Tutorial: Genetic circuits and noise

Matt Scott*
Quantitative Approaches to Gene Regulatory Systems
Summer School, July 2006
University of California, San Diego

The analysis of natural genetic networks and the construction of new synthetic analogues is complicated by fluctuations associated with discrete reaction events in small-number reactant pools. While deterministic models are often invoked with success, there are increasingly many examples where deterministic descriptions fail to capture essential features of the underlying stochastic system. In this tutorial, we shall discuss how genetic circuit models are formulated mathematically from biological principles, leading to deterministic systems of differential equations. We shall then examine the role of *molecular noise* in circuit function and ask how the full stochastic problem may be formulated. The resulting models can rarely be solved exactly, so we conclude with a look certain useful approximation schemes.

I. GENETIC CIRCUITS

A. Modeling gene expression

The process of gene expression and protein synthesis is shown as a biological schematic in Figure 1a, and as a further simplified representation in Figure 1b. To codify the process as a mathematical model, we must assign quantitative reaction rates to each event (Table I).

How do we turn a biological idea into a mathematical model? We appeal to conservation laws - e.g. Mass-balance: Change = flux in - flux out. This change is most conveniently represented as a *differential equation* for the species concentrations. The rate of change of mRNA (m) and of protein (p) concentration is written as,

$$\frac{dm}{dt} = \alpha_m - \beta_m\, m \qquad \frac{dp}{dt} = \alpha_p\, m - \beta_p\, p. \tag{1}$$

We are often concerned with the *steady-state* solution of the governing equations, *i.e.* we set $\frac{dm}{dt} = 0 = \frac{dp}{dt}$ and solve for $m^\star$ and $p^\star$, the levels of mRNA and protein attained when the system is no longer changing. At steady-state, the differential equations above reduce to a system of algebraic equations,

$$0 = \alpha_m - \beta_m\, m \qquad 0 = \alpha_p\, m - \beta_p\, p. \tag{2}$$

The solution is readily calculated, $m^\star = \frac{\alpha_m}{\beta_m}$ and $p^\star = \frac{\alpha_p}{\beta_p} m^\star = \frac{\alpha_m\, \alpha_p}{\beta_m\, \beta_p}$. For the ball-park estimates of the various reaction rates given in the Table I, we have $m^\star \approx 1 - 10nM$ and $p^\star \approx 100 - 1000nM$. In *E. coli*, the cell volume is

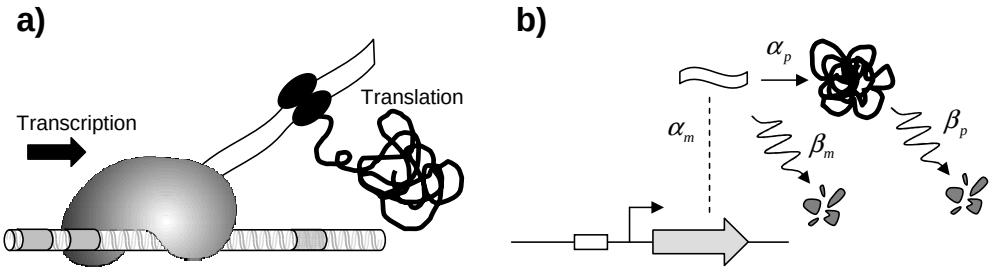


FIG. 1: a) A simplified biological model of gene expression. b) A further simplified mathematical formulation of the same.

*Electronic address: mscott@ctbp.ucsd.edu

Event	Symbol	Description	Ball-park Magnitude Estimates
Transcription	α_m	Rate of mRNA synthesis	~ few nM/min < 60 nM/min
mRNA Degradation	β^{-1}_m	Average mRNA lifetime	~ few min < 20 min
Translation	α_p	Translation rate	~ few mRNA/min
Protein Degradation	β^{-1}_p	Average protein lifetime	~ doubling time (dilution) (* Faster with active proteolysis*)

TABLE I: Estimates of reaction rates for various aspects of gene expression in *E. coli*.

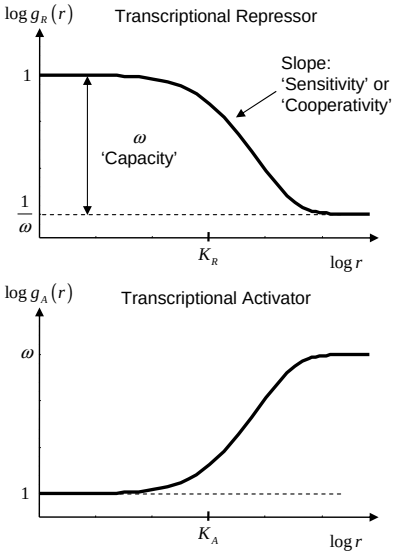


FIG. 2: (Top) Cartoon schematic of the promoter activity function $g_R(r)$ for *repressing* action. The transcription factor binding dissociation constant is called K_R.

(Bottom) Cartoon schematic of the promoter activity function $g_A(r)$ for *activating* action. The transcription factor binding dissociation constant is called K_A. In both cases, the fold-change due to regulation is denoted by the capacity ω.

See Bintu L, Buchler NE, Garcia HG, Gerland U, Hwa T, Kondev J, Phillips R (2005). Transcriptional regulation by the numbers: models. Current Opinions in Genetics & Development 15: 116, and

Buchler NE, Gerland U, Hwa T (2005). Effects of nonlinear protein degradation on the functions of genetic circuits. PNAS 102: 9559.

about 10^{-15} L, and consequently $1nM$ corresponds to about 1 molecule per cell. The steady-state levels of mRNA and protein are then typically $\ll 10^4$ molecules per cell.

B. Genetic circuits - adding regulation to expression

Under certain assumptions (most important to the present discussion, we assume *fast* transcription factor / DNA binding), the rate of RNAp binding is represented by the promoter activity functions $g_R(\,\cdot\,)$ and $g_A(\,\cdot\,)$ (Figure 2). A simple way to include transcription factor control in the kinetic equations above is to modify α_m using the *static* promoter activity functions $g_R(\,\cdot\,)$ or $g_A(\,\cdot\,)$. This idea is best understood by example.

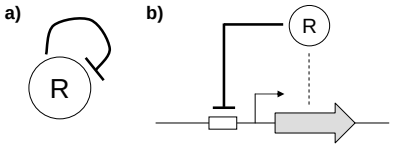


FIG. 3: a) A functional schematic of an auto-repressing circuit. b) A more detailed description of the same.

Note: A connector that ends with a blunt line ⊣ indicates a *repressing* action. In contrast, a connector ending in an arrow → indicates an *activating* action.

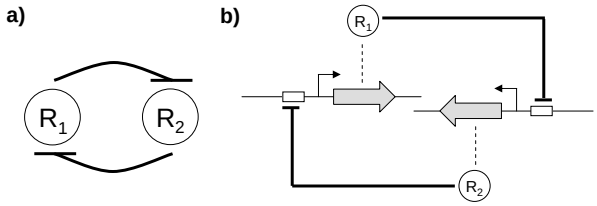


FIG. 4: a) A functional schematic of the toggle switch. b) A more detailed description of the same.

See T. S. Gardner, C. R. Cantor and J. J. Collins (2000) Construction of a genetic toggle switch in *E. coli*. Nature 403: 339.

a) Autorepressor: gene product represses its own transcription, as shown schematically in Figure 3.

The basic model of gene expression is modified in this case to read, (where p has been replaced by r to emphasize that the protein product is behaving as a repressor).

$$\frac{dm}{dt} = \alpha_m \cdot g_R(r) - \beta_m\, m \qquad \frac{dr}{dt} = \alpha_p\, m - \beta_p\, r. \tag{3}$$

Exercise: To derive an explicit expression for $g_R(r)$, we assume fast transcription factor / DNA binding. Write the rate equations for the autorepressor, explicitly including transcription factor binding to the promoter. (Assume there is one promoter with two operator binding sites and the repressor binds cooperatively as a monomer). Under what conditions does this new set of equations reduce to the set above? What is the explicit form of $g_R(r)$ in this case? The fast transcription factor / DNA binding is "fast" compared to what timescale?

b) Toggle switch: Mutually repressing network, shown schematically in Figure 4.

We would expect the system to exhibit *two* mutually exclusive behaviors - either r_1 is high, keeping expression of r_2 low, or conversely, r_2 is high, keeping expression of r_1 low. For simplicity, we assume that the switch is composed of symmetric elements so that the rate constants are identical for each half of the network. Then, as in the previous example, the mathematical model takes the form,

$$\begin{aligned} \frac{dm_1}{dt} &= \alpha_m \cdot g_R(r_2) - \beta_m\, m \quad \frac{dm_2}{dt} = \alpha_m \cdot g_R(r_1) - \beta_m\, m \\ \frac{dr_1}{dt} &= \alpha_p\, m_1 - \beta_p\, r_1 \quad \frac{dr_2}{dt} = \alpha_p\, m_2 - \beta_p\, r_2. \end{aligned} \tag{4}$$

What can we do with this? It is a system of coupled nonlinear differential equations, which are very difficult (usually impossible) to solve exactly. We have two choices:

1. Solve the system numerically using Matlab, Mathematica, *etc.*
2. Find some kind of approximate solution. We shall discuss a common method of approximation.

In *E. coli*, the mRNA typically degrades *much* faster than the protein, ($\beta_m \gg \beta_p$). Consequently, after a short transient has elapsed, $\frac{dm}{dt} \approx 0$ over the timescale of protein change. Solving the resulting algebraic equations for m_i, we arrive an mRNA concentration that is *slaved* to the slower-changing protein concentration,

$$m_i(r_j) = \frac{\alpha_m}{\beta_m} \cdot g_R(r_j) \quad (i, j \in \{1, 2\}) \tag{5}$$

We have then effectively reduced the number of differential equations from four to two, contingent upon the assumption that $\beta_m \gg \beta_p$. Calling $\gamma^0 = \frac{\alpha_p\, \alpha_m}{\beta_m}$ the maximum rate of repressor synthesis, the governing equations are written,

$$\frac{dr_i}{dt} = \gamma^0 \cdot g_R(r_j) - \beta_p\, r_i. \tag{6}$$

Calling $\frac{\gamma^0}{\beta_p} \equiv r_i^0$ the maximum level of repressor i, we can likewise solve for the steady-state concentration of the repressors $r_i^\star$,

$$r_i^\star(r_j^\star) = r_i^0 \cdot g_R(r_j^\star). \tag{7}$$

So far, so good - but the $r_i^\star$'s are still unknown, as are the conditions that ensure bistability in the system. We can gain a great deal of insight into the model by considering the cartoon of the promoter activity function $g_R(r_i)$

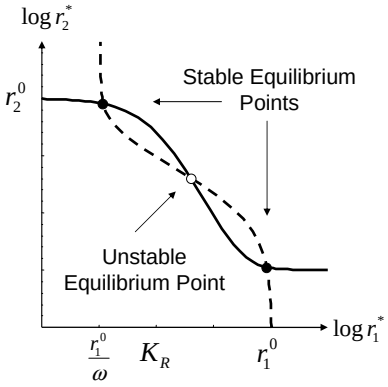


FIG. 5: We can plot the steady-state level of r_1 as a function of r_2 and vice-versa. Their points of intersection represent the *equilibrium points* of the system. Local analysis of the linearized dynamics near the equilibria will tell us if the points are stable or not.

A good reference for analysis of nonlinear systems of differential equations is Strogatz SH (2000) Nonlinear dynamics and chaos. Westview-Perseus, Cambridge, MA.

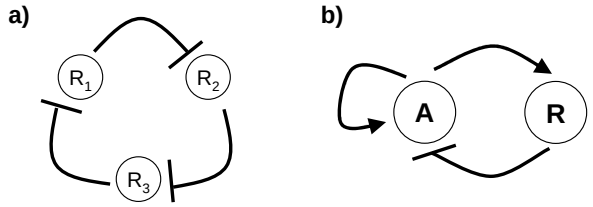


FIG. 6: a) The 'Repressilator'.
See Elowitz MB, Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. Nature 403: 335.

b) The 'Predator-prey' oscillator.
See Atkinson MR, Savageau MA, Myers JT, Ninfa AJ (2003) Development of genetic circuitry exhibiting toggle switch or oscillatory behavior in *E. coli*. Cell 113: 597.

(Figure 2) to plot how the steady-state concentrations of r_1 and r_2 depend upon one another. The curves determined by Eq. 7 are called the *nullclines* of Eq. 6. The intersection of the nullclines is called an *equilibrium point* since at this point $\frac{dr_1}{dt} = \frac{dr_2}{dt} = 0$. Exploring various choices of repressor binding affinity K_R and unrepressed protein level r_i^0 we can realize systems with one equilibrium point (called *monostable*), or three equilibrium points, two of which are stable (called *bistable*, see Figure 5). From the plot of the nullclines, we can see that the approximate conditions for bistability are that $\frac{r_i^0}{\omega} < K_R < r_i^0$.

Exercise: We have been operating under the assumption that the circuit parameters are symmetric, *i.e.* $r_1^0 = r_2^0$. In general, of course, this will not be true. Repeat the above analysis for asymmetric rate constants ($\alpha_{m_i}, \beta_{m_i}, K_{R_i}, \omega_i$, *etc*). How do the conditions for bistability change? (Try to guess the answer before you work it out in detail).

Exercise: Write the differential equations governing the evolution of the mRNA and protein products that correspond to the circuits illustrated in Figure 6. (Assume fast transcription factor / DNA binding.).

There are three major simplification underlying our approach to modeling regulated gene expression:

1. The concentrations of the reactants evolve continuously and differentiably so that description by differential equations is justified.
2. Reactions occur instantaneously and depend only upon the present state of the system.
3. Reactants are distributed *homogeneously* in space.

To adapt the mathematical formalism to accommodate each of these simplifications requires the following,

1. Consider each reaction event as essentially probabilistic, which changes reactant molecule numbers by a (finite) discrete amount.
2. Use delayed reaction rates - rates that depend upon the past state of the system.
3. Abandon *ordinary* differential equations in favor of *partial* differential equations that explicitly include spatial density of reactants.

Depending upon the context, various combinations of the preceding may be appropriate. In the next section, we shall address the first problem, reformulating our models of genetic circuits as probabilistic processes.

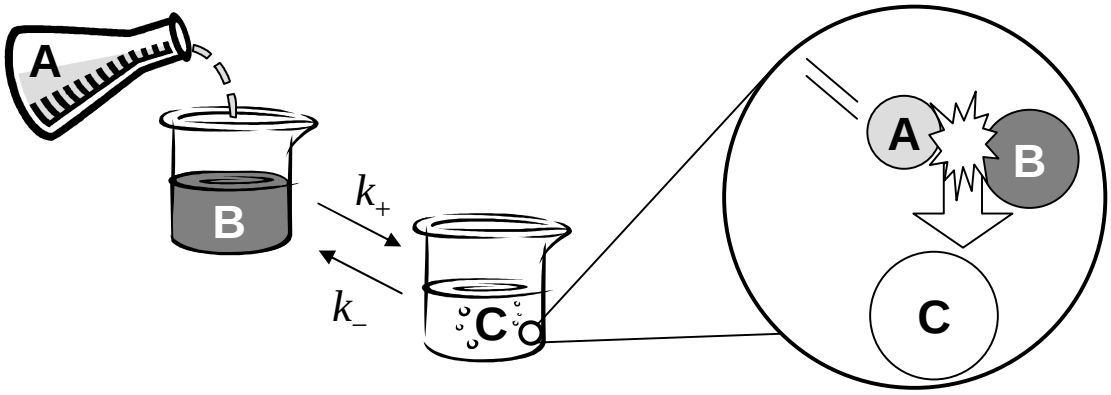


FIG. 7: For reaction networks with small reactant pools, individual reactions produce a noticeable change in the total population and are described probabilistically. The determinisitic rate constants k_+ and k_- correspond to the *average* rate of reaction. In reality, reaction events require reactant molecules to come together with enough energy to form the product. The true reaction rate will be roughly distributed exponentially about their average. We use the chemical Master equation to estimate the effect of the fluctuating reaction rates on the overall evolution of the system.

II. MOLECULAR NOISE

A. Origin of molecular noise and the mathematical characterization

In the previous section, we invoked the law of mass balance to construct the deterministic chemical rate equations, which take the form of a system of coupled nonlinear differential equations. Underlying that formulation is the implicit assumption that the concentration of the reactants varies both continuously and differentiably. For moles of reactants (*i.e.* molecule numbers of the order 10^{23}), these assumptions are perfectly justified since a change of one or two molecules in a population of 10^{23} is, for all intents and purposes, infinitesimal. That accounts for the great success of deterministic models in most macroscopic systems, including freshman chemistry labs. For small pools of reactants, however, the mathematical formulation becomes more delicate.

In the cellular environment, the reactant numbers tend to be of the order 10-1000, as we saw in the first section of these notes. A reaction altering the population by one or two therefore generates a large relative change, and the molecule numbers no longer evolve differentiably (Figure 8). Furthermore, reactions no longer occur 'continuously' over an infinitely small time interval, but rather progress in a series of steps of finite time width. By way of analogy, one can imagine the national birth rate as compared to the chances my next-door neighbor will have a baby. One often hears statements such as: "Every X minutes, a baby is born in the US." That clearly cannot be true of my next-door neighbor. Evolution of the population of an entire country can be well-described using differential equations, but individual courtship is an essentially probabilistic affair, requiring a more sophisticated formulation.

Instead of mass-balance, the conservation law we invoke for this microscopic description is probability balance. Call $P(\mathbf{n}, t)$ the probability that the system we're studying is in state $\mathbf{n}$ at time t. The vector $\mathbf{n}$ can be thought of as the 'inventory' of the system. For example,

$$\begin{aligned} n_1 &= \text{number of mRNA molecules translated from gene 1,} \\ n_2 &= \text{number of mRNA molecules translated form gene 2,} \\ n_3 &= \text{number of protein molecules translated from mRNA from gene 1,} \ldots \end{aligned} \tag{8}$$

Here, again, we can make our description as simple or as complicated as necessary. In this way, $P(\mathbf{n}, t)$ keeps track of how likely it is to find a given system in a particular state assuming all the systems began in state $\mathbf{n}_0$ at time t_0. Now we ask: How does $P(\mathbf{n}, t)$ change in time? How does it evolve?

As in the deterministic case, we must define flux into and out of our state of interest. The change in probability of being in state $\mathbf{n}$ over a small time increment Δt is equal to the probability we move *into* state $\mathbf{n}$ from a neighboring state $\mathbf{n}'$ minus the probability we move *out of* state $\mathbf{n}$ to some neighboring state $\mathbf{n}'$,

$$\Delta P(\text{being in a particular state}) = [P(\text{moving } \textit{from} \text{ a neighboring state}) - P(\text{moving } \textit{to} \text{ a neighboring state})]\, \Delta t \tag{9}$$

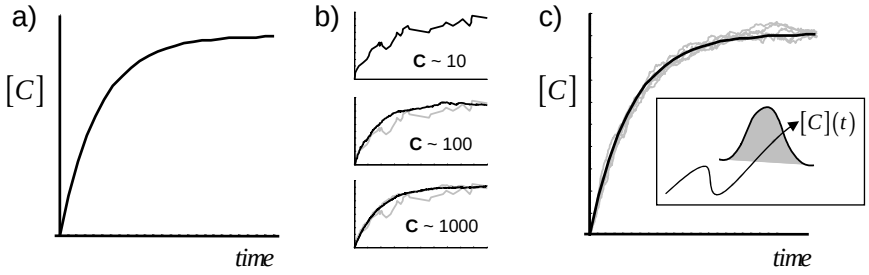
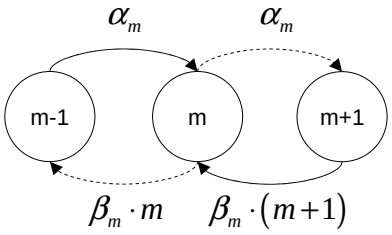


FIG. 8: a) For many reactant molecules, the species concentrations evolve both continuously and differentiably. b) When small numbers of reactants are involved, due to the probabilistic nature of individual reaction events and the finite change in molecule numbers incurred, the concentration evolves step-wise. Although as the reactant numbers increase, the relative size of the jumps decreases. c) Repeating an experiment many times, we typically obtain some repeatable averaged behavior that conforms very nearly to the deterministic description and some envelope around the average that accounts for the fluctuations in individual trajectories. *Inset:* We can imagine the full evolution of the system as composed of two parts: the determinsitic evolution of $[C](t)$ and a probability distribution for the fluctuations that moves along $[C](t)$. The width of the probability distribution scales roughly as $\frac{1}{\sqrt{N}}$ where N is the number of molecules.

FIG. 9: For unregulated mRNA synthesis, the system can move into the state of interest m in two ways, indicated by solid arrows. The system can likewise move out of the state of interest, indicated by dashed arrows.



We can further refine our definition of the probability flux by writing the gain and loss terms as a product of the probability of being in a state $\mathbf{n}$ multiplied by the transition probability $W_{\mathbf{n}\to\mathbf{n}'}$ of moving from $\mathbf{n}$ to a state $\mathbf{n}'$. Summing over all possible transitions, dividing through by Δt and taking the limit $\Delta t \to 0$, we arrive at what is called the *Master equation*,

$$\frac{\partial P(\boldsymbol{n},t)}{\partial t} = \sum_{\boldsymbol{n'}} W_{\boldsymbol{n'}\to\boldsymbol{n}} P(\boldsymbol{n'},t) - W_{\boldsymbol{n}\to\boldsymbol{n'}} P(\boldsymbol{n},t). \tag{10}$$

(For more details, see the classic review - McQuarrie, D (1967) Stochastic approach to chemical kinetics. Journal of Applied Probability 4: 413).

Since $P(\mathbf{n},t)$ is a probability distribution, we require $\sum_{\mathbf{n}} P(\mathbf{n},t) = 1$ for all t. Also note that $P(\mathbf{n},t)$ is actually a *conditional* probability distribution, conditioned by the initial condition $P(\mathbf{n},0) = \delta_{\mathbf{n},\mathbf{n}_0}$ where δ_{ij} is the *Kronecker delta*. That is to say, at $t = t_0$ the system is *guaranteed* to be in the initial state $\mathbf{n}_0$.

Throughout, we shall assume the transition probabilities $W_{\mathbf{n}\to\mathbf{n}'}$ and $W_{\mathbf{n}'\to\mathbf{n}}$ only depend upon the state at time t, irrespective of how the system arrived there. This is called the *Markov* assumption, or the assumption that our process is *Markovian*. A technical aside: The Markovian assumption is equivalent to the Master equation as I have written it above - we can derive one from the other. The more axiomatically-minded may wish to see the details in Chapters IV and V of the book by van Kampen [1].

For chemical systems, the transition probabilities W are particularly simple to compute since we assume the chemical reaction rates correspond to the mean of an exponentially distributed transition rate. Nevertheless, the chemical Master equation is very difficult to solve exactly. What makes it difficult to solve is the mixing of a *continuous* time evolution with a *discrete* evolution of the state variables. The approximation methods we shall discuss below operate by either discretizing the time evolution (simulation algorithms) or by approximating the discrete state evolution with a continuous operator (perturbation methods).

A simple example will make the above formulation more clear. Consider unregulated mRNA synthesis as we have modeled it in the previous section. Call $P(m,t)$ the probability of finding m mRNA molecules at time t. The Master

equation corresponding to this process is (Figure 9),

$$\frac{dP\left(m,t\right)}{dt}=\left[\alpha_m P\left(m-1,t\right)+\underset{GAIN}{\beta_m\left(m+1\right)P\left(m+1,t\right)}\right]-\left[\underset{LOSS}{\beta_m\, mP\left(m,t\right)}+\alpha_m P\left(m,t\right)\right], \tag{11}$$

or, rearranging slightly,

$$\frac{dP\left(m,t\right)}{dt}=\alpha_m\left[P\left(m-1,t\right)-P\left(m,t\right)\right]+\beta_m\left[\left(m+1\right)P\left(m+1,t\right)-mP\left(m,t\right)\right]. \tag{12}$$

A particularly elegant short-hand, which shall be useful in the following section on approximation methods, involves the *step-operator* $\mathbf{E}_i^k$. The operator acts by finding the i^{th} entry of $\mathbf{n}$ and incrementing it by an integer k:

$$\mathbf{E}_i^k f\left(\ldots,n_i,\ldots\right)=f\left(\ldots,n_i+k,\ldots\right). \tag{13}$$

Using the step-operator, the Master equation is written,

$$\frac{dP\left(m,t\right)}{dt}=\alpha_m\left[\mathbf{E}_1^{-1}-1\right]P\left(m,t\right)+\beta_m\left[\mathbf{E}_1^{1}-1\right]\left(m\,P\left(m,t\right)\right), \tag{14}$$

where in our example $n_1=m$.

Exercise: Show that Eq. 14 follows from Eq. 12.
Hint: Note that $\mathbf{E}_i^k\; n_i\; f(\ldots,n_i,\ldots)=(n_i+k)\; f(\ldots,n_i+k,\ldots)$

B. Moments and moment generating functions

Typically, we are not interested in the full distribution $P(\mathbf{n},t)$, but only the first two *moments*,

$$\underset{MEAN}{\left\langle n_i\right\rangle}=\sum_{\mathbf{n}=0}^{\infty}n_i P\left(\mathbf{n},t\right)\qquad\qquad\underset{MEAN-SQUARED}{\left\langle n_i^2\right\rangle}=\sum_{\mathbf{n}=0}^{\infty}n_i^2 P\left(\mathbf{n},t\right). \tag{15}$$

Instead of the mean-squared, we very often use the *variance* $\sigma_i^2=\left\langle\left(n_i-\left\langle n_i\right\rangle\right)^2\right\rangle=\left\langle n_i^2\right\rangle-\left\langle n_i\right\rangle^2$. The first two moments contain most of the physically accessible information - which is fortunate since the full distribution is in general impossible to determine exactly. Bringing us to the question: How does one solve the Master equation? *IF* the transition probabilities are *linear* or *constant* in $\mathbf{n}$, we can use the moment generating function $F(\mathbf{z},t)$. For the sake of concrete example, let us return to the mRNA synthesis above, Eq. 12. The one dimensional moment generating function $F(m,t)$ is a discrete transform defined as,

$$F\left(z,t\right)=\sum_{m=0}^{\infty}z^m P\left(m,t\right). \tag{16}$$

Electrical engineers call the moment generating function $F(z,t)$ the *z-transform* of $P(m,t)$. It is the discrete analogue of the more familiar *Laplace transform*. If you're interested in a more technical discussion, see a textbook on engineering mathematics, such as *Advanced modern engineering mathematics* by Glyn James (Pearson-Prentice Hall, 2004).

Under the transformation Eq. 16, the Master equation for mRNA synthesis becomes a simple partial differential equation,

$$\frac{\partial F\left(z,t\right)}{\partial t}=\alpha_m\left(z-1\right)F\left(z,t\right)-\beta_m\left(z-1\right)\frac{\partial F\left(z,t\right)}{\partial z}. \tag{17}$$

(Show this). We have transformed a discrete-differential equation (difficult) into a linear first-order partial differential equation (easier). The full time-dependent solution $F(z,t)$ can be determined using what is called the *method of characteristics*. Instead of the full distribution, we shall focus upon the first two moments. The moment generating function is so-named because the moments of $P(m,t)$ are generated by subsequent *derivatives* of $F(z,t)$,

$$F\left(1,t\right)=1,\quad\text{(Normalization condition on}P(\mathbf{n},t)) \tag{18}$$

$$\left.\frac{\partial F\left(z,t\right)}{\partial z}\right|_{z=1}=\left.\sum_{m=0}^{\infty}mz^{m-1}P\left(m,t\right)\right|_{z=1}=\left\langle m\left(t\right)\right\rangle, \tag{19}$$

$$\left.\frac{\partial^2 F\left(z,t\right)}{\partial z^2}\right|_{z=1}=\left.\sum_{m=0}^{\infty}m\left(m-1\right)z^{m-2}P\left(m,t\right)\right|_{z=1}=\left\langle m^2\left(t\right)\right\rangle-\left\langle m\left(t\right)\right\rangle, \tag{20}$$

$$\vdots$$

To determine the steady-state mean and variance for our mRNA example is a straightforward task. Solving the transformed Eq. 17 at steady-state $\frac{\partial F}{\partial t} = 0$, we have,

$$F^s(z) = \exp\left[\frac{\alpha_m}{\beta_m}(z-1)\right]. \tag{21}$$

The steady-state moments follow immediately,

$$\left.\frac{\partial F^s}{\partial z}\right|_{z=1} = \frac{\alpha_m}{\beta_m} = \langle m\rangle^s \tag{22}$$

$$\left.\frac{\partial^2 F^s}{\partial z^2}\right|_{z=1} = \left(\frac{\alpha_m}{\beta_m}\right)^2 = \left\langle m^2\right\rangle^s - \langle m\rangle^s, \tag{23}$$

with steady-state variance,

$$\sigma^2 = \frac{\alpha_m}{\beta_m}. \tag{24}$$

Having mean equal to the variance is the footprint of a *Poisson process*. We can measure how close to Poisson distributed a given process may be by considering the *Fano factor*, $\frac{\sigma^2}{\langle m\rangle}$. In our case (since our process is Poisson distributed), the Fano factor is 1,

$$\frac{\sigma^2}{\langle m\rangle} = 1. \tag{25}$$

Alternately, the fractional deviation $\eta = \sqrt{\frac{\sigma^2}{\langle m\rangle^2}}$ is a *dimensionless* measure of the fluctuations and often provides better physical insight than the Fano factor. In our mRNA example,

$$\eta = \frac{1}{\sqrt{\langle m\rangle}}, \tag{26}$$

substantiating the rule-of-thumb that relative fluctuations scale roughly as the *square-root* of the number of reactants [2]. We shall exploit this scaling in the next section on approximation methods.

Exercise: Write the Master equation for unregulated mRNA *and* protein synthesis. Solve for the steady-state mean and variance using a two-dimensional moment generating function $F(z_1, z_2, t) = \sum_{m,p}^{\infty} z_1^m z_2^p P(m,p,t)$. Calculate the steady-state fractional deviation of the protein number, $\eta_p = \frac{\sigma_p}{\langle p\rangle}$.

The answer to the previous exercise provides a crucial insight into the origins of noise in genetic circuits. For constitutive expression of protein as we have modeled it, the fractional deviation in protein number is,

$$\frac{\sigma_p^2}{\langle p\rangle^2} = \frac{1}{p}\left[1 + \frac{\alpha_p}{\beta_m + \beta_p}\right]. \tag{27}$$

For $\beta_m \gg \beta_p$, the result simplifies to,

$$\frac{\sigma_p^2}{\langle p\rangle^2} = \frac{1}{p}\left[1 + \frac{\alpha_p}{\beta_m}\right] \equiv \frac{1}{p}\left[1 + b\right]. \tag{28}$$

We call the parameter $b = \frac{\alpha_p}{\beta_m}$ the *burstiness* of the gene. The burstiness is a measure of the amplification of transcription noise by translation, since each errant transcript is amplified by $b = \frac{\alpha_p}{\beta_m}$ =(protein molecules / mRNA) $\times$ (average mRNA lifetime) [3]. What is surprising is that we can observe this burstiness experimentally (Figure 10)!

It is important to bear in mind that there are many other sources of noise beyond molecular noise, including the doubling of the gene copy number during replication, random partitioning of cell contents during division, *etc.* Focusing upon the molecular noise, we model the system not in terms of differential equations but rather as a probability-conservation, leading to the chemical Master equation. In the simple example we considered of *unregulated* protein synthesis, we were able to solve for the entire probability distribution using moment generating functions. But this approach has a very strong limitation - it can only be used when the transition probabilities are linear or constant in the state variables. What this means physically is that we cannot use moment generating functions to solve Master equations that describe systems with bimolecular reactions (such as dimerization), systems with feedback, or, in short, systems with *any regulation whatsoever!* How then are we to deal with realistic models of regulated gene expression? We must resort to approximation methods.

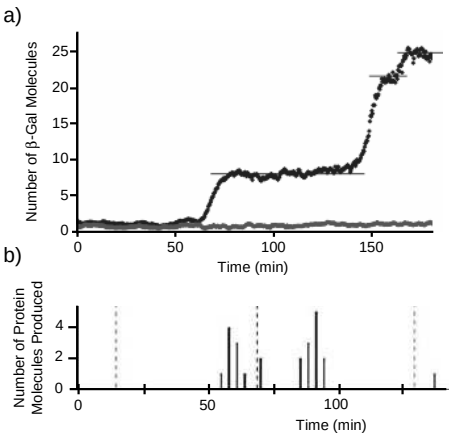


FIG. 10: Bursts of protein production. The Xie lab has developed sophisticated methods to observe bursts of protein off of individual mRNA transcribed from a highly repressed *lac* promoter. a) Trapping individual *E. coli* cells in a microfluidic chamber, it is possible to observe the step-wise increase of β-gal off the *lacZ* gene. Cai L, Friedman N, Xie XS (2006) Stochastic protein expression in individual cells at the single molecule level. Nature 440:358-362.

b) In a second study, a fluorescent marker *Venus* is fused to a membrane localizing protein Tsr and again used to observe bursts in protein expression off a highly repressed *lac* promoter. The dashed lines denote a cell division. Yu J, Xiao J, Ren X, Lao K, Xie XS (2006) Probing gene expression in live cells, one protein molecule at a time. Science 311:1600-1603.

III. APPROXIMATION METHODS IN STOCHASTIC SYSTEMS

There are two broad classes of approximation methods - numerical simulation algorithms and perturbation methods. Each has clear advantages and disadvantages.

1. Numerical Simulation: The classic reference for these types of algorithms is the paper by Gillespie: Gillespie DT (1977) Exact simulation of coupled chemical reactions. Journal of Chemical Physics 81:2340.

   The method simulates a *single* trajectory $\mathbf{n}(t)$ that comes from the unknown probability distribution $P(\mathbf{n}, t)$ characterized by the Master equation.

2. Perturbation Methods: The classic reference for the perturbation scheme we will consider is the paper by van Kampen: van Kampen NG (1976) Expansion of the Master equation. Advances in Chemical Physics 34:245. Another often-used and related method is Langevin's approach, described in detail in the appendix of Swain [4].

   For perturbation methods, the discrete jump in $\mathbf{n}(t)$ that occurs with each reaction is treated as a *nearly* continuous and differentiable process. The 'nearly' is what we use as a perturbation parameter.

These two approaches are best understood and contrasted in light of a simple example. Let us return again to the symmetric toggle switch considered previously (see Figure 4). Under the assumption that the transcription factor/ DNA binding is fast, the deterministic rate equations are,

$$\frac{dm_1}{dt} = \alpha_m \cdot g_R(r_2) - \beta_m\, m \quad \frac{dm_2}{dt} = \alpha_m \cdot g_R(r_1) - \beta_m\, m$$
$$\frac{dr_1}{dt} = \alpha_p\, m_1 - \beta_p\, r_1 \quad \frac{dr_2}{dt} = \alpha_p\, m_2 - \beta_p\, r_2, \tag{29}$$

where $g_R(r_j)$ is the promoter activity function (see Figure 2).

WARNING: The promoter activity function $g_R(r)$ assumes rapid transcription factor / DNA binding. Care must be taken when using this lumped description in stochastic models [5, 6]. For simplicity and ease of presentation, we shall assume in what follows that the transcription factor / DNA binding is sufficiently fast. A more detailed model would include the state of the promoter (bound/unbound) and the associated binding/unbinding rates.

As before, we shall simplify the model by assuming the mRNA level equilibrates much faster than the protein, *i.e.* $\frac{dm}{dt} \approx 0$. The reduced system is then,

$$\frac{dr_i}{dt} = \gamma^0 \cdot g_R(r_j) - \beta_p\, r_i. \tag{30}$$

Notice the protein synthesis rate has the form $\frac{\alpha_p}{\beta_m} \times \alpha_m \cdot g_R(r_i) = b \times$ transcription rate, where b is the *burstiness* of the promoter discussed above. The burstiness will play an important role in the stochastic description we develop below.

A. Stochastic description of reaction networks

It is most convenient to codify our reaction network in terms slightly different from the deterministic description. We shall consider each individual reaction in turn and record the *reaction propensity* ν and the reaction *stoichiometry* $\mathbf{S}$, where the propensity tells us how *frequently* a reaction occurs and the stoichiometry tell us how *much the system is changed* when the reaction is completed. For example, in the reduced toggle switch model we have four reactions, each representing the combined effect of several more elementary reactions. We have two reactions representing protein degradation. Each degradation reaction decreases the population of r_i by 1 and proceeds at a rate $\beta_p\, r_i$,

$$r_1 \xrightarrow{\nu_1} r_1 - 1, \quad \nu_1 = \beta_p\, r_1 \tag{31}$$

$$r_2 \xrightarrow{\nu_2} r_2 - 1, \quad \nu_2 = \beta_p\, r_2 \tag{32}$$

Likewise, we have two reactions representing protein synthesis, but here we must take care in how we assign the rate and how we assign the stoichiometry. The synthesis reaction is the combined effect of transcription and translation. We know that protein is generated as a burst of average size $b = \frac{\alpha_p}{\beta_m}$ from a single translated mRNA produced at a rate $\alpha_m \cdot g_R(r)$. We therefore write the synthesis reactions as,

$$r_1 \xrightarrow{\nu_3} r_1 + b, \quad \nu_3 = \alpha_m \cdot g_R(r_2) \tag{33}$$

$$r_2 \xrightarrow{\nu_4} r_2 + b, \quad \nu_4 = \alpha_m \cdot g_R(r_1) \tag{34}$$

Generally, we record the reaction propensities in a vector ν and the stoichiometries in a matrix $\mathbf{S}$ defined such that when the j^{th} reaction occurs it increments the i^{th} reactant by an integer S_{ij} : $n_i \xrightarrow{\nu_j} n_i + S_{ij}$. The collection of the elements S_{ij} compose the stoichiometry matrix. In our reduced toggle switch example,

$$\begin{matrix} \nu_1 & \nu_2 & \nu_3 & \nu_4 \end{matrix}$$
$$\begin{bmatrix} -1 & 0 & b & 0 \\ 0 & -1 & 0 & b \end{bmatrix} \begin{matrix} r_1 \\ r_2 \end{matrix}$$

where each column of the stoichiometry matrix corresponds to a particular reaction and each row to a particular reactant. Using this notation, the deterministic rate equations are written,

$$\frac{d\mathbf{r}}{dt} = \mathbf{S} \cdot \nu, \tag{35}$$

so the separation of stoichiometry from propensity is of no importance at this level of description, but it becomes very important when we want to characterize the fluctuations.

B. Stochastic simulation algorithms - Gillespie's method

As we have said, for Master equations with nonlinear transition probabilities, the full distribution $P(\mathbf{n}, t)$ can rarely be solved exactly. Gillespie's algorithm is a method by which an individual sample path, starting at a given initial point, can be simulated in time such that it conforms to the unknown probability distribution we seek; that is, for a sufficiently large population of sample paths, the inferred probability distribution is as near to the exact solution as we wish. The algorithm proceeds in 3 steps:

1. The propensities ν_j are used to form a probability distribution for the next reaction *time*, τ and τ is drawn from this distribution.

2. The propensities are used to form a probability distribution for *which reaction in the network will occur next*, *i.e.* which of the v_j's is completed at time $t + \tau$. Call the reaction index μ.

3. The time is advanced $t \to t + \tau$ and the state is updated using the stoichiometry matrix - for each reactant, $n_i \to n_i + S_{i\mu}$. Repeat ...

In this way, we generate a discrete time series for the reactant numbers. As an example, in the toggle switch we assume the transcription factors are active as dimers and that dimerization proceeds immediately to 100% completion. Furthermore, we assume each promoter has two operator site to which the dimers can bind independent of one another

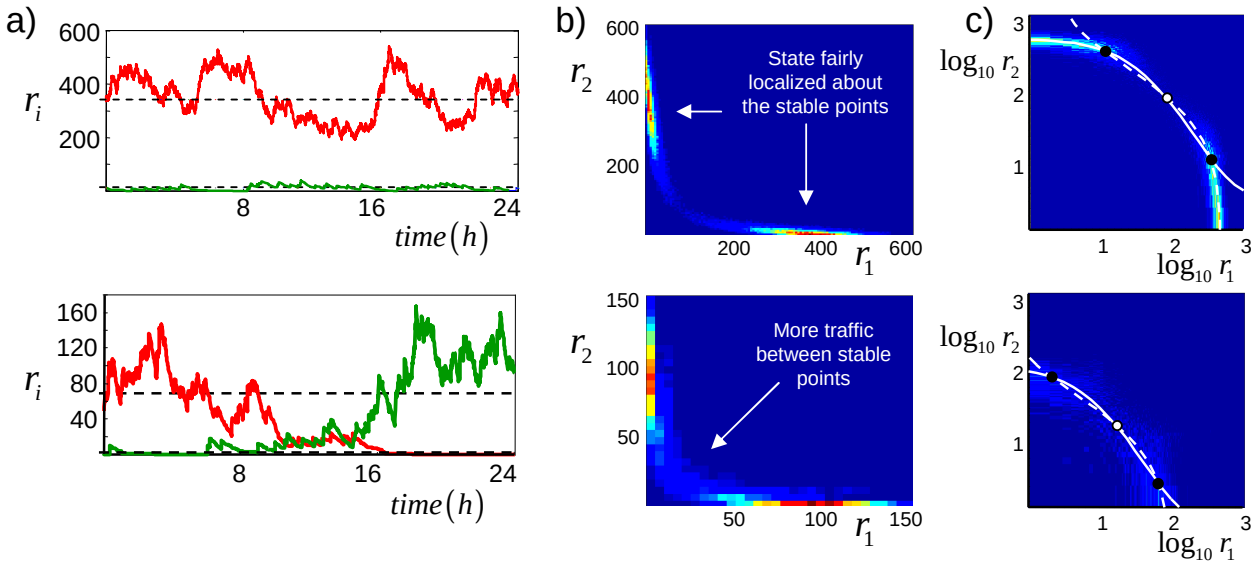


FIG. 11: a) Time series data for a stochastic simulation of the reduced toggle switch model using Gillespie's method. The top plot shows a typical trajectory, with high and low states clearly separated. The bottom plot is an example of a stochastic switching event where fluctuations cause the system to change states, moving from one equilibrium point to the other. The deterministic steady-states are indicated as dashed lines. The lower plots have $\frac{1}{5}$ the number of molecules of those above (see text). b) Collecting together a large number of simulations, we are able to make a histogram density plot in two dimensions that approximates the steady-state probability distribution of the process. Density goes from red (high) to blue (low). (Top) Notice the system is largely confined to an elliptical region close to the two equilibrium points $(r_i, r_j) = (11, 341)$. Using the approximation methods in the next section, we will characterize this elliptical region in more detail. (Bottom) Due to the increased magnitude of the fluctuations, the state is less localized about the macroscopic equilibrium points, and often travels between them. c) The probability density distribution on a log-log scale to facilitate comparison with Figure 5. Notice the fluctuations tend to move along the nullclines. Why do you think that is?

to repress transcription ω-fold. Under these conditions, the promoter activity function $g_R(r) = \frac{\left(1+\frac{q_R}{\omega}(2+q_R)\right)}{(1+q_R)^2}$, where $q_R = \frac{r}{2\,K_R}$ and K_R is the transcription factor / DNA dissociation constant for the *dimer*. Using the nominal parameter values: $\alpha_m = 1$ nM/min, $\beta_p^{-1} = 50$ min, $K_R = 25$ nM and $\omega = 200$ fold, with a burst parameter of $b = 10$ and starting from one of the stable equilibrium points $(r_1, r_2) = (11, 341)$, Figure 11a (Top) shows the result of typical simulations. (The matlab code used to generate the data in the figures is appended to the end of this note set). Collecting together a large sample of independent simulations, we can estimate the underlying probability distribution, Figure 11b (Top), and plot the same on a log-log scale, Figure 11c (Top), in analogy with Figure 5. The lower plots are the the same as the top, except that we have $\frac{1}{5}$ the number of molecules, using parameter values: $\alpha_m = \frac{1}{5}$ nM/min, $\beta_p^{-1} = 50$ min, $K_R = 5$ nM and $\omega = 200$ fold, with a burst parameter of $b = 10$ and starting from one of the stable equilibrium points $(r_1, r_2) = (\frac{11}{5}, \frac{341}{5})$. There fluctuations are substantially larger in the lower plots, and stochastic switching between equilibria is common.

The advantages of the stochastic simulation algorithm is that it is simple to program and provides an output trajectory that *exactly* conforms to the solution distribution of the chemical Master equation. The disadvantages are that the original algorithm is computationally expensive and the method does not scale well as the number of molecules gets large (although there are approximate algorithms that alleviate some of the computational burden). Most importantly, the method suffers from the same limitations as any numerical scheme - there is a lack of deep insight into the model and it is difficult to systematically explore different regions of parameter space. Nevertheless, Gillespie's algorithm is the benchmark against which all other methods of solving the chemical Master equation are measured.

C. Perturbation expansions - Linear Noise Approximation

Often we can gain a better sense of a particular model by examining certain limiting regimes. For example, we previously simplified the description of the toggle switch dynamics by considering the limit of fast transcription factor

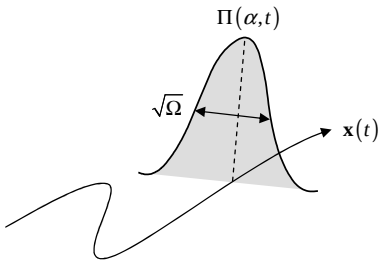


FIG. 12: In the linear noise approximation, we make the assumption that the deterministic trajectory $\mathbf{x}$ can be meaningfully separated from the fluctuations α as $n_i = \Omega\ x_i + \sqrt{\Omega}\ \alpha_i$, where Ω is an extensive measure of the system size (carrying the units of volume). The picture we have in mind is a probability distribution for the fluctuations of width $\sqrt{\Omega}$, sliding along the deterministic trajectory $\mathbf{x}$.

/DNA binding dynamics and fast mRNA degradation. The approximation method that we describe in this section examines system behavior in the limit of large numbers of reactant molecules.

We have already seen that as the number of molecules increases, the system evolution becomes more smooth and the deterministic formulation becomes more appropriate (Figure 8). The linear noise approximation exploits this behavior and rests upon the supposition that the deterministic evolution of the reactant concentrations, call them $\mathbf{x}$, can be meaningfully separated from the fluctuations, call them α, and that these fluctuations scale roughly as the *square-root* of the number of molecules. We introduce an extensive parameter Ω that carries the units of volume and is directly proportional to the molecule numbers, allowing the molecule numbers to be written

$$n_i = \Omega x_i + \sqrt{\Omega}\alpha_i. \tag{36}$$

We are lead to the square-root scaling of the fluctuations by the suggestion from Poisson statistics. Recall that for a Poisson process, the fractional deviation is proportional to the square-root of the number of molecules (see Eq. 26). The picture that underlies the assumption above is that of a determinsitic, reproducible trajectory surrounded by a cloud of fluctuations. We would like a set of equations that govern the change in the deterministic part $\mathbf{x}$ and an equation that governs the change in the probability distribution of the fluctuations, call it $\Pi(\alpha, t)$, centered upon $\mathbf{x}$ (Figure 12).

With the assumption above, we are in a position to write the chemical Master equation in a compact, and convenient manner (see [7] for more details), but first we must make a distinction between the macroscopic reaction propensities ν_j discussed in Section III A above and the *microscopic* propensities $\tilde{\nu}_j$ that will appear in the Master equation below. The distinction lies in the fact that microscopically we must keep track of *each individual* reactant molecule. For example, consider a tri-molecular reaction such as : $2\ X_1 + X_2 \xrightarrow{a} 3\ X_1$. The macroscopic and microscopic propensities are written,

$$\nu_j\left(\frac{n}{\Omega}\right) = a\ \frac{n_1^2\ n_2}{\Omega^3} \qquad \tilde{\nu}_j\left(\frac{n}{\Omega}\right) = a\ \frac{n_1\ (n_1-1)\ n_2}{\Omega^3}, \tag{37}$$

where the $(n_1 - 1)$ term appears in the microscopic propensity since we need at least $n_1 = 2$ for the reaction to proceed. Obviously, the microscopic propensities must coincide with the macroscopic propensities in the limit of very many molecules, $\lim_{\Omega\to\infty} \tilde{\nu}_j\left(\frac{\mathbf{n}}{\Omega}\right) = \nu_j\left(\frac{\mathbf{n}}{\Omega}\right)$. With $\tilde{\nu}$, we are able to explicitly write the chemical Master equation for a network of N reactions involving d species as,

$$\frac{dP\left(\mathbf{n}, t\right)}{dt} = \Omega \sum_{j=1}^{N} \left[\left(\prod_{i=1}^{d} \mathbf{E}_i^{-S_{ij}} \right) - 1 \right] \tilde{\nu}_j\left(\mathbf{n}, \Omega\right) P\left(\mathbf{n}, t\right). \tag{38}$$

(where we have used the step-operator $\mathbf{E}_i^k$ defined in Eq. 13). We have repeatedly emphasized that if the transition probabilities $\tilde{\nu}_j$ are *nonlinear* functions (as they must be for models of regulated expression), then there is no systematic way to obtain an exact solution of the Master equation, and we must resort to approximation methods. The linear noise approximation, which is the subject of this section, proceeds in three steps.

1. First, we replace the full probability distribution $P(\mathbf{n}, t)$ by the probability distribution for the fluctuations $\Pi(\alpha, t)$ centered on the macroscopic trajectory $\mathbf{x}$,

$$P(\mathbf{n}, t) \mapsto \Omega^{-\frac{d}{2}} \Pi(\alpha, t). \tag{39}$$

The pre-factor $\Omega^{-\frac{d}{2}}$ comes from the normalization of the probability distribution.

2. Recall that what makes the Master equation difficult to solve exactly is the discrete evolution over state-space characterized by the step-operator $\mathbf{E}_i^k$. To make headway, we must find some *continuous* representation of the action of the operator. To that end, consider the action of the operator - it increments the i^{th} species by an *integer* k. Using the assumption above (Eq. 36), we write,

$$\mathbf{E}_i^k \; f(\ldots,\, n_i, \ldots) = f(\ldots,\, n_i + k, \ldots) = f(\ldots,\, \Omega\, x_i + \sqrt{\Omega}\, \left(\alpha_i + \frac{k}{\sqrt{\Omega}}\right), \ldots). \tag{40}$$

The term $\frac{k}{\sqrt{\Omega}}$ becomes negligibly small as $\Omega \to \infty$, suggesting a Taylor series around $\frac{k}{\sqrt{\Omega}} = 0$,

$$f(\ldots,\, \Omega\, x_i + \sqrt{\Omega}\, \left(\alpha_i + \frac{k}{\sqrt{\Omega}}\right), \ldots) \approx f(\ldots,\, n_i, \ldots) - \frac{k}{\sqrt{\Omega}} \frac{\partial f}{\partial \alpha_i} + \frac{k^2}{2\,\Omega} \frac{\partial^2 f}{\partial \alpha_i^2} + \ldots \tag{41}$$

allowing us to approximate the *discrete* step-operator by a *continuous* differential operator,

$$\mathbf{E}_i^k \approx \left[1 - \frac{k}{\sqrt{\Omega}} \frac{\partial}{\partial \alpha_i} + \frac{k^2}{2\,\Omega} \frac{\partial^2}{\partial \alpha_i^2} + \ldots\right]. \tag{42}$$

(This is called the *Kramers-Moyal expansion* of the step-operator).

3. Finally, to remain consistent in our perturbation scheme, we must likewise expand the propensities in the limit $\Omega \to \infty$,

$$\tilde{\nu}_j \left(\frac{n}{\Omega}\right) \approx \nu_j\,(\mathbf{x}) + \sqrt{\Omega} \sum_{i=1}^{d} \frac{\partial \nu_j}{\partial x_i}\; \alpha_i + \ldots \tag{43}$$

Putting all of this together, taking care to write $\frac{\partial \Pi}{\partial t}$ using the chain rule,

$$\frac{\partial P}{\partial t} = \Omega^{\frac{1-d}{2}} \left[\Omega^{-\frac{1}{2}} \frac{\partial \Pi}{\partial t} - \sum_{i=1}^{d} \frac{dx_i}{dt} \frac{\partial \Pi}{\partial \alpha_i}\right], \tag{44}$$

we collect Eq. 38 in like powers of $\sqrt{\Omega}$ taking the limit $\Omega \to \infty$. To zero'th order (Ω^0), we have,

$$\Omega^0 : \qquad \frac{dx_i}{dt} \frac{\partial \Pi}{\partial \alpha_i} = \left[\frac{\alpha_p}{\beta_m}\; \alpha_m \cdot g_R(x_j) - \beta_p\; x_i\right]\; \frac{\partial \Pi}{\partial \alpha_i}. \tag{45}$$

This system of equations is identically satisfied if $\mathbf{x}$ obeys the deterministic rate equations,

$$\frac{dx_i}{dt} = \frac{\alpha_p}{\beta_m}\; \alpha_m \cdot g_R(x_j) - \beta_p\; x_i \equiv f_i(\mathbf{x}). \tag{46}$$

At the next order, $\sqrt{\Omega}^{-1}$, we have the equation characterizing the probability distribution for the fluctuations,

$$\sqrt{\Omega}^{-1} : \qquad \frac{\partial \Pi}{\partial t} = -\sum_{i,j} A_{ij} \partial_i (\alpha_j \Pi) + \frac{1}{2} \sum_{i,j} B_{ij} \partial_{ij} \Pi, \tag{47}$$

where $\partial_i \equiv \frac{\partial}{\partial \alpha_i}$ and,

$$A_{ij}(t) = \left.\frac{\partial f_i}{\partial x_j}\right|_{\mathbf{x}(t)} \qquad \mathbf{B} = \mathbf{S} \cdot \mathrm{diag}[\nu] \cdot \mathbf{S}^T. \tag{48}$$

For our toggle switch example, the coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ are,

$$\mathbf{A} = \begin{bmatrix} -\beta_p & b\; \alpha_m\; g_R'(r_2) \\ b\; \alpha_m\; g_R'(r_1) & -\beta_p \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b^2\; \alpha_m\; g_R(r_2) + \beta_p\; r_1 & 0 \\ 0 & b^2\; \alpha_m\; g_R(r_1) + \beta_p\; r_2 \end{bmatrix}. \tag{49}$$

Exercise: Expand the Master equation and verify Eqs. 46 and 47. Include the $O(\Omega^{-1})$ correction to Eq. 47 for the toggle switch model.

We now have in hand a coupled set of nonlinear differential equations that govern the deterministic evolution of the system, which happen to coincide with the macroscopic reaction rate equations. We also have a partial differential equation that characterizes the probability distribution of the fluctuations. Some comments are in order:

1. The equation describing $\Pi(\alpha, t)$ is an example of a diffusion equation with drift. In the context of stochastic systems, this equation is called the *Fokker-Planck* equation. The equation derived above, Eq. 47, is a special sub-class of Fokker-Planck equations since the coefficient matrices $\mathbf{A}$ and $\mathbf{B}$ are *linear* in α. You can prove (see [1]) that for linear coefficient matrices the solution distribution is *Gaussian* for all time. Furthermore, the moments of α are easily computed from Eq. 47 by multiplying with α_i and integrating by parts to give,

$$\frac{d\langle\alpha\rangle}{dt} = \mathbf{A}\cdot\langle\alpha\rangle \tag{50}$$

If we choose the initial condition of $\mathbf{x}$ to coincide with the initial state of the system $\mathbf{n}_0$ (*i.e.* $\langle\alpha(0)\rangle = 0$), then $\langle\alpha\rangle = 0$ for all time. Without loss of generality, then, we set $\langle\alpha\rangle = 0$. For the covariance $C_{ij} = \langle\alpha_i\alpha_j\rangle - \langle\alpha_i\rangle\langle\alpha_j\rangle = \langle\alpha_i\alpha_j\rangle$, integration by parts of Eq. 47 gives,

$$\frac{d\mathbf{C}}{dt} = \mathbf{A}\cdot\mathbf{C} + \mathbf{C}\cdot\mathbf{A}^T + \mathbf{B}. \tag{51}$$

The covariance determines the width of $\Pi(\alpha, t)$ as it moves along $\mathbf{x}(t)$. The full distribution satisfying Eq. 47 is the Gaussian,

$$\Pi(\alpha, t) = \left[(2\pi)^d \,\mathrm{det}\mathbf{C}(t)\right]\ \exp\left[-\frac{1}{2}\alpha^T\cdot\mathbf{C}(t)\cdot\alpha\right], \tag{52}$$

with covariance matrix $\mathbf{C}(t)$ determined by Eq. 51.

2. The Fokker-Planck equation provides an even deeper insight into the physics of the process. Notice $\mathbf{A}$ is simply the *Jacobian* of the deterministic system, evaluated pointwise along the macroscopic trajectory. As such, it represents the *local damping* or dissipation of the fluctuations [8]. The diffusion coefficient matrix $\mathbf{B}$ tells us how much the microscopic system is changing at each point along the trajectory. As such, it represents the *local fluctuations*. The balance of these two competing effects - dissipation and fluctuation - occurs at steady-state and is described by the fluctuation-dissipation relation (Eq. 51 with $\frac{d\mathbf{C}}{dt} = 0$),

$$\mathbf{A}_s\cdot\mathbf{C}_s + \mathbf{C}_s\cdot\mathbf{A}_s^T + \mathbf{B}_s = 0 \tag{53}$$

where each of the matrices is evaluated at a stable equilibrium point of the deterministic system. For the physicists, compare Eq. 47 with Kramer's equation for a Brownian particle trapped in a potential well. You will see that $\mathbf{A}$ plays the role of the curvature of the potential (the spring constant), and $\mathbf{B}$ plays the role of temperature.

3. Notice the role stoichiometry plays in the magnitude of the fluctuations through the coefficient matrix $\mathbf{B} = \mathbf{S}\cdot\mathrm{diag}[\nu]\cdot\mathbf{S}^T$. Although $\mathbf{A}$ is unchanged by lumping together the propensity and stoichiometry by setting $\gamma^0 = b\times\alpha_m$, the fluctuation matrix $\mathbf{B}$ is *not*! Let me impress upon you once again how important it is to separate stoichiometry from propensity in stochastic models. So important, in fact, that I will write it as a warning:

WARNING: In a stochastic description, it is of the utmost importance to distinguish between the reaction *stoichiometry* and the reaction *propensity* [7].

Returning to our toggle switch example, consider the fluctuations around one of the two stable steady-states. The steady-state probability distribution is

$$\Pi(\boldsymbol{\alpha})_s = \frac{1}{\sqrt{(2\pi)^d\,\det\left[\boldsymbol{C}_s\right]}}\exp\left[-\frac{1}{2}\boldsymbol{\alpha}^T{\boldsymbol{C}_s}^{-1}\boldsymbol{\alpha}\right]. \tag{54}$$

With the choice of promoter activity function and nominal rate parameters used to generate the stochastic simulations in the previous section, we have,

$$\mathbf{C}_s = \begin{bmatrix} 87.14 & -684.00 \\ -684.00 & 9498.96 \end{bmatrix}. \tag{55}$$

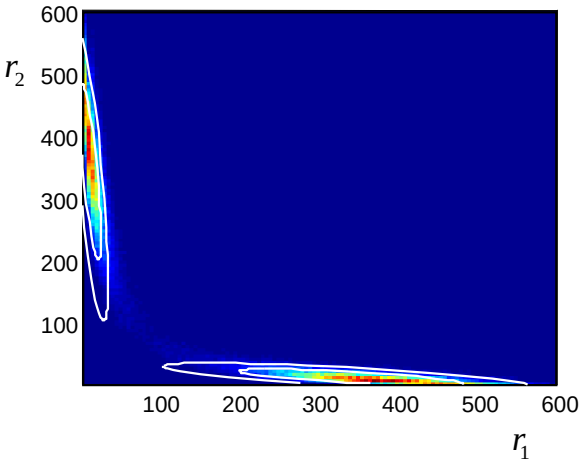


FIG. 13: The steady-state probability distribution for the fluctuations about the equilibrium points in the reduced toggle switch model. The two ellipses surrounding each stable point are the first- and second-standard deviation ellipses of the Gaussian distribution computed from the linear noise approximation, Eq. 54, with Eq. 55 and $\Omega = 1$. Compare Figure 11b.

For more examples of the linear noise approximation applied to multistable systems, see Tomioka R,Kimura H, Kobayashi TJ, Aihara K (2004) Multivariate analysis of noise in genetic regulatory networks. Journal of Theoretical Biology 229: 501.

Comparing the approximation of the steady-state probability distribution to the distribution inferred from stochastic simulation, we see the two compare well (Figure 13).

As $\Omega \to \infty$, stochastic simulation becomes more costly, but the analytic approximation becomes more reliable. In that way, we are able to cover the various regimes required in the stochastic modeling of genetic circuits.

IV. RECOMMENDED READING AND REFERENCES

I have included general review articles that are predominantly tutorial. More specialized applications and refinements of the ideas we've considered in these notes are available in the references of the papers cited below.

Two general survey articles from which the discussion on genetic circuits was adapted are,

- Bintu L, Buchler NE, Garcia HG, Gerland U, Hwa T, Kondev J, Phillips R (2005). Transcriptional regulation by the numbers: models. Current Opinions in Genetics & Development 15: 116
- Bintu L, Buchler NE, Garcia HG, Gerland U, Hwa T, Kondev J, Kuhlman T, Phillips R (2005). Transcriptional regulation by the numbers: applications. Current Opinions in Genetics & Development 15: 125

There are several recent review articles on noise in genetic systems, including

- Kaern M, Elston TC, Blake WJ, Collins JJ (2005) Stochasticity in gene expression: From theories to phenotypes. Nature Reviews Genetics 6: 451.
- Raser JM, O'Shea EK (2005) Noise in gene expression: origins, consequences, and control. Science 309: 2010.
- Thattai M, van Oudenaarden A (2001) Intrinsic noise in gene regulatory networks. PNAS 98: 8614-8619.

that themselves contain many useful references. For the approximation methods discussed, the seminal references are by Gillespie and van Kampen,

- Gillespie DT (1977) Exact simulation of coupled chemical reactions. Journal of Chemical Physics 81:2340.
- van Kampen NG (1976) Expansion of the Master equation. Advances in Chemical Physics 34:245

and a couple of recent applications of the linear noise approximation in the context of genetic circuits,

- Elf J, Ehrenberg M (2003) Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. Genome Research 13: 2475
- Scott M, Ingalls B, Kaern M (2006) Estimations of intrinsic and extrinsic noise in models of nonlinear genetic networks. Chaos 16: 026107.

A terrific general introduction to stochastic processes is van Kampen's book

- van Kampen NG, Stochastic Processes in Physics and Chemistry. (North-Holland-Elsevier, 1992)

---

[1] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry* (North-Holland-Elsevier, 1992), chapter VII.
[2] A. Bar-Evan, J. Paulsson, N. Maheshri, M. Carmi, E. O'Shea, Y. Pilpel and N. Barkai, Nature Genetics **38**, 636 (2006).
[3] M. Thattai and A. van Oudenaarden, PNAS **98**, 8614 (2001).
[4] P. S. Swain, Journal of Molecular Biology **344**, 965 (2004).
[5] A. M. Walczak, J. N. Onuchic, P. G. Wolynes, Proceedings of the National Academy of Science U.S.A. **102**, 18926 (2005).
[6] C. Rao and A. P. Arkin, Journal of Chemical Physics **118**, 4999 (2003).
[7] J. Elf and M. Ehrenberg, Genome Research **13**, 2475 (2003), see supplementary material available online at http://www.genome.org.
[8] F. Ali and M. Menzinger, Chaos **9**, 348 (1999).

V. SAMPLE MATLAB CODE FOR THE GILLESPIE SIMULATION OF THE REDUCED TOGGLE SWITCH

```
function [tOut, YOut]=toggle(inOM,tmax)
% Stochastic simulation of reduced toggle switch based on Gillespie's direct method:
% Gillespie DT (1977) Exact stochastic simulation of coupled chemical
% reactions. Journal of Physical Chemistry, 81:2340-61.

global yInt; % Defines the vector of initial conditions as a global
global am bp b n Kr w OMEGA;% Defines the reaction parameters as globals

if nargin > 0      % User can define these two parameters at the command line
    OMEGA=inOM;       % System size
    totalTime=tmax;  % Total time of the simulation
else
    OMEGA=1;
    totalTime=1500;
end

% Nominal rate parameters
am=1; % mRNA synthesis
bp=1/50; % Degradation of mRNA and protein
b=10; % 'Burstiness' - ap/bm

% Details of the repressor / DNA binding
Kr=25; w=200; % Dissociation constant, and fold repression
%                   r1 r2
yInt=round(OMEGA*[ 11 341])';     %Initial conditions

M = 4           ; %Number of reactions
N = length(yInt); %Number of reactants

% Matlab is very bad at alocating resources dynamically
% It seems to run faster if the memeory is set aside before
% the algorithm is actually run.

[R_mu] = defineReactions(N,M) ;
X=zeros(N,5000);t=zeros(1,5000);
[X(:,1),fixedflagX] = initialNumber(N)    ;

rand('state',sum(100*clock));  %Set the uniform random number generator

tdom = 0:1/10:totalTime; % We break the time series into convenient packets, and only
ltime = length(tdom);    % save every few data points

%Preallocates enough memory to store the output
tOut=zeros(1,ltime);
YOut=zeros(N,ltime);
rxCount=1;

%Assigns the inital output data, scaled as concentration
YOut(:,1)=yInt/OMEGA;

%------------------------ MAIN LOOP -----------------------
% I really like to watch the waitbar tick away - but if you find it
% distracting, comment out the next line and the two other lines indicated
% below
```

```
h = waitbar(0,'Waiting for stochastic simulation...');% *** DELETE to remove waitbar ***

k = 2;
while k <= ltime %main loop
    waitbar(k/ltime) % *** DELETE to remove waitbar ***
    while t(rxCount) <= tdom(k) %Breaks time domain into length(tdom) independent pieces

    %step 1: Calculate a_mu & a_0
    % This step calculates the rate at which *any* reaction is expected to occur
        a_mu = h_mu(X(:,rxCount),N,M);
        a_0  = sum(a_mu)   ;

    %Step 2: calculate tau and mu using random number generators
    % Assuming an exponential distribution of reaction rates, the algorithm
    % draws a time for the next reaction to occur (tau) and decides which
    % reaction it will be (next_mu)
        r1  = rand;
        tau = (1/a_0)*log(1/r1);

        r2 = rand;
        for i=1:M
            if (sum(a_mu(1:i)) >= r2*a_0)  next_mu=i;  break; end
        end

    %Step 3: carry out the reaction mu_next
    % Carries out the reaction and advances the time
        t(rxCount+1) =t(rxCount) + tau;

        prod = R_mu(next_mu,1:N)  ;  %carry out reaction next_mu
        last = X(:,rxCount) ;
        for i=1:N %Checks to see if any reactants are fixed
               if(fixedflagX(i)~=1)
                last(i) = last(i)+prod(i);
               end
        end
       X(:,rxCount+1) = last;
       rxCount=rxCount+1;
    end; %end while t(end) < tdom(i)

    % Stores the reactants of interest
    YOut(:,k) = X(:,rxCount)/OMEGA;
    tOut(k) = t(rxCount);

    % Clears out extra data
    Xnew = X(:,rxCount);
    tnew = tOut(k);
    k = k + 1; % Advances counter

    % Re-initialize the loop
    yInt=Xnew;
    rxCount=1;
    t(1) = tnew;
    [X(:,1),fixedflagX] = initialNumber(N);

end; %end of main loop
close(h) % *** DELETE to remove waitbar ***

clear X t; %Cleans out memory
```

```
% This will plot the result
figure;
plot(tOut,YOut(1,:),tOut,YOut(2,:),tOut(1:1000:end),11,tOut(1:1000:end),341);
xlim([0 totalTime])

%%%%%%%%%%%%%%%%%%%%%%%END OF MAIN PROGRAM%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [h_mu]=h_mu(X,N,M)
% This is where the reaction propensities are coded.
% The *stoichiometry* is coded in the next function

global am bp n w Kr OMEGA; %Rate parameters
% Reactant names, scaled as concentrations
r1=X(1)/OMEGA;r2=X(2)/OMEGA;

h_mu(1)=OMEGA*am*(1+(r2/Kr/2)/w*(2+(r2/Kr/2)))/(1+(r2/Kr/2))^2; % Synthesis of mRNA for r1
h_mu(2)=OMEGA*am*(1+(r1/Kr/2)/w*(2+(r1/Kr/2)))/(1+(r1/Kr/2))^2; % Synthesis of mRNA for r2
h_mu(3)=OMEGA*bp*r1;% Degradation of r1
h_mu(4)=OMEGA*bp*r2;% Degradation of r2
%-------------------------------------------------------------------------
function R_mu = defineReactions(N,M)
% This function codes the reaction stoichiometry

global b
R_mu = zeros(M,N);
%             r1  r2
R_mu(1,:) = [ b   0] ;  % 'b' r1s are made
R_mu(2,:) = [ 0   b] ;  % 'b' r2s are made
R_mu(3,:) = [ -1  0] ;  % one r1 is lost
R_mu(4,:) = [ 0  -1] ;  % one r2 is lost
%-------------------------------------------------------------------------
function [X,fixedFlag] = initialNumber(N)
% This function re-initializes the initial condition. You may also specify
% if you want to keep one population fixed (fixed=1)

global yInt;
X = yInt;  % r1 r2
fixedFlag = [0  0];
```