) + Z_{NS}(P-1, N_{NS})\mathrm{e}^{-\beta\epsilon_{pd}^{S}}.$$

Hence the ratio of configuration weights where promoter is bound, to all weights, is:

$$p_{bound} = \frac{\frac{N_{NS}!}{(P-1)![N_{NS}-(P-1)]!}\mathrm{e}^{-\beta\epsilon_{pd}^{S}}\mathrm{e}^{-\beta(P-1)\epsilon_{pd}^{NS}}}{\frac{N_{NS}!}{P!(N_{NS}-P)!}\mathrm{e}^{-\beta P\epsilon_{pd}^{NS}} + \frac{N_{NS}!}{(P-1)![N_{NS}-(P-1)]!}\mathrm{e}^{-\beta\epsilon_{pd}^{S}}\mathrm{e}^{-\beta(P-1)\epsilon_{pd}^{NS}}}$$

As in the previous subsection, the factorials can be simplified, and we can write the result to show that only the energy difference matters:

$$p_{bound} = \frac{1}{1 + \frac{N_{NS}}{P}\mathrm{e}^{\beta\Delta\epsilon_{pd}}},$$

this is the familiar result for two-state models, with the unoccupied state of the promoter having weight =1, and the occupied having weight $P/N_{NS}\mathrm{e}^{-\beta\Delta\epsilon_{pd}}$.
The energy differences $\Delta\epsilon_{pd}$ are negative, and can range between minus a few to $\sim -10\,k_BT$.

Activation and repression of promoter regions

Now that the 'combinatorics' is fresh from above, we can make another construction along this line, and tackle the more complex cases of promoter regulation by transcription factors.

Activators. Activators are proteins that bind to a specific site, and promote the recruitment of RNA polymerase to a nearby promoter site. We now have 4 classes of outcome to sum over to make the total partition function: the activator and promoter site can each be occupied or unoccupied. So:

$$\begin{aligned} Z_{tot}(P, A, N_{NS}) &= Z(P, A, N_{NS}) \text{ (empty)} \\ &+ Z(P-1, A, N_{NS})\mathrm{e}^{-\beta\epsilon^S_{pd}} \text{ (only RNAP on promoter)} \\ &+ Z(P, A-1, N_{NS})\mathrm{e}^{-\beta\epsilon^S_{ad}} \text{ (only activator bound)} \\ &+ Z(P-1, A-1, N_{NS})\mathrm{e}^{-\beta(\epsilon^S_{pd}+\epsilon^S_{ad}+\epsilon_{pa})}. \text{ (both RNAP and activator bound)} \end{aligned}$$

(Here A, a are the activator, P, p the polymerase, d the DNA). ϵ_{ap} is the energy that favors the activator and the RNA polymerase being close.

The algebra is more lengthy but follows the exact steps as previously. To get promoter occupancy, we can take the ratios of the weights of the two 'favorable' states, against the sum of all weights, and we get:

$$p_{bound}(P, A, N_{NS}) = \frac{1}{1 + \frac{N_{NS}}{P F_{reg}(A)}\mathrm{e}^{\beta\Delta\epsilon_{pd}}},$$

where the function $F_{reg}(A)$ is:

$$F_{reg}(A) = \frac{1 + (A/N_{NS})\mathrm{e}^{-\beta\Delta\epsilon_{ad}}\mathrm{e}^{-\beta\epsilon_{ap}}}{1 + (A/N_{NS})\mathrm{e}^{-\beta\Delta\epsilon_{ad}}},$$

and the $\Delta\epsilon$ are the energy differences between specifically and non-specifically bound conditions.

This is a neat result, because it shows that activating molecules make an $F > 1$, i.e. have an effect that is mathematically equivalent to increasing the number of polymerases. Given realistic values of the other energies, a few $-k_BT$ for ϵ_{ap} is enough to significantly change the bound probability, see (and reproduce your own?) Figs.19.10 and 19.11 in (Phillips et al., 2013).

If the approx $(N_{NS}/PF_{reg})\mathrm{e}^{\beta\Delta\epsilon_{pd}} >> 1$ holds, i.e. the promoter is not too strong, then you can obtain (exercise) that the fold increase is approximately $F_{reg}(A)$ itself.

Repressors. Repressor proteins occupy the promoter region, and prevent the PRNA binding there. The statistical mechanics

approach is a variant of the above. The partition function associated with binding of repressors to the non-specific sites is:

$$Z(P,R,N_{NS}) \;=\; \frac{N_{NS}!}{P!R!(N_{NS}-P-R)!}\mathrm{e}^{\beta P\epsilon_{pd}^{NS}}\,\mathrm{e}^{\beta R\epsilon_{rd}^{NS}}.$$

Now the total partition function is:

$$\begin{aligned} Z_{tot}(P,R,N_{NS}) \;&=\; Z(P,R,N_{NS}) \;\text{(empty promoter)}\\ &+ Z(P-1,R,N_{NS})\mathrm{e}^{-\beta\epsilon_{pd}^{S}} \;\text{(RNAP on promoter)}\\ &+ Z(P,R-1,N_{NS})\mathrm{e}^{-\beta\epsilon_{rd}^{S}}. \;\text{(repressor on promoter)}\end{aligned}$$

With the same algebra steps and approximations as previously, we obtain

$$p_{bound}(P,R,N_{NS}) \;=\; \frac{1}{1\;+\;\frac{N_{NS}}{P}\mathrm{e}^{\beta(\epsilon_{pd}^{S}-\epsilon_{pd}^{NS})}[1+\frac{R}{N_{NS}}\mathrm{e}^{-\beta(\epsilon_{rd}^{S}-\epsilon_{rd}^{NS})}]}.$$

To obtain a compact expression of the same form as for activators, a regulating function $F_{reg}(A)$ can be defined as:

$$F_{reg}(R) \;=\; \left(1\;+\;\frac{R}{N_{NS}}\mathrm{e}^{-\beta\Delta\epsilon_{rd}}\right)^{-1},$$

with $\Delta\epsilon_{rd} = \epsilon_{rd}^{S} - \epsilon_{rd}^{NS}$. Here, $F_{reg} < 1$, which means that the systems behaves as if fewer polymerases were present.

Towards the real case: activation and repression! In a real regulatory system, both mechanisms can interplay. Again we can build on the same lines as before, and there are now six distinct possible outcomes:

$$\begin{aligned} Z_{tot}(P,A,R,N_{NS}) \;&=\; Z(P,A,R,N_{NS}) \;\text{(empty promoter)}\\ +\quad & Z(P-1,A,R,N_{NS})\mathrm{e}^{-\beta\epsilon_{pd}^{S}} \;\text{(RNAP on promoter)}\\ +\quad & Z(P,A-1,R,N_{NS})\mathrm{e}^{-\beta\epsilon_{ad}^{S}} \;\text{(activator on promoter)}\\ +\quad & Z(P-1,A-1,R,N_{NS})\mathrm{e}^{-\beta(\epsilon_{ad}^{S}+\epsilon_{pd}^{S}+\epsilon_{pa})} \;\text{(RNAP and activator on)}\\ +\quad & Z(P,A,R-1,N_{NS})\mathrm{e}^{-\beta\epsilon_{rd}^{S}} \;\text{(repressor on promoter)}\\ +\quad & Z(P,A-1,R-1,N_{NS})\mathrm{e}^{-\beta(\epsilon_{ad}^{S}+\epsilon_{rd}^{S})}. \;\text{(activator and repressor on)}\end{aligned}$$

As before the RNA polymerase binding probability can be calculated and has the form:

$$p_{bound}(P,A,R,N_{NS}) \;=\; \frac{1}{1\;+\;\frac{N_{NS}}{P\,F_{reg}(A,R)}\mathrm{e}^{\beta(\epsilon_{pd}^{S}-\epsilon_{pd}^{NS})}},$$

where the regulating function $F_{reg}(A,R)$ is richer:

$$\begin{aligned} F_{reg}(A,R) \quad = \quad & \Big[1+(A/N_{NS})\mathrm{e}^{-\beta(\Delta\epsilon_{ad}+\epsilon_{ap})}\Big] \;/\\ & \Big[1\;+\;(A/N_{NS})\mathrm{e}^{-\beta\Delta\epsilon_{ad}}+(R/N_{NS})\mathrm{e}^{-\beta\Delta\epsilon_{rd}}+\\ & (A/N_{NS})(R/N_{NS})\mathrm{e}^{-\beta(\Delta\epsilon_{ad}+\Delta\epsilon_{pd})}\Big].\end{aligned}$$

The *lac* Operon

The *lac* Operon has played a key role historically in understanding physical and biological aspects of gene regulation. In the *lac* Operon there is an activator, the protein CAP: in order to recruit RNAP, CAP has to be bound to a molecule called cyclic AMP (cAMP), whose concentration goes up when amount of glucose decreases. There is also a repressor, the Lac repressor, which decreases the amount of transcription unless it is abound to allolactose, a byproduct of lactose metabolism.
Keep in mind that this regulation is just to ensure that the enzymes to digest lactose are produced only when glucose is not present, and lactose is present. It seems an apparent simple objective, but selecting reliably for one of four situations requires a mechanism of both activation and repression as outlined here.

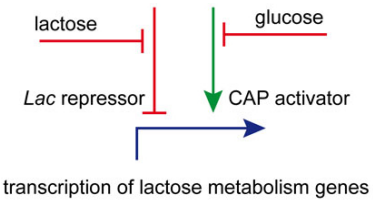

Fig. 1.14 Idealised (logic) lac network. A convenient way to illustrate the molecular interactions that make up the lac regulatory network. Here, positive molecular interactions (activation) are shown by arrows and negative molecular interactions (repression) are shown by "blocker" bars. The input to the network are the concentrations of lactose and glucose, the output is the activation of gene transcription for the machinery required to metabolise lactose. This type of diagram is often used to represent regulatory networks and is convenient when the networks are complicated, involving a lot of interactions.

Our thermodynamical model is in fact still too simple to describe quantitatively the *lac* Operon. There is another important detail which is worth mentioning, because it brings in the nature of the DNA double-helix as a polymer, with all the 'polymer physics' concepts that have been studied in other contexts. What we have not considered in the models above is the fact that (a) each lac repressor molecule has two binding sites, combined with (b) that there are three operator regions on the DNA for lac to bind (with slightly different binding energies, and situated 92 and 401bp on each side of the main operator). This fact corresponds to the possibility for the lac repressor to form a loop of DNA. Bending of double-stranded DNA carries of course a free energy cost, and this cost can in principle be regulated by the cell through associations of proteins, or physical chemistry changes, that lead to changes in DNA persistence length.

On one hand this *lac* Operon, is a classic system of study, and considered understood well enough to be used in an 'engineering' building-block spirit in synthetic biology constructions. On the other hand, it is still the study of refined experiments and models, aiming to understand it fully quantitatively. That systems open up new refined questions as we understand more of them is a familiar theme in various areas of physics.

Regulating gene expression by DNA conformation, with loops or compact regions stabilised by protein adhesion, is a very general mechanism heavily exploited in eukaryotic cells.

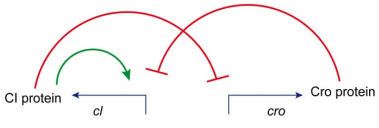

Fig. 1.15 Idealised (logic) and simplified diagram of the phage lambda genetic switch. The *cro* gene results in cell lysis; the *cI* gene promotes lysogeny.

Case study: lambda phage

This is another very well studied "hydrogen atom" situation in biology. A virus called bacteriophage lambda infects *E. coli*. Once a bacterial cell is infected, the virus has two options: it can either hijack the cell machinery to replicate itself and then kill the cell (known as lysis), resulting in its release, or it can add its DNA

to the DNA sequence of the bacterium and lie dormant inside the host cell (known as lysogeny) until conditions are more favourable for lysis. Which of these developmental pathways is adopted is determined by (a more complex version of) the regulatory network shown in Figure 1.15. This network contains two genes, *cI* and *cro*. When the *cro* gene is activated, cell lysis results; when the *cI* gene is activated, lysogeny follows. What prevents both pathways from being activated simultaneously?

As shown in Figure 1.15, the *cI* gene encodes a protein, CI, which acts as a repressor of the *cro* gene and an activator of its own gene. Thus, when *cI* is active, *cro* is repressed and remains inactive, while *cI* remains active. Likewise, the *cro* gene encodes a protein, Cro, which acts as a repressor of the *cI* gene. Thus, when the *cro* pathway to lysis has been adopted, the *cI* pathway to lysogeny is automatically shut down. In this way, the virus ensures that a binary all-or-nothing "decision" is made between lysis and lysogeny. This is an example of a bistable switch: a regulatory network with two distinct outcomes. Bistable switches are important not just for bacteriophage lambda but also in developmental and cell-fate decisions in many other cells, including human ones. (Bistable switches are also used in electronic control networks, where they maintain a circuit in one of two stable states until some external trigger is applied - very similar to their biological analogue.)

1.4 Simulating chemical reaction dynamics

Gene expression in a living organism is not a steady state process: at the embryo development level, regulation evolves within a cell cycle, and very significantly at cell division; the cell cycle also defines genes that are only expressed at certain times; 'zooming in' at even shorter times, the gene expression can often be seen to be happening in bursts. A dynamical description of concentrations can be important. Careful experiments, and models, can highlight the various sources of 'noise' (stochasticity) in expression, which can be quite different in origin: for example from the molecular binding event, to fluctuations in concentrations, to noise that comes from the coupling of the dynamical process.

Other processes in the cell for example the translation of proteins, or reaction networks of proteins, also can exhibit transients in time, and noise. How can we model this? Except in the simplest cases, there is not much that can be one analytically. Given a set of coupled differential equations, one can solve numerically. In a brute-force approach, a constant timestep for integration could be chosen: this would have to be much smaller than any reaction or decay timescales, and can be very wasteful of simulation time. A very elegant way to address these problems computationally was

proposed by Gillespie in 1977, and his algorithm is still in current use.

the Gillespie algorithm

The Gillespie algorithm instead of working with a constant Δt provides a strategy for adapting the timestep to the problem, by choosing it at random from a particular probability distribution. A second (biased) random number then determines which of the reactions take place at the simulation step. Running this algorithm is equivalent to following one particular realisation of the stochastic dynamics of a system. It is powerful because it has 'real time', and because by running it with several iterations one can build up distributions.

Let's see how the algorithm works (what is the correct probability distribution for Δt, and how to choose the reaction) with the example of the unregulated promoter. There are two reactions:
(1) an mRNA can be can be produced, with probability k per unit time;
(2) an mRNA can decay, with probability γ per unit time and per unit molecule.
Let's call $m(t)$ the number of mRNA molecules at time t.
Once we have a timestep Δt , we want to determine $P(i, \Delta t)dt$, the probability that reaction i takes place in the interval $\Delta t, \Delta t + dt$. First, we note that we also want to impose no reaction to have occurred before Δt. We call this probability $P_0(\Delta t)$. Thus the probability that reaction i takes place in the interval $\Delta t, \Delta t + dt$ is

$$P(i, \Delta t)dt \; = \; P_0(\Delta t)k_i dt.$$

How do we calculate $P_0(\Delta t)$? We can write

$$P_0(\Delta t + dt) \; = \; P_0(\Delta t)\left(1 - \sum_i k_i dt\right),$$

i.e. the product of the probability of no reaction having occurred up to Δt, time the probability of no reaction taking place in dt. The first term can be Taylor expanded around Δt, and we obtain

$$\frac{dP_0(\Delta t)}{d\Delta t} \; = \; -P_0(\Delta t)\sum_i k_i,$$

which has solution

$$P_0(\Delta t) \; = \; \mathrm{e}^{-\sum_i k_i \Delta t} \; = \; \mathrm{e}^{-k_0 \Delta t},$$

where we have used $P_0(\Delta t = 0) = 1$ and defined $k_0 = \sum_i k_i$.
Substituting back, we get

$$P(i, \Delta t)dt \; = \; \mathrm{e}^{-k_0 \Delta t} k_i dt.$$

If we sum this over all i, we get the probability that any of the possible reactions happens in the interval $\Delta t, \Delta t + dt$:

$$P(\Delta t)dt \; = \; \mathrm{e}^{-k_0 \Delta t} k_0 dt.$$

This is the distribution from which one needs to pick Δt.

Now we need to work out how to make a distribution from which to pick the random choice of which reaction takes place. The probability that reaction i happens at *some* time is:

$$P(i) \; = \; \int_0^\infty P(i, \Delta t) d(\Delta t) \; = \; \frac{k_i}{k_0}.$$

This tells us that the probability of a reaction to take place is just the ratio of its rate, and the sum of all the possible rates. This gives us the criterion to choose (randomly, but with the right bias) which reaction will take place at the simulation timestep.

In algorithm form, the steps in this example are:
1. given $m(t)$, calculate the rates. In this case only k_2 depends on $m(t)$.
2. draw a uniform random number x between $[0, 1]$. Compute k_0. $\Delta t = (1/k_0) \ln(1/x)$. This last formula is a way (you can check) to turn the uniform random number in a random number from the exponential distribution we want, calculated above. Advance simulation clock by Δt.
3. draw a uniform random number between $[0, 1]$. If the number is between $[0, k_1/k_0[$, increase the mRNA molecule number by one. If it is between $[k_1/k_0, 1]$ then decrease the mRNA molecule number by one.
4. loop back to step (1).

Check that the distribution of m at steady state is well described by a Poisson distribution. This is a result that could have been obtained analytically, in this simple example.

Dynamical Systems: Systems and Circuits

2

Recall the introduction to one and two-dimensional dynamical systems, from previously in the course. Here we investigate, following section 19.3.5 of (Phillips et al., 2013), some interesting examples of processes at the cell biology level that can be modelled as dynamical systems.

2.1 Gene regulation switches

Let's consider a synthetic switch that was created in *E.coli* as a simple construction to understand possibly more complex biological switches. The construction consists of two repressor proteins, whose transcription is mutually regulated. This arrangement gives rise to feedback, and we will see that it allows for a very non-trivial switch between steady states, depending on the initial conditions of the system.

The concentrations of the two proteins are c_1 and c_2, and we want to write equations for the time derivatives of concentration. Each protein is subject to two processes:
(1) degradation at a rate γ, and
(2) its expression, but regulated via the concentration of the other protein. Let's assume that there is a basal (un-repressed) rate r, and that the actual rate of expression is $r(1 - p_{bound})$. If we take the rate of binding to be a Hill function of some order n,

$$p_{bound}(c_1) = \frac{K_b c_1^n}{1 + K_b c_1^n},$$

with K_b the binding constant for the repressor. The expression of protein 2 will then be given by:

$$r(1 - p_{bound}) = \frac{r}{1 + K_b c_1^n}$$

This gives us the coupled equations:

$$\begin{aligned} \frac{dc_1}{dt} &= -\gamma c_1 + \frac{r}{1 + K_b c_2^n} \\ \frac{dc_2}{dt} &= -\gamma c_2 + \frac{r}{1 + K_b c_1^n}. \end{aligned}$$

These can be made dimension-less by expressing concentrations in units of $K_b^{-1/n}$, and time in units of γ^{-1}. Then the equations are:

$$\begin{aligned} \frac{du}{dt} &= -u + \frac{\alpha}{1+v^n} \\ \frac{dv}{dt} &= -v + \frac{\alpha}{1+u^n}, \end{aligned}$$

where $\alpha = rK_b^{1/n}/\gamma$.

We can see that there is always one steady state solution:

$$u^* = v^* = \frac{\alpha}{1+v^{*n}}.$$

Let's see if there are other steady state solutions. Let's consider $n = 2$ to proceed with calculus. The steady state values have to satisfy

$$u^* = \frac{\alpha}{1+\left(\frac{\alpha}{1+u^{*2}}\right)^2},$$

and the corresponding equation for v^*. This can be expanded as:

$$(u^{*2} - \alpha u^* + 1)(u^{*3} + u^* - \alpha) \;=\; 0.$$

The cubic polynomial here can be shown to have only one zero, and by some inspection you can see that it is the solution with $u^* = v^*$. The quadratic however can have 0 (if $\alpha < 2$), 1 (if $\alpha = 2$), or 2 (if $\alpha > 2$) solutions, depending on the value of α. In the 2-solution regime, the concentrations are not the same! The solution with $u^* = v^*$ exists for all α, but it is unstable for $\alpha > 2$.

Calculate phase portraits of this system.

2.2 Oscillations in gene expression

Another ubiquitous dynamical element are coupled equations capable of sustaining oscillations. It has even been proposed that, much like FM vs. AM radio, oscillatory dynamics is used by some cell processes to code and transmit information robustly. One simple set of equations that gives rise to oscillations is a gene regulated by both an activator and a repressor:
- the repressor binds as a dimer, and represses production of the activator
- the activator also binds as a dimer, and increases the production of itself, and also of the repressor.

Then the rate equations can be written as:

$$\begin{aligned}\frac{dc_A}{dt} &= -\gamma_A c_A + r_{0A}\frac{1}{1+(C_A/K_d)^2+(C_R/K_D)^2} + \\ &\quad + r_A\frac{(c_A/K_d)^2}{1+(C_A/K_d)^2+(C_R/K_D)^2} \\ \frac{dc_R}{dt} &= -\gamma_R c_R + r_{0R}\frac{1}{1+(C_A/K_d)^2} + r_R\frac{(c_A/K_d)^2}{1+(C_A/K_d)^2},\end{aligned}$$

where r_{0A}, r_{0R} are the basal expression rates, and r_A, r_R are the regulated rates in the presence of the activator bound.
As before, it is possible to write the equations in dimension-less form:

$$\begin{aligned}\frac{d\tilde{c}_A}{dt} &= -\tilde{\gamma}_A\tilde{c}_A + \frac{\tilde{r}_{0A}+\tilde{r}_A\tilde{c}_A^2}{1+\tilde{c}_A^2+\tilde{c}_R^2} \\ \frac{d\tilde{c}_R}{dt} &= -\tilde{c}_R + \frac{\tilde{r}_{0R}+\tilde{r}_R\tilde{c}_A^2}{1+\tilde{c}_A^2}.\end{aligned}$$

Oscillations can arise if there is a separation of timescales between the activator and repressor dynamics. 'Nullclines' are the locus of points achieved by the repressor or activator at steady state, given fixed values of activator or repressor, respectively. They are obtained by setting the time derivatives equal to zero, and we have:

$$\begin{aligned}\tilde{c}_R &= \sqrt{-1-\tilde{c}_A^2+\frac{\tilde{r}_{0A}+\tilde{r}_A\tilde{c}_A^2}{\tilde{\gamma}_A\tilde{c}_A}} \\ \tilde{c}_R &= \frac{\tilde{r}_{0R}+\tilde{r}_{AR}\tilde{c}_A^2}{1+\tilde{c}_A^2}.\end{aligned}$$

See fig.19.51 (Phillips et al., 2013).

References

Phillips, R., Kondev, J., Theriot, J., Garcia, H., and Orme, N. (2013). *Physical Biology of the Cell, 2nd Ed.* Garland Science, London.

Strogatz, S. S. (2014). *Nonlinear Dynamics and Chaos*. Westview Press, Boulder.