

# Integrated Genomic Circuits

---

## 11.1 Natural Gene Circuits

Discover how genes can form toggle switches.

Understand how computer models of genomic circuits can lead to discoveries about learning.

Read genomic circuit diagrams to understand cancer.

Evaluate the influence of genome organization on the whole system.

## 11.2 Synthetic Biology

Utilize design principles to construct synthetic toggle switches.

Apply engineering principles to measure our understanding of genomes.

Integrate stochastic behavior of proteins and gene regulation.



## LINKS

Adam Arkin  
Harley McAdams

In biology, investigators must balance the utility of creating models against the danger of believing their models accurately represent a living system. Models of biological processes are never perfect, but they can help us make new discoveries. In Section 11.1, we examine gene regulation from a different level of control. Rather than determining exactly which DNA sequences control each aspect of a gene's overall productivity (Chapter 10), we will study gene regulation at the level of protein production. How can one gene influence another? Can genes work together to toggle between two alternative outcomes? Can we model genomic circuits to gain insights into how cells work? To answer these questions, we will explore a series of case studies that have led the way in modeling genomic responses on a small scale. Once we understand these types of **integrated circuits** (multigene interactions), can we calculate their reliability and ask why we are diploids and why we have apparently redundant genes? To understand how we learn new information, we will integrate a series of small circuits into a larger network. From this complex integrated circuit, we hope to discover new properties that were not apparent when each circuit was studied in isolation. Similar principles have been applied to understand cancer. Ultimately, we want to understand how organisms function by understanding how proteins work, both alone and as a part of integrated circuits.

## 11.1 Natural Gene Circuits

We know our genes are regulated to be activated in some cells and repressed in others (Chapter 10). We also know that proteomes are dynamic, changing in response to environmental influences and aging (Chapter 8). How does a cell know when to alter a particular gene's transcription? Cells need a mechanism to switch from on to off and vice versa. Genes need to sense their intracellular environment and respond accordingly. However, we don't want our cells to change so rapidly that genes are turned on and off every second of every minute. It would be a disaster for our brain cells to sense a drop in glucose and respond by converting themselves into liver cells that can store sugar. Therefore, our genes have to be tolerant of some cellular variations. Furthermore, cells need to have alternative means for accomplishing vital functions. Our genomes must be prepared for circumstances that might block one circuit from performing its cellular role. For example, human cells normally consume oxygen to produce adenosine triphosphate (ATP). Aerobic ATP production is a good strategy until you are being chased by a bear; then it is good to have an alternative (anaerobic) means to produce enough ATP to continue running. Knowledge of natural genomic circuits allows us to calculate the reliability of each component in

the circuit, which can further our understanding of genomes as they are regulated in living cells.

### Can Genes Form Toggle Switches and Make Choices?

Let's look at one universal issue related to networks: **bistable toggle switches**. You know what a toggle switch is; it turns on your lights, computer, iPod, etc. A biological, bistable toggle switch will remain in one position (on or off) until the circuit determines the switch should be toggled to the other position. Bistable toggle switches are easy to understand in electrical engineering terms, but how can biological circuits determine when to flip a switch? In Chapter 10, we saw how transcription factors regulate whether a gene will be on or off, but what controls the transcription factors? And what controls the proteins that control them? Part of the answer is that an egg is not just an empty bag of water, but is filled (thanks to Mom) with many lipids, carbohydrates, nucleic acids (including mRNA ready for translation), and proteins (including transcription factors). Developmental biologists have discovered what causes an egg to enter mitosis and cytokinesis, and form a new organism. Nonembryonic cell division repeats itself according to some internal regulatory mechanism. Normally, our cells can control their cell-division toggle switch, but if they lose control of this switch, we develop cancer. How biological toggle switches exert control over gene expression is neither esoteric nor insignificant.

### How Do Toggle Switches Work?

There are two ways to start answering this question: Start with data and build a model, or start with a model using engineering principles and improve the model with experimental data. **Harley McAdams** (Stanford University School of Medicine) and **Adam Arkin** (Physical Biosciences Division of the Lawrence Berkeley National Laboratory) combined the best of both approaches in an elegant analysis of genetic toggle switches. The first issue they had to address was the concept of noise.

**Noise** in a regulatory system such as a toggle switch means that, unlike your computer, genetic switches have to deal with a degree of uncertainty. We know that gene activation occurs when transcription factors bind to cis-regulatory elements. When a cell undergoes mitosis and cytokinesis (eukaryotes) or cell division (bacteria), the first source of noise is introduced: will both daughter cells receive the same number of transcription factors? Of course, if cells were as wise as Solomon, the pool of transcription factors would be split right down the middle, 50:50. However, cells are not "wise," and to some extent the partitioning process during cell division is random, or **stochastic**. For example, if a cell had 50 copies of the Otx

transcription factor, 6% of the time a particular daughter cell might get 19 or fewer copies, while 6% of the time it might get at least 31 copies (Math Minute 11.1). That could have a profound effect on the subsequent regulation of *Endo16* expression.

Another component of genetic noise is the fact that few binding sites exist for each protein, and binding occurs at a slow rate. For example, Otx may be able to bind to only a few cis-regulatory elements in the entire genome, and it has to find these elements. Each cis-regulatory element must be found by a small number of DNA-binding proteins. The limited number of transcription factors and binding sites results in an increased range of times when all the transcription factors are in the right places for any given gene. Another example of slow reaction rate is that once the cis-regulatory element is fully occupied and ready to initiate transcription, the first RNA will be produced a variable amount of time later due to noise in the initiation of the transcription machinery. Transcription takes an average of several seconds to begin, but again this is an average, with a distribution of times both shorter and longer than the average.

## What Effect Do Noise and Stochastic Behavior Have on a Cell?

In prokaryotes and eukaryotes, proteins are produced in bursts of translation of varying durations and with varying outputs. Therefore, the total number of proteins produced from any gene is not the same each time, but rather an average with a **normal distribution** (see Math Minute 11.1). By producing proteins in bursts rather than at a constant rate, the cell provides proteins a higher probability of forming a quaternary structure (e.g., a dimer) that may be required for full function. Most students learn that “gene Y is activated and produces X proteins per minute,” but this summary statement is an oversimplification of a messy and mildly chaotic world inside each of your cells.

Genes are noisy, but what does this have to do with a genetic toggle switch? Everything. Let’s imagine two proteins that each bind to different but overlapping binding sites, and that these sites have competing roles. For example, look back at the *Endo16* cis-regulatory element in Figure 10.17 and find the Z and CG2 binding sites. Here is a small segment of DNA that can accommodate two different proteins, but

---

### Math Minute 11.1 How Are Stochastic Models Applied to Cellular Processes?

At first, it is hard to imagine that some cellular processes are random. But random doesn’t necessarily mean chaotic; it is just a way of saying that the outcome is not exactly the same every time the process is repeated. Even sophisticated machinery designed to manufacture thousands of identical automobile parts produces parts that are nearly the same, but not 100% identical. The field of probability theory provides stochastic models for random processes. We have already seen one example in Math Minute 8.1: a model for sampling from a finite population using the hypergeometric frequency function. Now we will explore two stochastic models for cellular processes.

#### The Binomial Model

If 50 molecules of Otx (see Chapter 10) are floating around inside a nucleus prior to cell division, it seems likely that the two daughter cells will not always inherit exactly 25 molecules each. In this situation, randomness captures the idea that if a large number of identical cells divided, the outcome (i.e., the number of molecules inherited by each daughter) would vary. Some outcomes would occur quite often, while others would be rare. The fraction of the time that each possible outcome occurs in the long run (i.e., in a large number of cells) is an estimate of the probability of that outcome.

The standard stochastic model for situations like the allocation of Otx molecules between two daughter cells uses the *binomial frequency function*. This model assumes that a particular “experiment” is repeated  $n$  times, where each repetition, or trial, is independent of all the others. In the example of Otx, a trial consists of determining which daughter cell gets a particular molecule. We assume that the fate of each molecule is independent of the other 49, a reasonable assumption if each molecule of Otx has randomly selected a location inside the nucleus. Since all 50 molecules must wind up in one of the two daughter cells, there will be 50 trials ( $n = 50$ ). Each trial results in a “success” with probability  $p$ . In this example, a trial is counted as a success if a particular daughter cell gets the Otx molecule in question. To keep track of how many

molecules go to each daughter, it is helpful to distinguish the cells by their relative positions after cell division: daughter L (cell on the left) and daughter R (cell on the right). Let's arbitrarily pick daughter L as the one we follow. In other words, the number of successes in 50 trials is the number of Otx molecules that go to daughter L. Since daughter L is just as likely to get each molecule as is daughter R,  $p = 0.5$ .

Under the binomial model, you can compute the probability of achieving  $k$  successes out of  $n$  independent trials with the binomial formula:

$$\binom{n}{k} p^k (1-p)^{n-k}$$

where  $\binom{n}{k}$  is the binomial coefficient defined in Math Minute 8.1. Therefore, the probability that daughter L receives 25 molecules of Otx is

$$\binom{50}{25} (0.5)^{25} (1-0.5)^{50-25} \approx 0.112$$

You can find the probability that daughter L receives 19 or fewer molecules (meaning daughter R receives 31 or more molecules) by computing the probability of each outcome satisfying this criterion (there are 20 such outcomes), and adding the 20 probabilities to get 0.06. Similarly, you can determine the probability that daughter L receives 31 or more molecules (meaning daughter R receives 19 or fewer molecules) to be 0.06. Thus, with probability 0.12, each daughter will be 6 or more molecules away from the average value of 25.

### The Normal Model

Many random factors influence the amount of protein produced by a gene at a particular time, including the number, location, and timing of all proteins needed to transcribe and translate the gene. In this situation, randomness means that if you measure the amount of protein produced by the same gene in thousands of identical cells (or in a single cell at thousands of time points), the outcome (i.e., number of protein molecules produced) will vary. Some outcomes will occur more frequently than others.

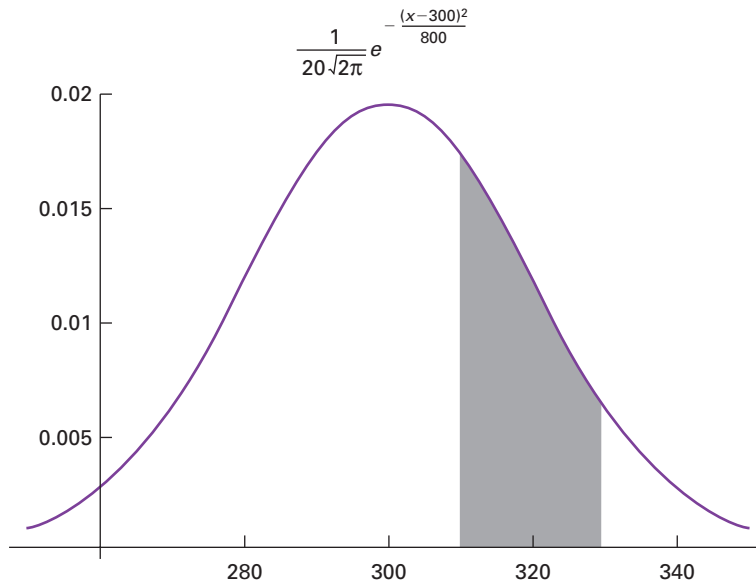
The standard stochastic model for a random quantity that represents the accumulation of many small random effects (e.g., protein production) is the normal distribution (also called the Gaussian distribution, or bell curve). The use of the normal distribution model is justified by one of the most powerful results in probability theory, the Central Limit Theorem.

Let  $X$  be the number of molecules of protein produced by a gene. You can compute the probability that the value of  $X$  is in a certain interval by finding the appropriate area under the curve given by the normal probability density function:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

In this function,  $\mu$  is the mean of the distribution (the average, or expected, value of  $X$ ) and  $\sigma$  is the standard deviation of the distribution (a measure of the variation in values of  $X$ ). The values of  $\mu$  and  $\sigma$  can be estimated by taking a random sample of measurements (i.e., measuring the quantity of protein produced at several randomly chosen times), and calculating the sample mean and sample standard deviation of these measurements.

For example, if  $\mu = 300$  and  $\sigma = 20$ , the probability that  $X$  is between 310 and 330 is given by the shaded area in Figure MM11.1. You can look up the numerical value of this area (approximately 0.2417) in a table of normal probabilities, or you can use numerical integration to estimate the area. In addition, many mathematical, statistical, and spreadsheet programs provide a function for computing probabilities using the normal probability density function.



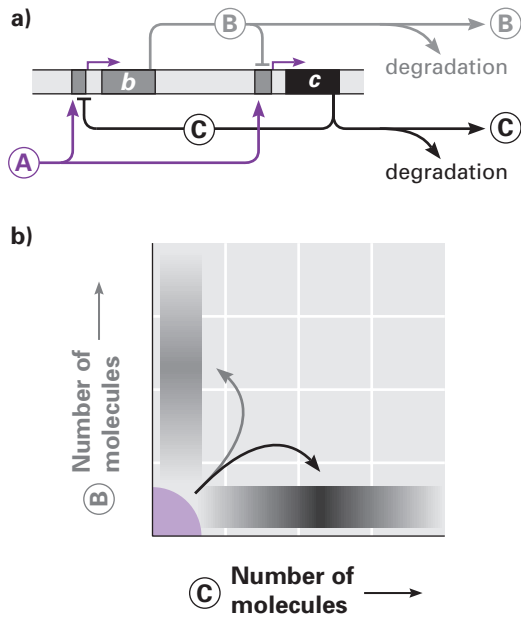
**Figure MM11.1** Normal probability density function with  $\mu = 300$  and  $\sigma = 20$ . The shaded area represents the probability that  $X$  is between 310 and 330.

A handy property of the normal probability distribution is that  $X$  is in the interval  $\mu \pm \sigma$  67% of the time;  $X$  is in the interval  $\mu \pm 2\sigma$  95% of the time; and  $X$  is in the interval  $\mu \pm 3\sigma$  99% of the time. For example, with  $\mu = 300$  and  $\sigma = 20$ , we know that  $X$  is between 260 and 340 ( $300 \pm 2 \times 20$ ) with probability 0.95. We used this property in Math Minute 8.2 to determine whether a particular node in a graph had an unusually large degree. Because the normal distribution is so often a reasonable approximation for random quantities, we can use this property any time we look at data with error bars to get a rough estimate of the probability that the measured quantity is within the interval denoted by the error bars.

Standard stochastic models are excellent starting points for understanding random cellular processes. However, these models rely on certain assumptions, which may or may not hold. Like all models, stochastic models can be refined after gathering experimental data.

only one at a time. Either  $Z$  can be occupied, or  $CG2$ , but not both. Both binding sites modify the output of module  $A$ ;  $Z$  is responsible for repressing and  $CG2$  for amplifying. Given the noise within the system, two genetically identical cells (descendants of the same fertilized sea urchin egg) may have exactly opposite developmental fates. Noise and stochastic genomic circuits help explain why even “identical” human twins have different fingerprints. As a result of noise, genetic toggle switches are affected by DNA-binding site competition and stochastic production of transcription factors. However, genetic toggle switches are too important to be determined by noise alone. Toggle switches need a feedback loop that reinforces what was initially a random “decision.”

Let’s understand Figure 11.1 (a theoretical switch) before we study a naturally occurring toggle switch. Protein  $A$  can bind to the cis-regulatory elements of genes  $b$  and  $c$  to initiate transcription for both genes. Protein  $B$  has three possible fates: it can be degraded by the cell; it can diffuse away and perform other functions; and, most importantly for us, it can repress the expression of gene  $c$ . Conversely, protein  $C$  has three fates, one of which is to repress gene  $b$ . Will protein  $A$  bind to  $b$  and indirectly suppress  $c$ ? Or will  $A$  bind to  $c$  and suppress  $b$ ? Either outcome is possible, because the determining factors are stochastic: the amount of  $A$  and its ability to find a limited number of binding sites upstream of  $b$  and  $c$ . Once the decision is made, a genetically identical population of cells can be split into two subtypes



**Figure 11.1** Toggle switch circuit.

**a)** Two promoters (small gray boxes) upstream of two genes (boxes with lowercase letters) are controlled by protein A (proteins are represented by capital letters inside circles). Protein B represses gene *c* and protein C represses gene *b*.  
**b)** The shading displays the number of genetically identical cells containing different numbers of molecules (B or C). Initially, cells contain A but neither B nor C (mass of cells at the origin). Later, more cells are expressing C, as indicated by the darker shading gradation along the X-axis. Within each shaded gradation, the number of B or C molecules varies around a mean value, because protein production is stochastic. Arrows indicate the choice made by cells to express either B or C.

(Figure 11.1b). An individual cell will produce only B or C. No cells will make both, nor are there any genetic hardwiring instructions that allow us to predict which path any particular cell will choose.

### DISCOVERY QUESTIONS

1. If two molecules of protein A were inside a single cell, would it be possible to produce equal amounts of proteins B and C in the same cell?
2. In Figure 11.1b, why did more cells produce protein C than protein B? Would you predict this same outcome if you repeated the experiment? Explain your answer.

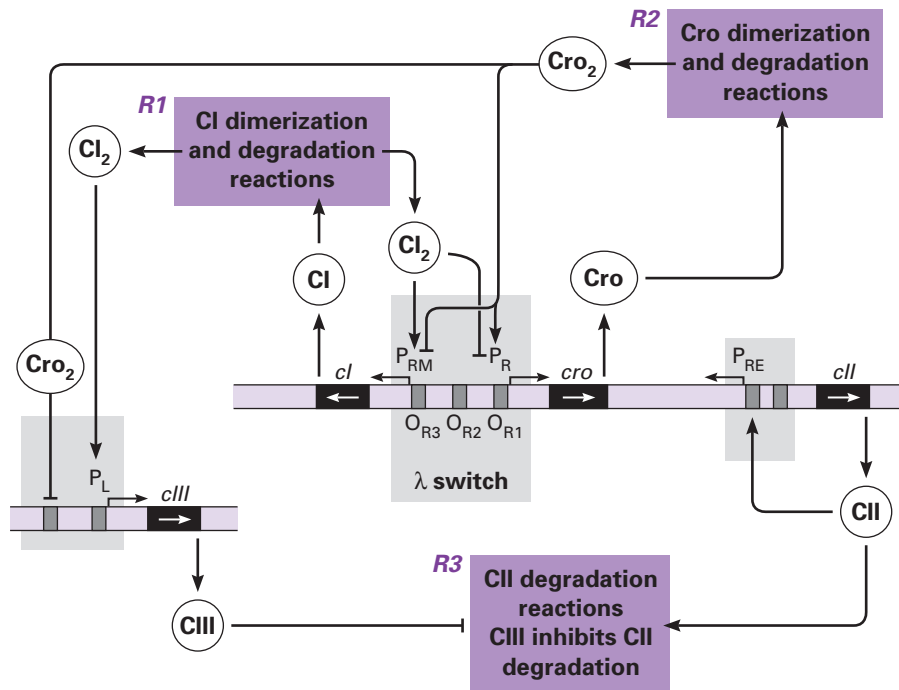
### Theory Is Nice, but Do Toggle Switches Really Exist?

Theoretical models help us comprehend general principles, but they are useful only if they approximate reality. Many pathogens evade our immune systems by changing their

protein exteriors on a regular basis. How can genetically identical pathogens present different exteriors? They take advantage of noise and toggle switches. As your immune system learns to search and destroy, the pathogen changes its appearance. For the pathogen, it is easy to see that there are evolutionary forces at work to maintain noise and toggle switches, but mechanistically, how does it work? This section discusses several naturally evolved circuits and the toggle switches in each one; Section 11.2 highlights some recent research into synthetic circuits constructed by investigators but tested inside cells. Both types of research help us measure our understanding of noise, toggle switches, and biological circuits.

Let's take a closer look at a naturally evolved toggle switch that controls the behavior of the bacterial virus called  $\lambda$  phage.  $\lambda$  has two behaviors from which to "choose." It can either live quietly within its *Escherichia coli* host (**lysogenic** lifestyle), or it can replicate rapidly and blow up its host as the progeny are launched to infect new hosts (**lytic** lifestyle). The choice between peaceful coexistence and lethal parasitism is made by a single protein with the inconspicuous name of CII (pronounced C two).

The  $\lambda$  phage toggle switch (Figure 11.2) and the theoretical switch in Figure 11.1 are very similar. CII is equivalent to protein A. The amount of CII is the critical parameter, and one of two outcomes is possible. If CII finds the promoter  $P_{RE}$ , transcription will proceed toward the left of  $P_{RE}$  and lead to the transcription of *cI* (pronounced C one) further downstream. CI can bind to the promoter  $P_L$  upstream of *cIII* and lead to the production of CIII. CIII prevents the destruction of CII; thus, CIII indirectly reinforces its own production in a positive feedback loop. Dimerized CI reinforces its own production indirectly by binding to sites labeled  $O_{R1}$  and  $O_{R2}$  to repress the production of Cro protein ( $CI_2$  acting as a repressor of *cro*).  $CI_2$  binding to  $O_{R1}$  and  $O_{R2}$  also promotes its own production in a positive feedback loop by acting as a transcription factor for its own gene, *cI*. Once CII initiates this bistable toggle switch,  $\lambda$  is locked into peaceful lysogenic coexistence with its host *E. coli* unless new environmental forces perturb the system (e.g., UV light, change in nutrient availability). However, the toggle switch could have flipped the other way, depending on the noise and stochastic protein behaviors. CII protein could have been degraded if it took too long to find  $P_{RE}$ , because *E. coli* makes a protease that can destroy CII. If **Brownian motion** (random motion driven by kinetic energy) causes the protease to find CII before CII finds  $P_{RE}$ , the lytic lifestyle is chosen. In the absence of CII, the promoter labeled  $P_R$  is weakly active and begins transcribing to the right, resulting in the production of Cro protein.  $Cro_2$  binds to  $O_{R3}$  and  $O_{R2}$ , which leads to repression of *cI* and increased transcription of *cro*. The positive feedback loop keeps the bistable toggle switch flipped toward *cro* transcription and a lytic lifestyle that eventually leads to the production of hundreds of fully mature viruses that swell and lyse the *E. coli* host cell.



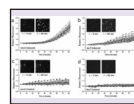
**Figure 11.2**  $\lambda$  toggle switch that chooses between coexistence and murder.

DNA (light purple bands) and promoters (light gray boxes with arrows pointing to their genes) from  $\lambda$  phage. Genes are black boxes with white arrows indicating the direction RNA polymerase travels to transcribe the genes. Genes are induced (black arrows) or repressed  $\perp$  as indicated. Three regulatory regions (purple boxes labeled R1, R2, and R3) determine the lifestyle “decision” for  $\lambda$  phage. Named circles are proteins, with subscript 2 indicating dimerization. Arrows into and out of regulatory regions represent a flow of information.

There are several noisy factors in the choice made by  $\lambda$  phage, such as the limited number of proteins and binding sites as well as the variable amount of time it takes to initiate transcription. Another factor is the burst of protein production. Notice in Figure 11.2 that both Cro and CI must form homodimers to be functional. Dimerization is more likely to happen when proteins are produced in bursts than when the same number of proteins is made at a slow but steady rate. A final component worth noting is that environmental influences can skew this decision. For example, if the bacterium host happens to be growing in a nutrient-rich environment (e.g., in a flask with lots of glucose), the bacterium produces more protease, resulting in faster destruction of CII and the production of many new  $\lambda$  phage (lytic lifestyle). Conversely, if the bacterium happened to be in a nutrient-poor environment (e.g., on the bottom of your shoe), there are fewer (but not zero) protease molecules, so CII has a higher probability of finding its binding site on  $P_{RE}$  before being destroyed. A longer half-life for CII leads to peaceful coexistence (lysogenic lifestyle), which makes good sense for the virus. Why should a virus reproduce rapidly if the environment is not conducive to making more potential hosts? Why not wait for the nutrients to arrive (e.g., when you step in something

yucky) so the bacteria can grow? When the nutrients arrive, bacteria will grow faster, proteases will be more numerous, CII will be destroyed more readily, more viruses will form, more bacteria will lyse, and viruses will infect more hosts. The selective advantage for a noise-tolerant toggle switch is impressive.

In recent studies, investigators have examined the amount of noise generated by different aspects of a genomic circuit. For example, graduate student Yina Kuang in David Walt’s Chemistry Department lab at Tufts University led a team that studied gene expression in single *E. coli* cells. The investigators placed two different promoters (*recA* and *lacZ*) upstream of the reporter gene *GFP* and then measured the production of fluorescence in 200 individual cells when induced or under control conditions (Figure 11.3). The *recA* promoter is **constitutively** on (always activated) at a low



**Figure 11.3** Monitoring *recA* and *lacZ* promoter activity in multiple individual cells.

Go to [www.GeneticsPlace.com](http://www.GeneticsPlace.com) to view this figure.


**DATA**  
*recA* movie

level, and there is considerable variation (noise) among the 200 different cells.

When induced, *recA* promoter stimulates large amounts of mRNA, as indicated by protein production, but the variation between cells is relatively low (see online *recA* movie for sample data). In contrast, *lacZ* exhibits a very low background level of transcription with little noise under control conditions, and induction does not produce as much increase over basal rate.

The behavior of *recA* and *lacZ* promoters might seem irrelevant until you consider the role each gene plays in a cell's life. RecAp is used to repair DNA damage. Cells need RecAp at all times, and thus cells tolerate a leaky and noisy *recA* promoter. When the cell senses DNA damage, the promoter requires only one step to switch to a higher expression rate with relatively less noise, because repairing DNA is a vital function that must be addressed before cell division can resume. In contrast, *lacZ*<sub>p</sub> metabolizes lactose, and the gene is induced in the absence of glucose and the presence of lactose (or experimentally applied IPTG). Basal expression of *lacZ* is normally low because alternative sugars would be available. The toggle switch for *lacZ* induction requires several other proteins, and each of those proteins has its own level of noise. Therefore, *lacZ* induction is noisy because each step in the induction process brings its own level of noise to the combined process of *lacZ* transcription. It appears that the amount of noise in a toggle switch is related to each gene's function. These findings indicate noise may be more than just tolerated; rather, it appears to be a phenotype subject to selection pressure. Cells appear to benefit from some promoters with loose regulation, while others provide greater fitness when their transcription is very tightly regulated.

**DISCOVERY QUESTIONS**

3. What would be the consequences if CI degradation were more prevalent than CI dimerization? How does Cro<sub>2</sub> affect the ability of CII to switch  $\lambda$  from lytic to lysogenic?
4. If the P<sub>L</sub> promoter were inactivated, would this change the outcome of the toggle switch for lysogenic vs. lytic lifestyles? Explain your answer.
5. Which of the three regulatory regions (purple boxes) in Figure 11.2 would be subjected to the most noise? Hypothesize why tolerance of noise in this area of the  $\lambda$  life cycle may be advantageous.

### How Can Multicellular Organisms Develop with Noisy Circuits?

The preceding examples may lead you to believe many genetic toggle switches are loaded with noise and impossible to coordinate—the genomic equivalent of herding cats.

But we know from our own experiences that life is not completely chaotic. You do not have brain cells trying to become liver cells. Every human went through gastrulation at the exact same time during gestation. How can cell populations with stochastic toggle switches work collectively toward a common goal? A team analogy may be useful, because coordinating genes in cells is similar to coordinating 11 football players on the field. Picture the offense with a quarterback (QB) who throws the ball, linemen who block defenders, and receivers who run downfield hoping to catch the ball, save the game, become heroes, etc. There are three keys to winning a football game, just as there are three keys to coordinating cell populations with noisy toggle switches.

1. Each player does not have to ensure that all the other players are in the right place. The QB and the two behind him can survey all other players and yell reminders to those who have lined up in the wrong place. This is called **cooperation through communication**.
2. At various times, the QB can consult a list of points to make sure everyone has made the right move. Watch how the QB will shout and sometimes raise and lower one leg to signal others to move a bit to the left or right. And what happens if everyone is confused? The QB can call a time-out to give the players a chance to get coordinated again. Each of these points prior to starting the play is called a **checkpoint**.
3. Any team that really wants to win has a contingency plan. Bill Cosby has a great comedy routine in which he relives a childhood football game where everyone is given very complex directions on where to go so the QB can throw the ball to someone. On real teams, the QB has two to four players running around, so if one is not a good target, the QB can look for other options. This duplication of options to accomplish a goal (winning the game) is beneficial **redundancy**.

Cells can use the same three keys to achieve coordination.

1. A subset of cells can secrete a product that will communicate a message to keep all cells synchronized.
2. Cellular proteins establish quality control at various checkpoints, such as DNA replication and the stages of mitosis. Checkpoints ensure the quality of the eventual outcome, but the exact timing for any given cell can vary due to noise and stochastic gene induction or protein function (e.g., regulation of *recA* and *lacZ* promoters).
3. Cells have redundant circuits to create fail-safe approaches to vital processes such as response to environmental signals. Some pathways have multiple ways of becoming activated and/or multiple ways of producing a cellular response. Redundancy also can be achieved by having isozymes that can perform

essentially the same job, though they may have slightly different tolerances to environmental perturbations.

### Redundancy: Does Gene Duplication Really Increase Genome Reliability?

Measuring reliability is essentially an engineering question. Is it really necessary to have more than one way to stop your parked car from rolling down a hill? In this context, the answer seems obvious, but genomic redundancy is less intuitive. For many years, it has been argued that having more than one locus encoding a particular function imparts a selective advantage. The second copy might be a backup in case one gene is mutated and loses its function. The duplicated gene can mutate over time and produce new functions for the cell. Freshly evolved duplications can provide a wider range of tolerance to environmental conditions, to ensure the common function is accomplished (e.g., one may work better in cold temperatures and the other in hot). Biologists should not reinvent the wheel to measure the value of redundancy. Why not borrow from well-established engineering methods for reliability analysis to assess the likelihood that a particular function will be performed successfully (Figure 11.4)?

In equation (a) and Figure 11.4, we are interested in producing protein B. The reliability of any given protein being produced is arbitrarily set at 0.9 or 90% and is represented by the line segment with the gene labeled  $x$ . In this reliability analysis, we follow what happens when the DNA required to produce B is altered in different ways.

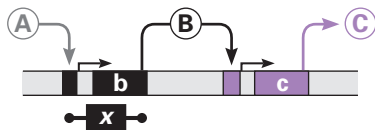
a. Reliability (R) = P

$$\begin{aligned} &= \text{probability of B being produced} \\ &= 0.9 \text{ or } 9,000 \text{ out of } 10,000 \text{ success rate.} \end{aligned}$$

If production of B requires two genes (Figure 11.5; equation b), the reliability of this step drops from 0.9 to 0.81, because both  $x$  and  $y$  have to be functional and the rule of multiplication applies.

b.  $x$  and  $y$  must work:

$$\begin{aligned} R &= P^2 \\ &= 0.9 \times 0.9 \\ &= 0.81 \text{ or } 8,100 \text{ out of } 10,000 \text{ success rate.} \end{aligned}$$



**Figure 11.4** A three-gene pathway for the production of protein C.

The genetic unit represented by the line segment marked “ $x$ ” indicates one or more genes that accomplish the task of producing B. In Figures 11.5–11.9, more than one gene/allele will be included in line segment  $x$ , but these are assumed to be unlinked.

If  $x$  were duplicated in the absence of  $y$  (Figure 11.6; equations c and d), the reliability of producing B substantially increases to 0.99. To determine this reliability, first we must calculate the probability of failure (Q) by taking the probability of either failure or success (total of 1) minus the probability of success (P). For a single gene  $x$  in a haploid genome (see Figure 11.4) the probability of failure is:

c. 
$$\begin{aligned} Q &= 1 - P \\ &= 1 - 0.9 \\ &= 0.1. \end{aligned}$$

For diploids (equation d), reliability equals all possible outcomes (1) minus the probability of failure ( $Q^2$ ;  $Q \times Q$ ) because both the upper  $x$  and the lower  $x$  have to fail (multiplication rule again) for the production of B to be unsuccessful.

d. 
$$\begin{aligned} R &= 1 - Q^2 \\ &= 1 - (0.1 \times 0.1) \\ &= 0.99 \text{ or } 9,900 \text{ out of } 10,000 \text{ success rate.} \end{aligned}$$

By evolving a diploid genome, we have increased reliability from 90% to 99%. This increase in reliability would also be true for haploids that duplicated a single locus. What is the reliability consequence if a diploid organism duplicates a locus (Figure 11.7; equation e)? Using the multiplication rule, the probability of failure for each gene ( $x$  and  $x'$ ) is multiplied, which produces another substantial increase in reliability.



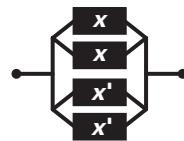
**Figure 11.5** A two-gene model to produce B.

These two genes are from unlinked loci in a haploid, though the genes have been diagrammed as adjacent for simplicity.



**Figure 11.6** A diploid model to produce B.

The two alleles are on homologous chromosomes.



**Figure 11.7** Diploid genome with a duplicated gene to produce B.

The two genes ( $x$  and  $x'$ ) are unlinked; alleles for a given gene are located on homologous chromosomes.



## LINKS

Ravi Iyengar  
Upinder Bhalla

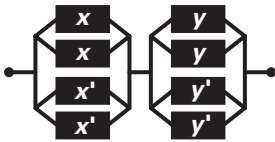
$$\begin{aligned}
 \text{e. } R &= 1 - (Q^2 \times Q^2) \\
 &= 1 - Q^4 \\
 &= 1 - (0.1^4) \\
 &= 1 - 0.0001 \\
 &= 0.9999 \text{ or } 9,999 \text{ out of } 10,000 \text{ success rate.}
 \end{aligned}$$

We calculated that two different genes in a haploid genome were less reliable (equation b) than one gene (equation a). This makes sense because with two genes there are two ways to fail instead of just one. Redundancy should help ameliorate the weakness of two genes. Let's determine the reliability in diploids with duplicated genes when two genes are required to complete the function (Figure 11.8; equation f). To calculate this reliability, we need to combine equations b and e.

$$\begin{aligned}
 \text{f. } R &= (1 - Q^4)^2 \\
 &= (1 - 0.0001)^2 \\
 &= (0.9999)^2 \\
 &= (0.9998) \text{ or } 9,998 \text{ out of } 10,000 \text{ success rate.}
 \end{aligned}$$

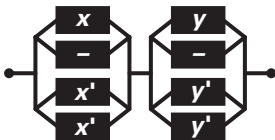
Note that the reliability for two genes that have been duplicated in a diploid (equation f and Figure 11.8) was not quite as high as for one duplicated gene in a diploid (equation e and Figure 11.7). For half the number of alleles (4 instead of 8), the organism has a slightly higher reliability; if one allele is mutated, the function will still be completed with 3 remaining alleles. Even if the individual in Figure 11.8 carries a nonfunctional  $x$  allele and a nonfunctional  $y$  allele, the redundancy of the system maintains a high degree of reliability (Figure 11.9; equation g).

$$\begin{aligned}
 \text{g. } R &= \text{modified equation f to take into account that} \\
 &\quad \text{only three viable alleles exist for each locus} \\
 &= (1 - Q^3)^2 \\
 &= (1 - 0.001)^2 \\
 &= (0.999)^2 \\
 &= (0.998) \text{ or } 9,980 \text{ out of } 10,000 \text{ success rate.}
 \end{aligned}$$



**Figure 11.8** Diploid genome requiring two genes ( $x$  and  $y$ ) to produce B in which both genes have been duplicated ( $x'$  and  $y'$ ).

None of the genes ( $x$ ,  $x'$ ,  $y$ , and  $y'$ ) are linked, though the pair of alleles for each locus is located on homologous chromosomes.



**Figure 11.9** Mutant alleles affect the genome's reliability.

Genome from Figure 11.8, but one  $x$  allele and one  $y$  allele are nonfunctional, as indicated by “-”.

## DISCOVERY QUESTIONS

- Does the reliability in Figure 11.9 surpass the reliability in Figure 11.7 if you assume one allele of the four in Figure 11.7 is nonfunctional? Explain your answer and support it mathematically.
- Based on reliability calculations, would a tetraploid be more or less reliable? If tetraploids are more reliable, why aren't more organisms tetraploid?
- If you were designing a metabolic pathway to be as reliable as possible, would you design:
  - fewer components that could multitask (each one performing multiple roles), or
  - more components, each with a specialized function? Explain your answer.

Every time we model a biological circuit, we are trying to create a simple version of a complex system. From the lessons of simple circuits, is it possible to understand complex circuits? If we can understand complex circuits, will we discover new (emergent) properties that were not present in the dissected circuits? For the remaining two cases in Section 11.1, we will study complex genomic circuits that might reveal emergent properties that are undetectable when genes are studied one at a time. How do we convert environmental stimulation into memories? Is it possible to understand cancer formation by studying protein circuits? It is worth noting that the following two examples were investigated by mining data available in public databases and the literature. The investigators combined experimental data with *in silico* research to form new models and make predictions to refine the models with increasing accuracy.

## Does Memory Formation Require Toggle Switches?

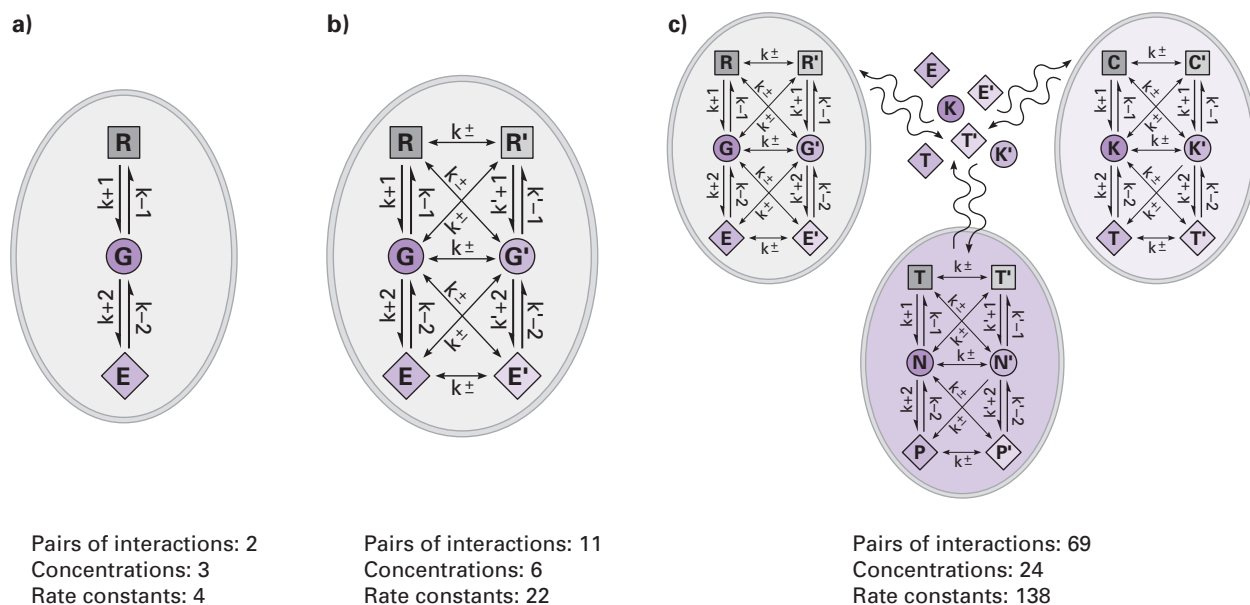
You may be familiar with the saying “easier said than done.” In other words, to say that there is a way to coordinate all these genomic circuits sounds simple and is intuitively appealing—but where is the proof? By now, you should have an appreciation for the complexity of the problems ahead. Only with the benefit of genomic data analysis—sequences, variations, expression profiles, proteomics, biochemistry, computer science, mathematics, etc.—can we begin to piece together the necessary information to see coherent patterns and circuits. **Upinder Bhalla** (the National Center for Biological Sciences, Bangalore, India) and **Ravi Iyengar** (Mount Sinai School of Medicine) analyzed many years' worth of data, made some insightful discoveries, and set the pace for others to follow. They used an engineering approach to understand four cell-signaling circuits and discovered some interesting emergent properties. They used data in the public domain to create a complex computer model that accurately simulates the neurocircuitry necessary for learning.

As we have seen before, to comprehend complexity, we need to simplify. That sounds like an oxymoron, but in fact, we do this all the time. No one says “compact disc read-only memory”; we just say CD-ROM. The five-letter acronym encapsulates a lot of information and facilitates comprehension. Bhalla and Iyengar decided to make a few simplifying assumptions first, rather than beginning with the most complex model possible. This simplification employs the principle known as **Occam’s Razor**: start with the simplest possible explanation first, rather than more complex ones. As a model is compiled and analyzed, the need for specific experiments becomes apparent and the model may become more complex if necessary.

It is interesting to note how certain pathways are more popular than others. For example, every introductory biology textbook discusses how an adrenaline rush can stimulate the production of cyclic adenosine monophosphate (cAMP). Bhalla and Iyengar decided to model a different aspect of cAMP signaling and made a couple of simplifying assumptions. They assumed that the cytoplasm was a well-stirred bag of liquid in which all components have equal access to each other. Uniform distribution within the cytoplasm requires each component to have a mechanism for delivering its message only to the correct target molecule and only in one direction. If this were not the case, it would be like having uninsulated wires in your iPod. The electrical currents would be unorganized and multidirectional, and you would never hear any music. To generate a

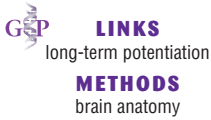
computer model of their uniformly mixed cell, the investigators needed to know the reaction rates of every enzyme and the concentration of every component in the four signaling circuits. However, easier said than done . . . (Figure 11.10). The level of complexity in genomic circuits can grow to staggering proportions. A 3-molecule system requires only 7 measurements, but an 18-molecule system requires 162 measurements!

It became too difficult to use standard computational approaches for this level of complexity, so Bhalla and Iyengar utilized a neural network simulation program, called **GENESIS**, to analyze the four interacting pathways. Even with the help of very sophisticated computation, two simplifying assumptions were made that do not reflect reality. First, the investigators ignored **compartmentalization**. For example, some components may be embedded in phospholipid bilayers and thus not freely available to all other components. In fact, we know this is true of many components included in the analysis, but it was impossible to quantify this aspect. Second, they ignored **regional organization of components**. Real cells cluster some components near each other to significantly increase biochemical efficiency, rather than letting them drift by Brownian motion. The mitochondrion is an excellent example; metabolism is more efficient because of the clustering of molecules used in the electron transport pathways. Microorganisms also use gene clustering for the production of antibiotics.



**Figure 11.10** Increased need for information as circuits become more complex.

**a) to c)**  $k+x$  represents the rate constant for the forward reaction and  $k-x$  the rate constant for the reverse reaction; 1 and 2 refer to pathways 1 and 2. In this simplified model, each component in a pathway can communicate only with its nearest neighbors. The effect of this simplifying assumption is most apparent in panel c). The symbols used are: R = receptor; G = G protein; E = effector; T = transcription factor; N = nucleic acids; P = proteins in the nucleus.



## Are Simple Models of Complex Circuits Worthwhile?

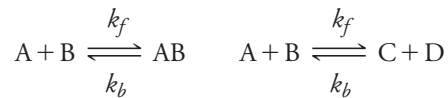
Most biologists assume that interconnected circuits work synergistically. The existence of synergy is why many in field biology dislike the reductionist approach used by molecular biologists. However, the reductionists' goal is not merely to disassemble a cell to look at the parts, but to understand the parts well enough to explain synergistic interactions and move beyond descriptions toward testable predictions. The particular case under investigation was one that philosophers and biologists have pondered for centuries: How do we learn?

Neurobiologists have given us great insights into the mechanisms utilized whenever an animal (e.g., flies or humans) learns something new. Intuitively, we know our neurons must undergo some sort of change in order for us to retain information. There is a genetic component to memory formation, so we know proteins are involved. Somehow, proteins must alter what a neuron does—become depolarized when stimulated and release neurotransmitters to relay this information. Sounds simple, right? A neuron's change in function is called **long-term potentiation (LTP)**, which means the consequence of neuronal stimulation is maintained after the original stimulus is gone. To see a simple example of this, stare at a bright light and then turn away. Even though the light is no longer hitting your neurons, you still “see” it. Your neuron's function was changed so that it performed differently after the stimulus was removed. Bhalla and Iyengar decided to model the complex circuitry of the mammalian brain.

Before we begin dissecting, we need to learn a little **brain anatomy** to provide the context for learning. Where in the brain does learning taking place? Deep within your cerebrum is a collection of neurons called the **hippocampus**. For at least 30 years, memory research has focused on the hippocampus as the center of learning/memory. Inside the hippocampus are layers of neurons, and each layer has a name. We will focus on the CA1 layer. On the cell body and dendrite of CA1 neurons are bumps called spines. Embedded in these spines are integral membrane proteins, three of which are of particular interest to us. The neurotransmitter glutamate binds to its receptor, called mGluR (*mouse glutamate receptor*). As with all receptors, it facilitates **signal transduction**; that is, it transmits the extracellular signal across the plasma membrane. NMDAR (*N-methyl-D-aspartate receptor*) is a voltage-sensitive calcium ion channel. The final plasma membrane component is AMPAR ( $\alpha$ -amino-3-hydroxy-5-methyl-4-isoxazolepropionate receptor), another glutamate receptor that acts as an ion channel when stimulated by glutamate. LTP is initiated when mGluR and NMDAR are stimulated by a certain amount and frequency of stimuli. Experimentally, LTP can be induced in mouse neurons when stimulated with 3 mild electrical inputs of 100 Hz pulses, 1 second each and separated by 10 minutes. Bhalla and Iyengar set out to construct a computer model of the complex series of events from stimulation to LTP.

## How Much Math Is Required to Model Memory?

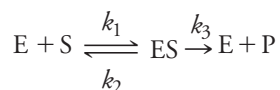
Molecular biology is becoming mature enough to need the assistance of many other disciplines, especially mathematics. However, the math used in this study is quite simple. To start their *in silico* research, Bhalla and Iyengar considered only two types of connections: protein-protein interactions and **second messengers**. Two additional facts they considered were: (1) proteins **degrade**, that is, they are destroyed by the cell over time; and (2) enzymes have reaction rates—for example, there is an average amount of time it takes a kinase to consume ATP and add a phosphate onto its substrate. In the first reaction below, A and B join to form AB. This illustrates how two proteins, such as a kinase and its protein substrate, can bind to each other. The binding has a forward reaction rate ( $k_f$ ) and a backward reaction rate ( $k_b$ ). The second reaction shows the conversion of A and B into C and D. For example, we could measure the amount of  $\text{Na}^+$  inside a cell (A) and the amount of  $\text{K}^+$  outside a cell (B) as they are changed into  $\text{Na}^+$  outside a cell (C) and  $\text{K}^+$  inside a cell (D). The forward rate of this conversion ( $k_f$ ) is the rate of the Na/K pump, and the backward rate ( $k_b$ ) is the rate of ions passing through ion channels. This second reaction can be written as an equation, which says that the change ( $d$ ) in the concentration of A ( $[A]$ ) over a change in time ( $dt$ ) is equal to the production of A (the backward rate times the concentrations of C and D, which is written:  $k_b[C][D]$ ) minus the amount of A lost (due to the forward reaction that consumes A, which is written:  $k_f[A][B]$ ). You have studied these types of interactions and rates before, so this level of circuitry should be comprehensible so far. The math is only multiplication, division, and subtraction. To determine LTP, all we need is numbers to replace the variables.



$$d[A]/dt = k_b[C][D] - k_f[A][B]$$

One more enzymatic interaction is needed to analyze the learning circuit. The next equation states that an enzyme (E) binds to its substrate (S) to form a complex of the two (ES). This step can go forward ( $k_1$ ) or backward ( $k_2$ ), meaning the reaction is reversible. However, the second step is irreversible, as indicated by the forward arrow and only one rate constant ( $k_3$ ). Therefore, the ES complex can be converted into the original enzyme (E; an enzyme is never consumed in a reaction) plus a new product (P). Each step occurs at a measurable rate, and these values are called **rate constants** (the  $k$  values). For example, the enzyme adenylyl cyclase produces cAMP from the substrate ATP. Adenylyl cyclase (E) binds to ATP (S) and forms (at rate  $k_1$ ) a complex of the two (ES). ATP and adenylyl cyclase can fall

apart ( $k_2$ ) or proceed ( $k_3$ ) toward an irreversible production of adenylyl cyclase plus cAMP (P).



That's all the math and chemistry you need to understand to grasp this very sophisticated *in silico* analysis of learning! We still have one problem, though. We don't have any real numbers to put into these equations. Luckily, biochemists and cell biologists have published all the values needed to initiate a GENESIS software analysis of a nerve's ability to learn.

### How Do You Build Complex Models?

Fifteen well-studied circuits were needed to produce Bhalla and Iyengar's LTP simulation (Figure 11.11). They started by building computer models of each of the 15 circuits individually. At each step, they made sure that the models matched experimental data. This alone was an impressive task, but here is where the fun starts. They began to build integrated circuits by gradually combining each of the 15 individual circuits. Initially, they only allowed two types of connections between circuits: (1) secondary messengers arachidonic acid (AA) and diacylglycerol (DAG), and (2) an enzyme in one circuit bound to its substrate produced in another circuit.

Permitting these two types of interactions allowed the investigators to combine the circuits labeled a, b, e, f, h, k, and l in Figure 11.11 into one integrated circuit (Figure 11.12). Notice that epidermal growth factor (EGF) leads to the activation of two enzymes: mitogen-associated protein kinase (MAPK) (in circuit a) and PLC $_{\gamma}$  (in circuit f). MAPK has been studied intensively for many years. It is associated with substances that stimulate mitosis, and it phosphorylates proteins. Loss of control of MAPK can lead to the formation of cancerous cells. PLC $_{\gamma}$  is one isoform, or version, of phospholipase-C (there are several PLC genes in each person, and this version, or isoform, is called  $\gamma$ ) that cuts the phospholipid called phosphatidyl inositol bisphosphate (PIP $_2$ ) into inositol trisphosphate (IP $_3$ ) and diacylglycerol (DAG). By connecting these seven circuits, a new layer of complexity became evident—feedback loops. A **feedback loop** occurs when the product has a stimulatory or inhibitory effect on one of the upstream components, such as an enzyme that leads to product formation.

### Can a Transient Stimulus Produce Persistent Kinase Activation?

Do individual circuits behave synergistically when they are integrated into a larger pathway? Can a computer model accurately simulate LTP? Can we use this model to make predictions that lead to improved understanding of how we learn new information? We know that these individual components and circuits are involved in learning, and we

know that LTP is the result of long-term activation of a kinase. But is it really possible to simulate something as complex as learning on a few megabytes of computer microcircuits?

How did the computer model compare with real data? In Figure 11.13a, you can see the simulation approximated the experimental data. In this graph, the MAPK activity was plotted as a function of time ( $d[A]/dt$ ). The activation of MAPK was transient due to the normal degradation mechanisms in the cell. Equally impressive is Figure 11.13b, comparing simulated PLC $_{\gamma}$  activity (dashed lines) with experimental data (solid lines) in the presence (triangles) or absence (squares) of EGF over a 10,000-fold range of calcium concentrations. Given the close agreement between experimental data and the computer model, these two investigators had a good foundation on which to build more complex integrated circuits.

In Figure 11.12, you can see two areas that provide feedback: PKC and MAPK. PKC activates Raf, which activates another kinase MEK, which activates MAPK (two isoforms of MAPK were used in this simulation, numbers 1 and 2). MAPK activates cytosolic phospholipase A-2 (PLA2), to produce AA, which is half of the stimulus needed to activate PKC. What concentration and duration of stimulus are needed to produce a long-lasting activation of PKC and MAPK? Figure 11.14 is data intensive and worth the effort required to dissect it carefully. First, let's look at the PKC data (open symbols) in the figure. Neither 10 minutes at 5 nM EGF (open circle) nor 100 minutes at 2 nM EGF (open square) led to any significant activation of PKC. However, 100 minutes at 5 nM EGF (open triangle) was sufficient to lead to a protracted activation of PKC. Note how PKC was activated during the 100 minutes, and when the stimulation was removed, PKC activity remained elevated. A long-term change in function after removal of stimulus is a hallmark of LTP. MAPK activity (closed symbols) was a bit more complicated, but the final outcome was similar. Stimulation for 10 minutes at 5 nM produced transient activation, while 100 minutes at 2 nM produced a more prolonged activation, but at a lower amplitude, and the activation did not extend much beyond the 100 minutes of stimulation. However, 100 minutes of 5 nM stimulation produces a tenfold increase in activity (from 0.01 to 0.1  $\mu$ M) as well as a prolonged activation (i.e., a bistable toggle switch). Notice how MAPK was activated during the 100 minutes and what happened at about 50 minutes: it jumped to a higher level. Something extraordinary (synergistic) was happening.

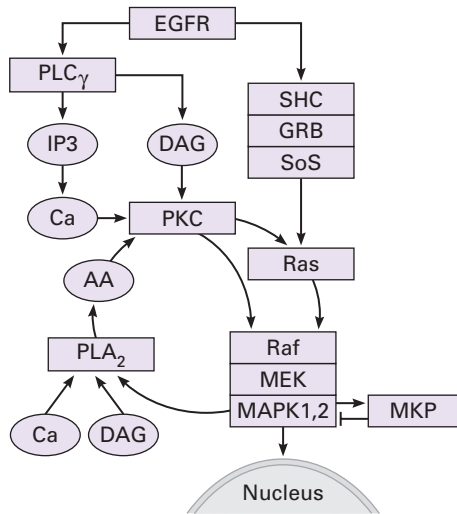
**LINKS**  
15 circuits



### DISCOVERY QUESTION

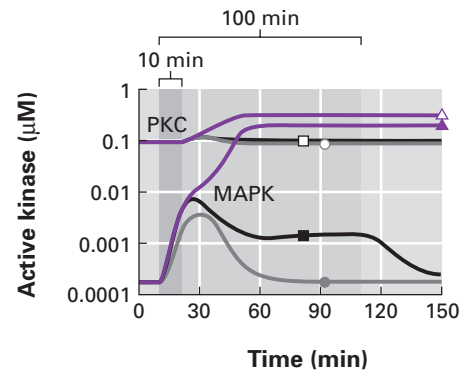
- Look at Figure 11.14 and hypothesize what may be the cause of the tenfold increase in MAPK activity between 30 and 50 minutes for the 100 minutes at 5 nM stimulation.





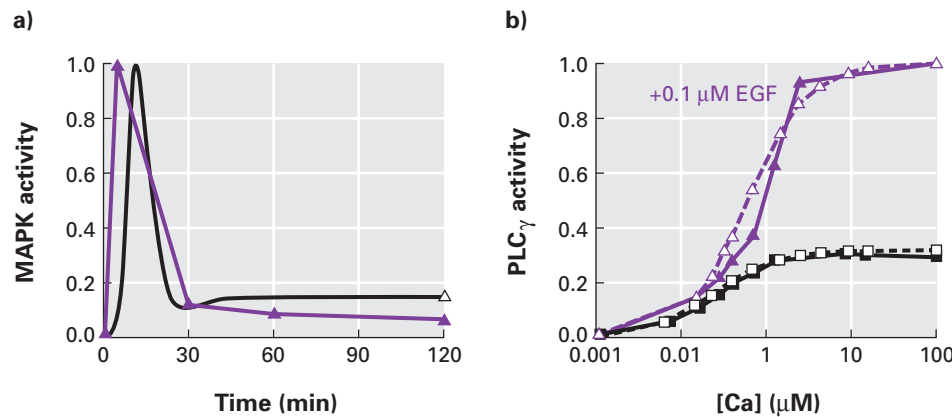
**Figure 11.12** Circuit diagram of signaling pathway beginning with EGFR and ending with new gene activation inside the nucleus.

Rectangles represent enzymes, and circles represent messenger molecules. This integrated circuit utilized pathways a, b, e, f, h, k, and l from Figure 11.11.



**Figure 11.14** Activation of the feedback loop.

PKC (open symbols) and MAPK (closed symbols) activities were graphed to show the effect of a positive feedback loop. Three stimulus conditions are represented: 10 min at 5 nM EGF (circles), 100 min at 2 nM EGF (squares), and 100 min at 5 nM EGF (triangles). Dark and light gray shading in the graph represents the 10 and 100 minutes of EGF exposure, respectively.



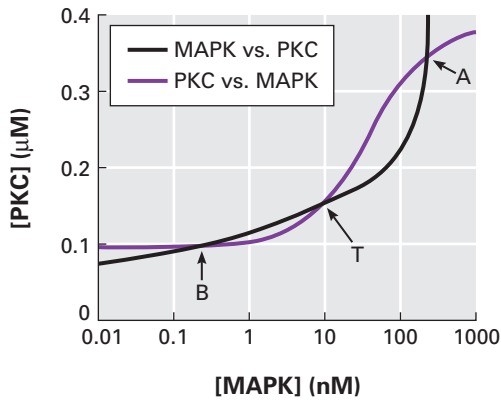
**Figure 11.13** Computer model matches experimental data.

**a)** Measuring MAPK activity as a function of time. Simulation (open triangle) and real data (filled triangle) are very similar. The stimulus in both cases was a steady supply of 100 nM EGF. **b)** Measuring PLC<sub>γ</sub> activity as a function of calcium concentration, where dashed lines represent computer simulation and solid lines represent experimental data. Triangles indicate the presence of 100 nM EGF; squares indicate the absence of added EGF.

(purple) and MAPK vs. PKC (black) intersect at three points: A, B, and T. Point A represents high activity for both PKC and MAPK, whereas point B represents basal activity for both. A and B represent distinct steady-state levels of PKC and MAPK activities. A system with two distinct steady states is a **bistable circuit**. The bifurcation point T is important because it defines the **threshold stimulation**, which can be thought of as the stimulus to flip a toggle switch. If the initial stimulation of EGF (amplitude and duration) is sufficient to activate either PKC or MAPK

above T, then both enzymes would switch to steady state at point A. In contrast, if the initial stimulation is below T for both PKC and MAPK, then upon removal of the stimulus, both enzymes will relax to B, the basal level of activity.

We have returned to the concept of a bistable toggle switch, but this particular switch is produced by many more components than the simpler switch we studied earlier. In  $\lambda$  phage, the choice between lytic and lysogenic lifestyles was determined by the amount of CII protein present (see Figure 11.2). For EGF stimulation of simulated



**Figure 11.15** Bistability plot for feedback loop in Figure 11.12.

The PKC vs. MAPK activities plot (purple line) was constructed by holding the level of active MAPK constant, running the simulation until steady state, and reading the value for active PKC. This process was continued for a series of MAPK levels spanning the range of interest. A similar process was repeated for MAPK vs. PKC activities plot (black line) by holding the level of active PKC constant and calculating MAPK activity. Both plots were drawn with the concentrations of PKC on the Y-axis and MAPK on the X-axis. The curves intersect at three points: B (basal), T (threshold), and A (active). A and B are stable points, but T is the toggle switch point for the two steady-state “choices” of A and B.

LTP, a critical level of kinase activity in a positive feedback loop determines whether the stimulus leads to LTP (i.e., learning) or not.

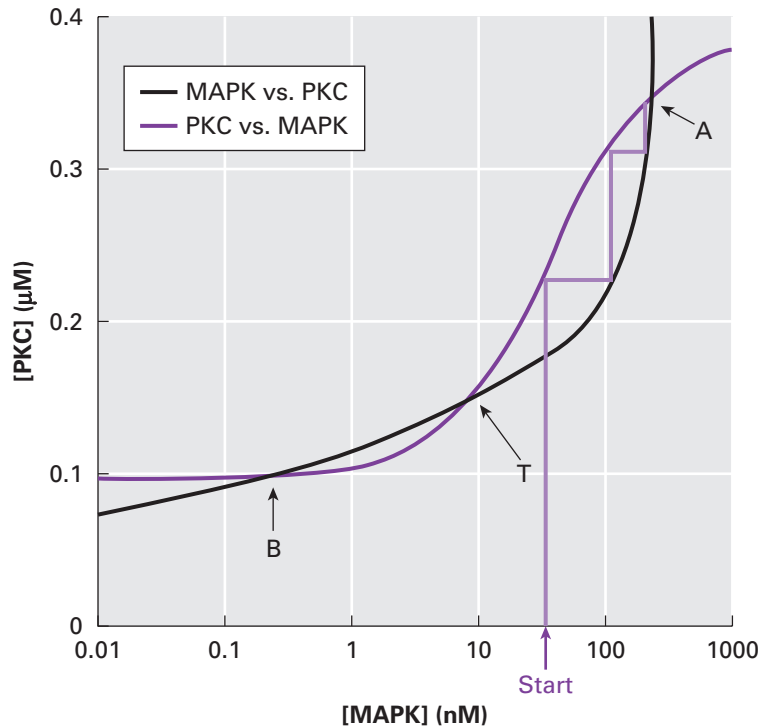
Bhalla and Iyengar made an interesting analogy that was very appropriate given that they were simulating learning. They reminded us that a bistable (i.e., digital) system can store information the same way that RAM stores information on a computer. It is also worth noting that the simulated enzymatic bistable system was very reliable, due to the redundant mechanisms for achieving steady-state level A. High activity of either PKC or MAPK was sufficient to flip the system from off (steady-state B) to on (steady-state A). Earlier, we described biological systems in engineering terms, including “redundant,” “reliable,” and “fail-safe.” Another engineering term, **robust**, means the circuit tolerates a wide range of environmental conditions. According to data not shown here, when the activities of five different enzyme concentrations were varied (PKC, MAPK, Raf, PLA2, and MAPKK, which phosphorylates MAPK to activate it), the simulated bistable circuit remained functional. An interesting consequence of this analysis was that most of the permutations were equally effective in achieving steady-state A, but PKC was the least tolerant to change, indicating it was the least robust enzyme of the five tested. The investigators hypothesized that PKC’s lack of robustness was due to a limited number of PKC isoforms in their computer simulation (i.e., too little redundancy).

### Math Minute 11.2 Is It Possible to Predict Steady-State Behavior?

Figure 11.15 depicts the steady-state MAPK concentration when [PKC] is held constant (black line), and the steady-state PKC concentration when [MAPK] is held constant (purple line). If we want to know the response of MAPK to a particular concentration of PKC, we begin at that value on the PKC axis, move horizontally until we hit the black line, move vertically until we hit the MAPK axis, and read the concentration at the point where we hit the MAPK axis. Conversely, if we want to know the response of PKC to a particular concentration of MAPK, we begin at that value on the MAPK axis, move vertically until we hit the purple line, move horizontally until we hit the PKC axis, and read the concentration at the point where we hit the PKC axis. Note that A, T, and B are stable points; if [PKC] or [MAPK] is at one of these three points, the response of the other kinase is at that same point.

You can analyze the stability of the feedback loop between MAPK and PKC by iteratively determining the response of [MAPK] to [PKC] and the response of [PKC] to [MAPK]. The assumption behind this iterative process is that [MAPK] at a particular time depends on [PKC] at an earlier time. Likewise, [PKC] at a particular time depends on [MAPK] at an earlier time. A graphical technique known as a cobweb diagram is helpful in following the iterative process.

Figure MM11.2 illustrates a cobweb diagram for an initial value of [MAPK] just above the threshold T, about halfway between 10 and 100 nM ( $[MAPK] \approx 10^{1.5} \approx 32$  nM). From this point on the MAPK axis, move vertically to the purple line and horizontally to the [PKC] axis. As described earlier, the resulting point (approximately 0.23  $\mu$ M) is the response of PKC to the initial MAPK concentration. MAPK will now respond to this value (0.23  $\mu$ M) of [PKC]. The resulting [MAPK] (approximately



**Figure MM11.2** Cobweb diagram of MAPK and PKC concentration response curves.

Start by determining the response of [PKC] to 32 nM [MAPK], and repeat this process with MAPK's response to the new PKC concentration, and so on, until converging to a point (e.g., point A).

100 nM) is found by moving horizontally from 0.23  $\mu\text{M}$  on the PKC axis to the black line, and vertically to the MAPK axis. Repeat the process to find the following successive concentration responses: [PKC] = 0.31  $\mu\text{M}$ ; [MAPK] = 200 nM (point A); [PKC] = 0.35  $\mu\text{M}$  (point A). The diagram, and thus the concentration of both kinases, has converged to A, as was observed in Figure 11.14.

You may have noticed that the lines from the curves to the axes were merely for keeping track of the concentrations of each kinase, and are always backtracked in the next kinase response determination. Therefore, the cobweb diagram can be constructed by beginning at [MAPK] = 32, moving vertically to the purple line, horizontally to the black line, vertically to the purple line, and so on, until the diagram converges to A.

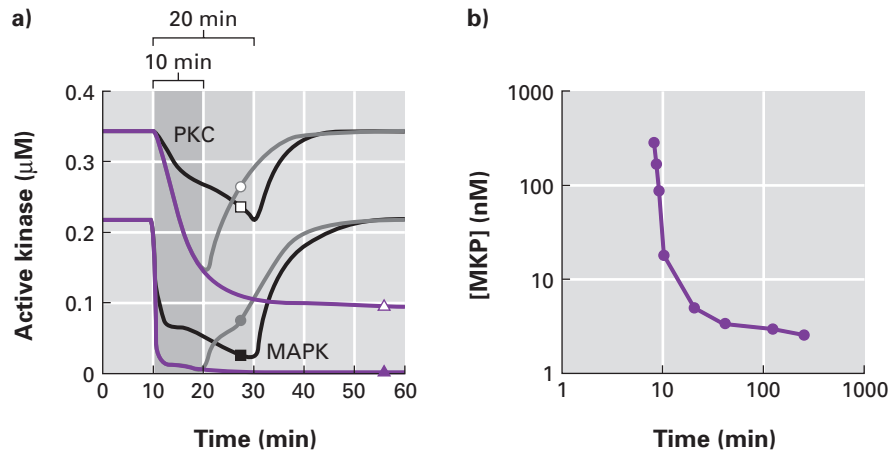
You can construct cobweb diagrams starting at various values of [MAPK]. If you begin at [MAPK] > T, the diagram should converge to A, but if you begin at [MAPK] < T, it should converge to B. This shows that T is a threshold stimulation level. Given concentration-effect curves like those in Figure 11.15, cobweb diagrams allow us to predict the behavior of systems like this feedback loop.

### DISCOVERY QUESTIONS

10. Design an experiment that allows you to test the prediction that robust LTP (i.e., learning) requires more than one isoform of PKC.
11. Describe how the variation in different isoforms of PKC relates to the equations we used to calculate reliability (see pages 377–378).
12. How do the terms “robust” and “reliability” relate to each other in Figure 11.12?

### Can the Modeled Circuit Accommodate Learning and Forgetting?

Although we are oversimplifying what is required to form a memory in your brain, LTP is one critical step. If LTP requires the long-term activation of either PKC or MAPK, is it ever possible to inactivate these kinases and turn off LTP (i.e., forget a memory)? Bhalla and Iyengar added this aspect of learning to their already successful model. MAPK must be phosphorylated (by MAPKK) to be activated; to



**Figure 11.16 Inactivation of LTP by MKP.**

**a)** Activities of PKC (open symbols) and MAPK (filled symbols) were simulated to show status of feedback. The feedback loop was initially activated by a stimulus above point T in Figure 11.15, and then one of three inhibitory inputs was applied: 10 minutes of 8 nM (circles), 20 minutes of 4 nM (squares), or 20 minutes of 8 nM (triangles) MKP activity. Dark and light gray shading denote 10- and 20-minute exposure times of MKP, respectively. **b)** MKP was applied for varying durations and concentrations to determine thresholds for inactivation of the feedback loop.

inactivate MAPK, there is a phosphatase (MAPK phosphatase [MKP]) that removes the activating phosphate. When MKP was added to their model, the investigators discovered more about the bistable circuit. Figure 11.16 is another data-rich figure worth careful dissection. PKC activity (Figure 11.16a, open symbols) was stimulated and then exposed to three levels of MKP activity: 10 minutes at 8 nM (open circle), 20 minutes at 4 nM (open square), or 20 minutes at 8 nM (open triangle). Only the greatest exposure to MKP was sufficient to fully inactivate PKC. Likewise, MAPK was treated with 10 minutes of 8 nM (closed circle), 20 minutes of 4 nM (closed square), or 20 minutes of 8 nM (closed triangle) MKP activity, achieving long-term inactivation (steady-state B in Figure 11.15) only with the greatest MKP exposure.

Only the most intense MKP activity was able to inactivate the feedback loop. The rebound in PKC and MAPK activities after the two lower exposures of MKP was due to a couple of factors: the persistence of AA, due to a relatively slow time course of degradation; and the time required to dephosphorylate the previously activated kinases in the MAPK circuit. The investigators simulated a wide range of MKP concentrations and durations of exposure to the bistable circuit and plotted the combinations needed to switch the circuit off (Figure 11.16b). At high MKP concentrations, inactivation occurs quickly, but there is a minimum threshold of nearly 10 minutes exposure. Conversely, when MKP was applied for very long times, at least 2 nM MKP was required to inactivate the feedback loop.

Bhalla and Iyengar created an integrated circuit simulation that exhibited feedback loop properties, a model that

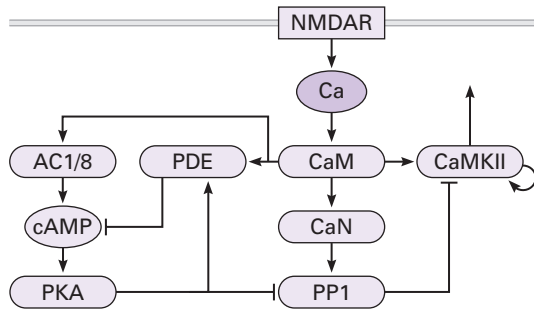
has improved our understanding of LTP. Both the model and real neurons produce prolonged and elevated levels of activation that were initiated by an external signal (EGF binding to its receptor) even after the initial stimulation was removed. Their simulation has quantified what is required to turn off LTP by breaking the feedback loop.

### DISCOVERY QUESTIONS

13. Look at Figures 11.12 and 11.16b and hypothesize which enzyme(s) were responsible for the required ten minutes of MKP exposure to inactivate the feedback loop even when the concentration of MKP was increased tenfold.
14. Explain what was happening to the level of phosphorylated MAPK when the amount of MKP was below 2 nM for 2 hours.

### What Roles Do Other Integrated Circuits Play in LTP?

Having enjoyed this much success, the investigators decided to add more circuitry that interacts with the MAPK circuit we just studied. But first, we need to understand a little bit more about LTP (Figure 11.17). When calcium floods the stimulated neuron, these ions bind to a protein called calmodulin (CaM) to form a calcium/calmodulin (CaM) complex. CaM activates adenylyl cyclase isoforms 1 and 8 (AC1/8), calcineurin (CaN), and a kinase called Ca<sup>2+</sup>/calmodulin-dependent protein kinase II (CaMKII). LTP is stimulated when CaMKII is activated for extended periods



**Figure 11.17** Circuit diagram examining the role of cAMP in LTP.

NMDAR is a voltage-gated calcium ion channel in the plasma membrane of neurons. AC1/8 represents two isoforms of adenylyl cyclase that produce cAMP. PDE is a phosphodiesterase that destroys cAMP. PP1 is a protein phosphatase that removes a phosphate (and thus inactivates) CaMKII. CaM is calmodulin that is activated when it binds calcium, and CaMKII is a kinase activated by CaM. CaMKII can autophosphorylate (small looping arrow). CaN, calcineurin, is a kinase that becomes activated when it binds CaM. PKA inhibits PP1 by adding a deactivating phosphate onto PP1.

of time. To be activated, CaMKII must be phosphorylated (by itself or another kinase). Bhalla and Iyengar hypothesized that protein kinase A (PKA) (a cAMP-dependent kinase) played a critical indirect role in the prolonged activation of CaMKII, and they wanted to use their computer simulation to test their prediction. They called this cAMP/PKA regulation of CaMKII a “gating” control, which can be thought of as another toggle switch. If enough PKA were activated, then CaMKII would become activated (a second bistable switch). The connection between CaM and CaMKII produced a new hard-wired connection that integrated smaller individual circuits (panels c, i, j, m, n, o in Figure 11.11). The new integrated circuit (Figure 11.17) would help the investigators determine whether interactions between CaMKII, cAMP, and CaN were sufficient to produce prolonged activation of CaMKII even after the amount of cytoplasmic  $\text{Ca}^{2+}$  inside the neuron returned to resting concentrations. The critical switch point in this circuit is CaM, which activates two competing signals that determine whether CaMKII will be activated or not. As we have seen before, stochastic behavior of proteins and noise in the switching mechanism will influence the outcome.

### DISCOVERY QUESTIONS

- How can one kinase (e.g., CaMKII) phosphorylate so many different proteins? In other words, predict how so many different substrates can bind to a single active site.

- How can a kinase (e.g., PKA) activate one enzyme (PDE) and inhibit another (PP1) even though it adds a single phosphate in both cases?
- How many pairs of proteins can you find in Figure 11.17 that are competing to produce opposite outcomes? What role would stochastic bursts of enzymatic activity play in the competitions you found in this figure?

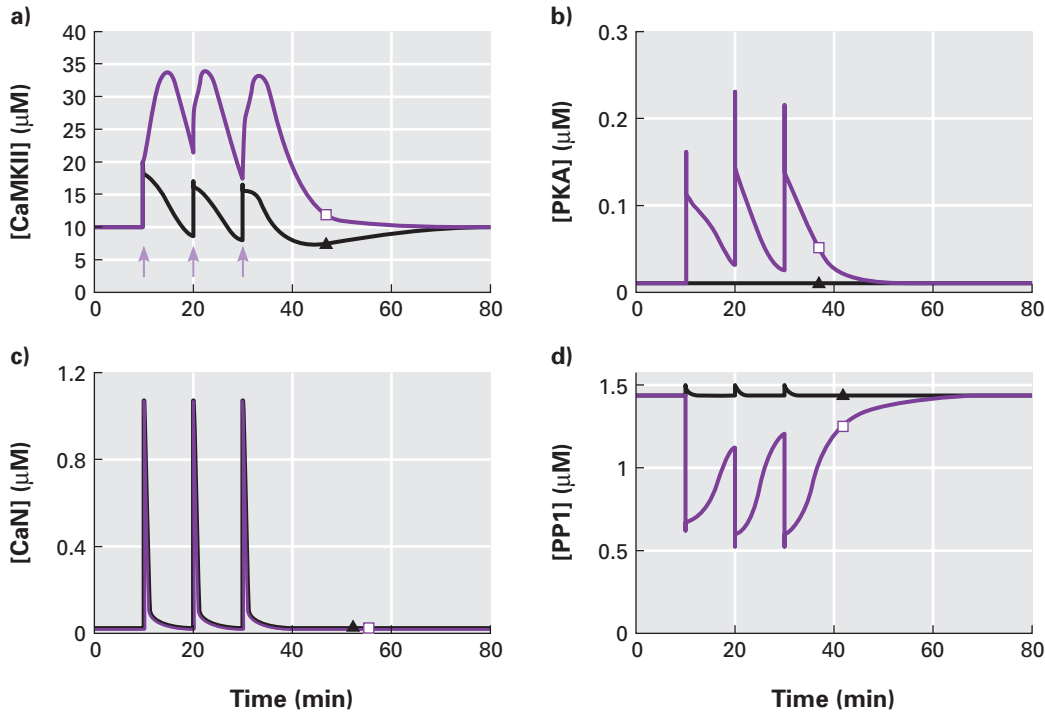
When the new integrated circuit was stimulated with conditions that produce LTP in a real neuron, the simulated activities of four enzymes were graphed (Figure 11.18). When the concentration of cAMP was maintained at basal levels in the computer simulation, the external stimulation produced three transient increases in activities for CaMKII, CaN, and PP1, but had no effect on PKA. When the concentration of cAMP was allowed to rise as directed by the model, CaMKII, PKA, and PP1 exhibited significant changes in activities that were not present when cAMP concentration was fixed at resting levels, though CaN activity was unaltered. PKA needed elevated cAMP in order to become activated and PP1 was substantially reduced instead of slightly increased. Since we are particularly interested in the system’s ability to initiate LTP via CaMKII, CaMKII’s response to cAMP concentration changes was especially noteworthy. The amplitude of CaMKII activity was increased by about twofold, and the time it took for the activity to return to resting levels also doubled to about 20 minutes. The prolonged activation of CaMKII in the presence of elevated cAMP was due in large part to the significant inhibition of PP1, which otherwise would have inactivated CaMKII quickly. Nevertheless, CaMKII activity only increased transiently, which is not the expected behavior for a bistable toggle switch. Thus, the bistable toggle switch under these simulated conditions quickly reverted to the off position (equivalent to steady-state level B in Figure 11.15).

### DISCOVERY QUESTIONS

- Predict what is needed to stimulate CaMKII to become activated for the long term. Design an animal experiment to test your hypothesis.
- Would cAMP levels rise or fall when given the “freedom” to vary after stimulation?
- Which enzyme(s) exhibited the greatest duration of activity after the stimulus was removed?

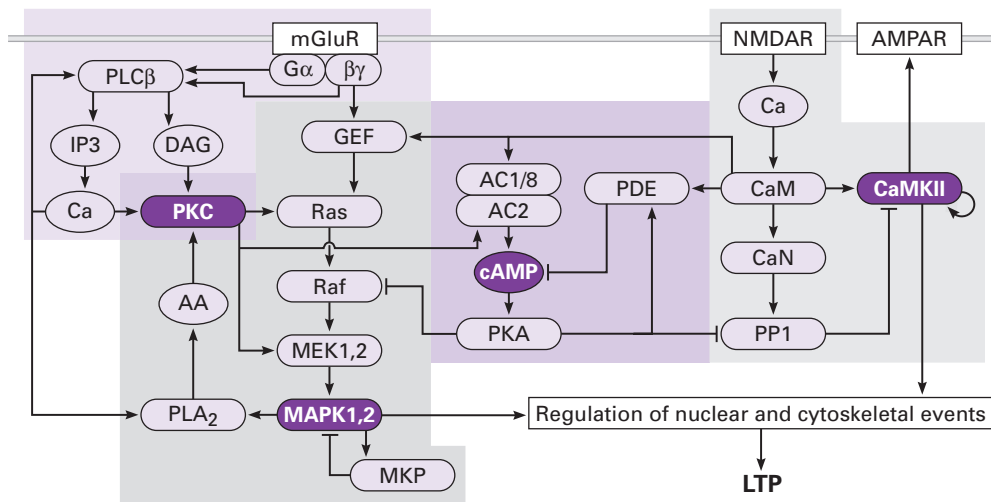
### Do They Need a More Complex Model to Match Reality?

Finally, Bhalla and Iyengar were ready to create a fully integrated circuit. This model integrates four individual signaling circuits (Figure 11.19). When electrical engineers view



**Figure 11.18 Temporary activation of four key enzymes for LTP.**

Graphs show the simulated activities of **a)** CaMKII, **b)** PKA, **c)** CaN, and **d)** PP1 after stimulation with three 100 Hz pulses lasting 1 second each and separated by 10 minutes (which produces LTP in real neurons). Open squares indicate that cAMP concentrations were allowed to rise; filled triangles indicate that cAMP levels were artificially maintained at resting concentrations. The arrows in panel a) indicate when the three pulses were given to the system, at the 10-, 20-, and 30-minute time points.



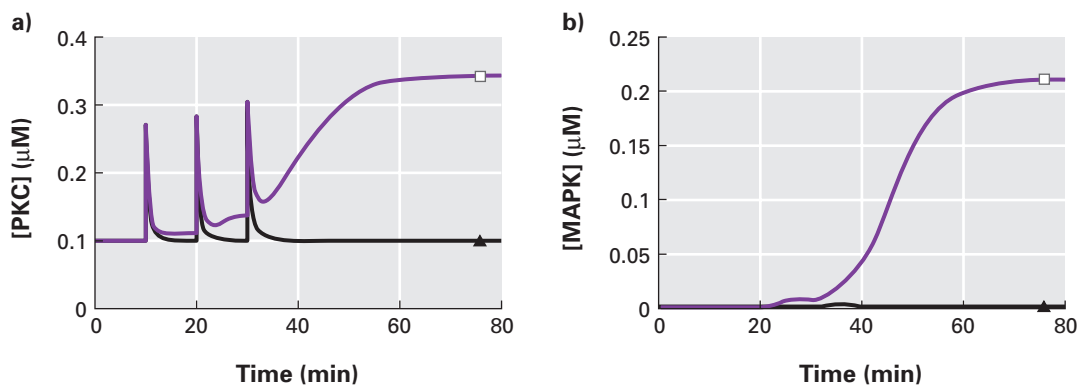
**Figure 11.19 Full-circuit diagram with feedback loop, synaptic output, and CaMKII activity and regulation.**

Two possible end points of the model are represented as AMPAR regulation and stimulation of nuclear/cytoskeletal events, which eventually leads to LTP. Note the feedback loop between PKC and MAPK with PLA<sub>2</sub> as the connecting enzyme. The four shaded regions highlight the four individual signaling circuits.

a circuit diagram such as this, they do not examine each piece individually. They look for functional units of circuitry and the connections between them. Figure 11.19 contains four functional units: PKC, MAPK, CaMKII, and the cAMP circuits. We have seen how PKC and MAPK circuits were connected (see Figure 11.12) and how the CaMKII and cAMP circuits were connected (see Figure 11.17). PKC was connected to the cAMP circuit via AC2. These connections enabled Bhalla and Iyengar to produce their final integrated circuit composed of four smaller circuits.

How powerful a research tool is the computer model that simulates integrated circuits? Can it elucidate critical features that result in LTP? Does the feedback loop between PKC and MAPK play a critical role in LTP? To build their integrated circuit shown in Figure 11.19, all the individual circuits shown in Figure 11.11 were incorporated. The *in silico* neuron was stimulated with glutamate and depolarized at the plasma membrane, which opens the NMDAR calcium ion channel. How did the full integrated circuit respond? Were there any unexpected activities?

The first pair of enzymes examined was PKC and MAPK (Figure 11.20), which are indirectly connected to each other. PKC exhibited a strong on/off response to the three electrical stimulations in the absence of the feedback loop (AA held at basal levels). When the feedback loop was allowed to function, notice how PKC activity substantially increased; also, the duration of the activation was maintained after the stimulation ceased. Compare the PKC activity to the activity of MAPK. In the absence of the feedback loop, MAPK activity was undetectable, but in the presence of the feedback loop, MAPK was strongly activated and the activation was sustained. Notice that MAPK activity began at about 30 minutes.



**Figure 11.20** Activity profile of a) PKC and b) MAPK.

LTP is achieved with a fully functional integrated circuit (open squares) but not when the feedback loop is blocked by holding AA at resting concentrations (filled triangles). Electrical pulses of 100 Hz for 1 second were applied at 10, 20, and 30 minutes.

## DISCOVERY QUESTIONS

21. Why does MAPK have to “wait” for 30 minutes before it gets activated? Explain your answer in terms of the full circuit in Figure 11.19.
22. Using the full circuit, follow the connections between NMDAR and PKC to explain the step-wise increase in PKC activation to its final level of activity with the feedback loop.

Four additional enzymes were examined in this simulation (Figure 11.21). PKA was activated  $\pm$  the feedback loop, as was CaMKII. But note that both of these enzymes exhibited a higher amplitude and sustained activity in the presence of feedback than in the absence of feedback. CaN was activated, but there was no difference  $\pm$  feedback loop. PP1 had its activity reduced, as one would expect, and its inhibition was of greater amplitude after the stimuli were removed. We are seeing the results of another bistable toggle switch that does not require any new transcription or translation. PP1 inhibits CaMKII, but PKA inhibits PP1 while also stimulating its own activity. In this toggle switch, CaN is not involved at all, so it is no surprise that CaN activity was not sustained when the feedback loop was functioning.

The most striking outcome we discovered was that complex circuits exhibit emergent properties without the need for new mRNA or protein production! What are the implications for microarray and proteomics research, when new biochemical properties emerge from a defined circuit that lacks additional transcription or translation? The emergent property is both good news and bad news for students who like to cram in material minutes before a test. The good news is that you can learn new information (i.e., LTP is produced)

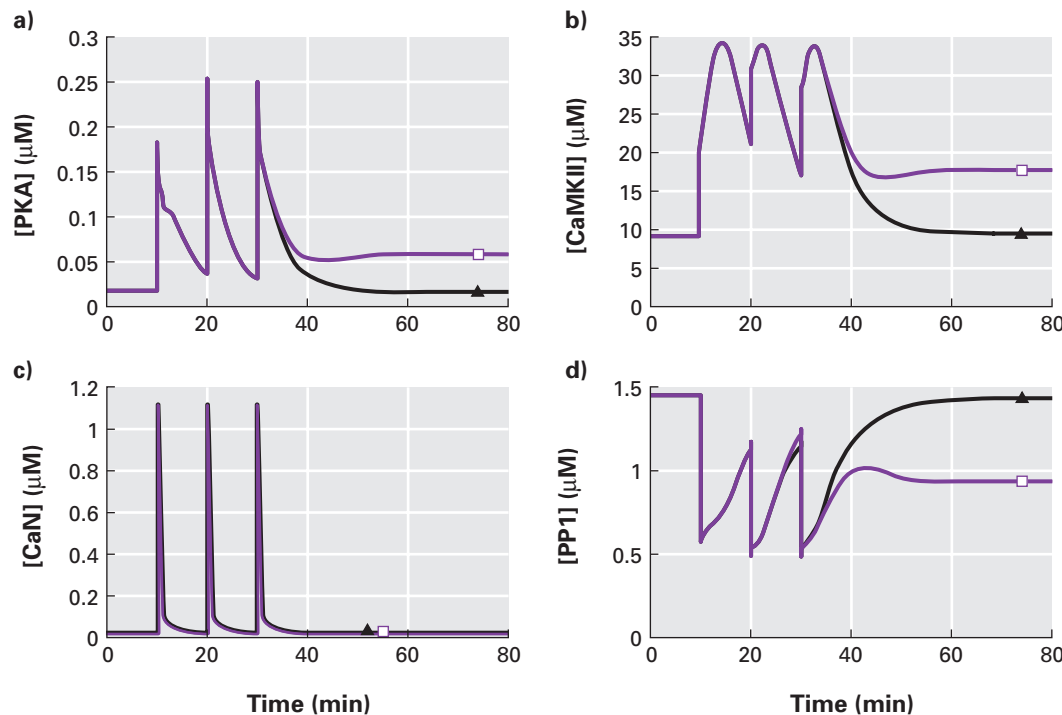
very quickly in the absence of new protein production, which would take about an hour to accumulate. The bad news is that LTP has two phases just like your memory: short-term and long-term. MKP can be thought of as the “off switch” to learning. MKP inactivates MAPK, which is needed to initiate transcription in the nucleus. MAPK indirectly controls the activity of CaMKII (via the feedback loop that includes PLA2, PKC, AC2, PKA, and PP1), which is also needed to produce LTP. Therefore, if your MKP activity exhibits a stochastic burst of activity, what you “learned” while cramming will be lost from your short-term memory and never reach your long-term memory.

MKP has the ability to determine which line (open squares vs. filled triangles in Figure 11.21) better represents your ability to remember something. The toggle switch of memory is controlled by the concentration of a few molecules to produce a system that is bistable. Therefore, MKP may act as a timer to bring to an end the early phase of LTP. Of course, you want to be able to retain information for a longer time, but occasionally you forget. MKP becomes the first checkpoint at which some people will not be able to commit a new memory to long-term storage. In Bhalla and Iyengar’s full integrated circuit, MKP is not connected to

any other proteins, but this is unlikely to accurately reflect reality. What circuits control MKP activity? How do our neurons “decide” which way to flip the toggle switch for LTP?

### Are LTP and Long-Term Memory Related?

It has been known for years that LTP is divided into two phases, slow and fast. You have just studied the fast phase, which was composed of four integrated circuits. In a real neuron, the slow phase of LTP changes the structure of synapses between neurons, requiring the production and intracellular transportation of new proteins (i.e., a dynamic proteome). The goal in learning is to change one synapse (to store a new memory) without altering other synapses (analogous to altering one computer file but not any others on your computer). MAPK and CaMKII lead to the slow production of new proteins that must be sent to the correct synapses. Bhalla and Iyengar hypothesize that the feedback loop described in their simulation controls the localization of new proteins via the cytoskeleton so they wind up at the correct synapse. What a huge conclusion to draw based on computer simulations! For the sake of science, it does not



**Figure 11.21** Activity profiles of PKA, CaMKII, CaN, and PP1.

The simulation was run with the feedback loop functioning (open squares) and with feedback blocked by holding AA fixed at resting concentrations (filled triangles). **a)** The Ca-stimulated PKA waveforms are almost identical, but the baseline rises when feedback is on, because PKC stimulates AC2 to produce additional cAMP. **b)** Activity profile of CaMKII. **c)** Activity profile of CaN is unaffected by the presence or absence of feedback. **d)** Activity profile of PP1 in the presence and absence of the feedback loop.

matter whether their hypothesis is right or wrong. Either way, the investigators have made a prediction based on their model that is testable in the lab. If they are correct, they may need to rent tuxedos and fly to Stockholm. If they are wrong, someone will experimentally demonstrate an inconsistency and advance the field by eliminating one incorrect possibility. Therefore, a completely computer-based research project has taught us a lot about integrated circuits, provided us with some new insights about LTP, and generated a new idea to explain long-term memory.

### DISCOVERY QUESTIONS

23. What difference do you see when comparing the activation of CaMKII in Figure 11.21b and 11.18a? Propose a mechanism for this difference.
24. What is the consequence for LTP when PP1 is more inhibited in the presence of the feedback loop?
25. Explain to DNA microarray and proteomics experts why the LTP integrated circuit has profound implications for their research.
26. Hypothesize how a person who has a photographic memory might have a genotype that permits him or her never to lose fast LTP.
27. Hypothesize how an older person might retain his or her long-term memories but not be able to learn new things.

### What Have We Learned (How Much LTP Have We Generated)?

Bhalla and Iyengar's paper had two major goals. Its primary goal was to understand how protein circuits work and to detect any synergistic properties. From this computer-based simulation, we now understand that integrated protein circuits can explain some old observations:

- It is possible to produce a prolonged signaling effect even after the original stimulus is removed.
- Protein circuits can activate feedback loops, which can provide the mechanism for synergistic properties.
- It is possible to understand the signaling threshold required to control a bistable toggle switch.
- A single stimulus can lead to multiple output pathways.
- Complex networks provide the mechanism for critical aspects of "biological design" that can provide the selective pressure for evolutionary steps. These design principles include redundancy, robustness, and fail-safe.
- All these features can be accomplished in the absence of new protein synthesis.
- Genetic variation in the population will result in certain genotypes that will respond differently to the same stimuli.

The second goal of their research was to gain a better understanding of the process of LTP, which leads to the formation of a new memory. In this area, the investigators predicted:

1. CaN does not play a major role in the LTP activation of CaMKII, which most believe is a critical protein in learning.
2. PKA is downstream of both PKC and MPAK and upstream of CaMKII.
3. PKC can be activated by more than one input, each with critical threshold concentrations to flip the bistable toggle switch to the on position.
4. The adenylyl cyclase isoforms play separate roles in LTP. AC1 is activated by CaM and allows the system to achieve a rapid inhibition of PP1 after  $\text{Ca}^{2+}$  influx. AC2 is activated by PKC and provides the sustained inhibition of PP1, which leads to the prolonged activation of CaMKII and thus LTP. AC2 provides the necessary step for the feedback loop, which is critical for the synergistic response.

We have spent a substantial amount of time and energy studying circuits. The work by Bhalla and Iyengar will lead to very focused and efficient research in the lab through computer-guided hypothesis formulation. These two investigators are quick to acknowledge that their model has limitations, such as compartmental constraints and changes in protein levels due to translation of new proteins. They wrote, "[M]odels such as these should not be considered as definitive descriptions of networks within the cell, but rather as one approach that allows us to understand the capabilities of complex systems and devise experiments to test these capabilities." They also leave us with an interesting question. Is the LTP integrated circuitry related to immunological "memory," in which memory white blood cells can "remember" pathogens years after initial contact? In silico circuit analysis may help us understand immunology as well as many other complex systems.

### Can We Understand Cancer Better by Visualizing Its Circuitry?

The human genome project is not an end point, but rather a beginning. Knowing all the DNA content of humans, even if we knew every single nucleotide polymorphism (SNP) in the gene pool, would only give us the raw material to ask interesting questions. How do we learn? What drives our sexuality? What makes some people early risers and others night owls? Why do some people live to be over 100 years old?

We cannot answer these questions yet, because genomics is a work in progress. However, each year we have better tools and more data, allowing us to assemble more complex and comprehensive models that better describe biological



**DATA**  
cell cycle  
DNA repair

**LINKS**  
Kohn's references  
Kurt Kohn

processes. For example, Kurt Kohn from the National Cancer Institute (NCI) produced two huge circuit diagrams of cell cycle control and DNA repair. The first question most nonbiologists ask at this point is, "Why would the NCI want to study cell cycle and DNA repair?" Cancer results from the loss of cell cycle control and the inability to fix damaged DNA. A lot was already known about cell cycle and DNA repair (see Kohn's references), and Kohn summarized all this information in comprehensive circuit diagrams. But first he had to invent a language (Figure 11.22). Unlike Bhalla and Iyengar, Kohn permitted many more interactions to take place in his circuit diagrams, and therefore he needed a larger vocabulary. You should note the different types of interactions needed to complete Kohn's circuits. Some symbols are more intuitive than others, but each interaction is found inside your cells. Do not memorize the symbols, but make sure you understand what each symbol means in the vocabulary.

processes. For example, Kurt Kohn from the National Cancer Institute (NCI) produced two huge circuit diagrams of cell cycle control and DNA repair. The first question most nonbiologists ask at this point is, "Why would the NCI want to study cell cycle and DNA repair?" Cancer results from the loss of cell cycle control and the inability to fix damaged DNA. A lot was already known about cell cycle and DNA repair (see Kohn's references), and Kohn summarized all this information in comprehensive circuit diagrams. But first he had to invent a language (Figure 11.22). Unlike Bhalla and Iyengar, Kohn permitted many more interactions to take place in his circuit diagrams, and therefore he needed a larger vocabulary. You should note the different types of interactions needed to complete Kohn's circuits. Some symbols are more intuitive than others, but each interaction is found inside your cells. Do not memorize the symbols, but make sure you understand what each symbol means in the vocabulary.

**DISCOVERY QUESTIONS**

- 28. Use Kohn's symbolic language to diagram how *Endo16* is regulated (see Figure 10.22).
- 29. Use Kohn's symbolic language to "translate" the PKC/MAPK feedback loop (see Figure 11.12).

Examine the web versions of the two circuit diagrams: the cell cycle and the DNA repair circuits. The first things you should notice are their sheer size and complexity. The circuit diagrams facilitate systems biology comprehension, which is appropriate because cancer is a malfunctioning system. The key to understanding complex circuit diagrams is to find functional units that are connected to other functional units. For example, look at the Myc functional unit that is magnified in Figure 11.23.

Myc is a known **oncogene**, and here is some information as it appeared in Kohn's annotations appendix:

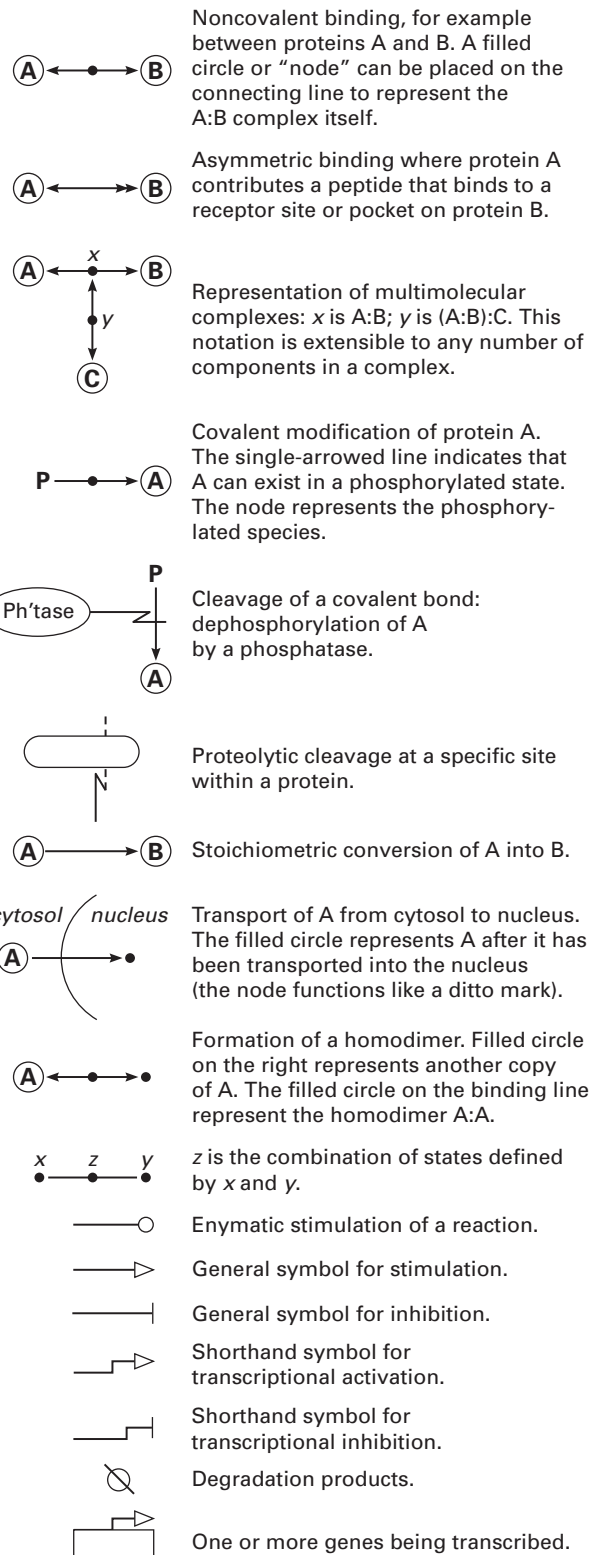
*C35*: *cdc25A* may be transcriptionally activated by c-Myc; the Myc:Max heterodimer binds to elements in the *cdc25A* gene and activates its transcription.

*M1*: c-Myc and pRb compete for binding to AP2.

*M2*: AP2 and Max compete for binding to c-Myc. AP2 and Myc associate in vivo via their C-terminal domains.

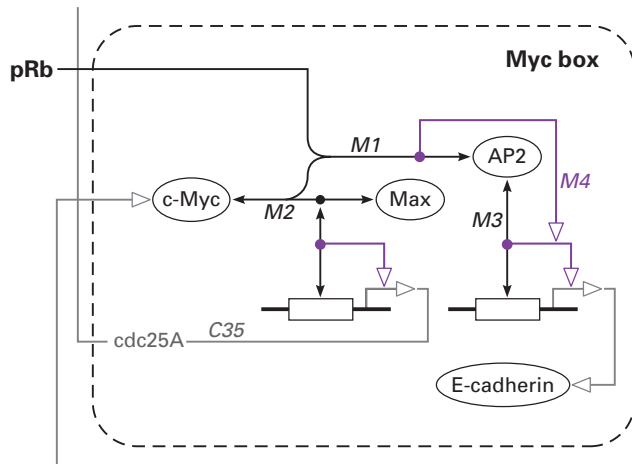
*M3*: The E-cadherin promoter is regulated via AP2 recognition elements.

*M4*: c-Myc and pRb enhance transcription from the E-cadherin promoter in an AP2-dependent manner in epithelial cells (mechanism unknown). Activation by pRb and c-Myc is not additive, suggesting that they act upon the same site, thereby perhaps blocking the binding of an unidentified inhibitor. No c-Myc recognition element is required for activation of the



**Figure 11.22** Symbolic language created for cell cycle and DNA repair circuit diagrams.

In the electronic version, black lines indicate binding interactions and stoichiometric conversions; red are covalent modifications and gene expression; green are enzyme actions; blue are stimulations and inhibitions.



**Figure 11.23** Myc subsection from Kohn's cell cycle circuit.

The Myc box can be found in area D10, in the top right corner of the DNA repair circuit diagram. Symbols are defined in Figure 11.22.

E-cadherin promoter by c-Myc. Max blocks transcriptional activation from the E-cadherin promoter by c-Myc, presumably because it blocks the binding between c-Myc and AP2.

The proteins retinoblastoma (pRb, a **tumor suppressor**) and c-Myc compete to bind to the transcription factor AP2. The two possible combinations c-Myc/Max and pRb/AP2 can each bind to cis-regulatory elements to initiate transcription of E-cadherin. However, c-Myc can also bind to Max and lead to the transcription of *cdc25A*, a phosphatase that regulates the activity of a number of proteins in other functional domains. However, note that regulation of c-Myc in this functional unit is influenced by a different functional unit, as indicated by the arrow to the left of c-Myc.

From our examination of learning circuits, we know that feedback loops are very important. Let's look at one feedback loop in each of these circuit diagrams. Open the **cell cycle** file to locate *cdc25C* and *cycB:cdk1* (in area H4).

1. *cdc25C* is activated by phosphorylation (blue line with green *C18* label).
2. *cdc25C* removes phosphates from *cdk1* at amino acids T14 and Y15 (shown as a green line coming from *cdc25C*). Removal of the phosphates removes the inhibition from *cdk1*.
3. *cdk1* interacts with *cycA* (black *C5* label), and this interaction stimulates the phosphorylation of *cdc25C* (green *C36*). Although this positive feedback loop is difficult to recognize at first glance, with practice and familiarity with the system, you would be able to see other examples.

The second feedback loop is taken from the DNA repair diagram. Locate **p53** in area E7–9 on the right side of the

circuit. Immediately you can tell that p53 is a critical component, because there are so many interactions emanating from it. If you look below p53, you will see three black dots (black *P18*), which indicates that p53 (the protein) can form homotetramers. The homotetramer can initiate transcription (black *P17*) from seven different genes (one box with seven red lines emanating from it). One of the activated genes is *Mdm2* (red *P39*), and the protein MDM2 can form a heterodimer with p53 (black *P28*), which inhibits transcription of the *Mdm2* gene (blue *P29*). These components form a negative feedback loop. In the published paper accompanying this circuit diagram, Kohn reported that MDM2/p53 formation leads to the degradation of p53 (black *P30*), as indicated by the blue arrow labeled *P31*.

All these interactions were discernible from circuit diagrams with very few words, and they allow us to comprehend interactions and make predictions. Kohn's circuit diagrams are works in progress; don't make the mistake of assuming that if a component is shown in the diagram, all of its interactions are known. Our knowledge is limited, and studying circuit diagrams will focus our research more efficiently. We are beginning to make headway on larger-scale models, but we still have a lot to learn.

### DISCOVERY QUESTIONS

30. In Kohn's circuit diagrams, try to locate one small functional subcircuit (similar to the Myc subcircuit), and describe in words what you see. Which components are linked to other functional units?
31. Find BRCA1, the breast cancer gene, located in area F10 of the DNA repair circuit. Describe in words how BRCA1 interacts with p53.
32. One of the blue lines below p53 is labeled *P31*. Notice that two different repressors can block the degradation of p53. Given what you know about circuits, is blocking this degradation an important process or not? What effect would two repressive interactions have on the reliability of blocking the degradation of p53? Can you find any other pathways that also block the degradation of p53?
33. If you became fluent in this symbolic language, do you think you should get a foreign language credit?

## Summary 11.1

We have seen that genetic toggle switches can be formed with very few genes. Genomes contain hundreds or thousands of toggle switches so they can produce proteomes and metabolomes in response to environmental changes. The "choice" between two responses can be driven by particular factors (e.g., nutrient concentration for  $\lambda$  phage lysogenic



## LINKS

Jim Collins  
Charles Cantor  
Timothy Gardner

versus lytic lifestyles) or by stochastic behavior of proteins (e.g., *recA* and *lacZ* promoters). Given our understanding of how bistable genetic switches operate, can we design and

construct synthetic bistable toggle switches that function as predicted? If so, could these synthetic switches be used to produce useful devices? If you were designing a bistable switch for gene therapy, you would want to incorporate a level of redundancy that provided the patient with an acceptable degree of reliability. It might seem uncomfortable to treat genes like machines with reliability that can be calculated, but this approach to genome analysis is valuable. Applying engineering principles to genomics helps us understand why diploids evolved the apparent waste in complex circuits with redundancies. In Section 11.2, we study a series of synthetic genomic circuits that illustrate the cutting edge of our knowledge of how biological circuits work.

## 11.2 Synthetic Biology

Every time we build a model, we are trying to create a simple version of a complex system. Given the lessons in Section 11.1, is it possible to design and construct synthetic circuits? The new field of **synthetic biology** (analogous to synthetic chemistry) applies engineering principles to genomic circuits to construct small biological devices that should help us understand how naturally evolved circuits function. Perhaps someday we will be able to construct synthetic circuits that can perform beneficial tasks, but for the foreseeable future, we will use synthetic components, devices, and systems to elucidate how natural circuits function. In this section, we will consider four case studies that increase in complexity. Use these examples as starting places for synthetic circuits you might want to construct and test. Discovery Questions will allow you to explore the field of

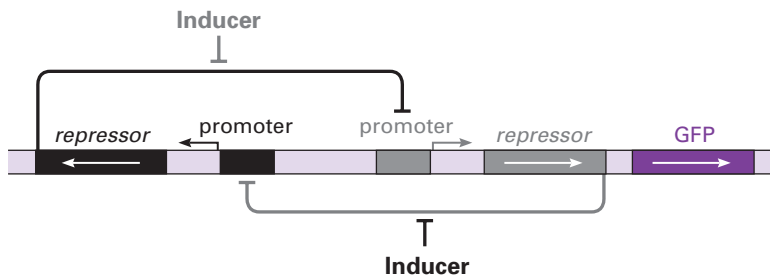
synthetic biology, including an interactive construction web site which is part of the **BioBricks** project at the Massachusetts Institute of Technology (MIT).

### Can Humans Engineer a Genetic Toggle Switch?

Often there is a division among biologists who like to argue about which experiments are “real science”: those working with naturally produced organisms and molecules vs. those working with theoretical models to unify experimental observations. Both approaches are necessary for a complete understanding, but it is important to remember that predictions based on theoretical models must be tested with real experiments. In 2000, two research groups described genetic toggle switches that had been design and constructed. Both groups used *E. coli* as the host for their switches, but they made two different types of switches. One group built a toggle switch very similar to the one used by  $\lambda$  phage to choose its lifestyle. We’ll examine this switch first. Later we examine a switch that oscillates like a circadian clock.

### How to Build a Toggle Switch

Timothy Gardner and his colleagues Jim Collins and Charles Cantor at Boston University built a toggle switch that could choose between two states. However, rather than letting *E. coli* determine which direction to go, they constructed a toggle switch that could be regulated by the investigators (Figure 11.24). In this switch, they needed two constitutive promoters (colored black and gray) and two repressor genes (also colored black and gray). The black repressor protein silences the gray promoter, which drives production of the gray repressor protein. Conversely, the gray repressor protein silences the black promoter, which drives production of the black repressor protein.




**Figure 11.24** Theoretical two-gene bistable toggle switch.

The black gene “repressor” is transcribed from its black promoter. The black repressor protein binds to the gray promoter to block the production of the gray repressor protein. The gray repressor protein blocks the production of the black repressor protein when the gray repressor protein binds to the black promoter. To detect which state the toggle switch is in, GFP was placed downstream of the gray promoter so that the gray repressor and GFP are produced simultaneously.

Thus, if the black repressor protein were produced, the gray repressor protein could not be produced, and vice versa. This is an example of a synthetic bistable toggle switch similar to the theoretical one in Figure 11.1a. To control their toggle switch, the investigators utilized two inducer drugs, each of which incapacitates one of the repressor proteins.

The group from Boston University built and tested several different plasmids with a variety of promoters to see which would successfully produce a bistable toggle switch (Figure 11.25). To measure switching, the **reporter gene** GFP was added to the device to produce a glow-in-the-dark protein when the gray repressor protein was produced. As shown in Figure 11.25a, the investigators compared two plasmids, pIKE107 and pIKE105. The only difference between these two was the use of two different ribosomal binding sites that affected the efficiency of translation of the RNA into protein. Both 107 and 105 were capable of producing GFP (and thus the gray repressor as well), using an **inducible promoter** and a drug called IPTG. Note that 105 did not produce as much GFP as 107. When the inducer IPTG was removed, 105 was incapable of sustaining its output; therefore, it was not a *bistable* switch. However, 107 was capable of sustaining the production of the gray repressor and GFP even after IPTG was removed; thus, it was stable. When 107 was exposed to the second inducer drug, called anhydrotetracycline (aTc), production of GFP (and thus the gray repressor) was eliminated within a couple of hours after induction of the black repressor. The

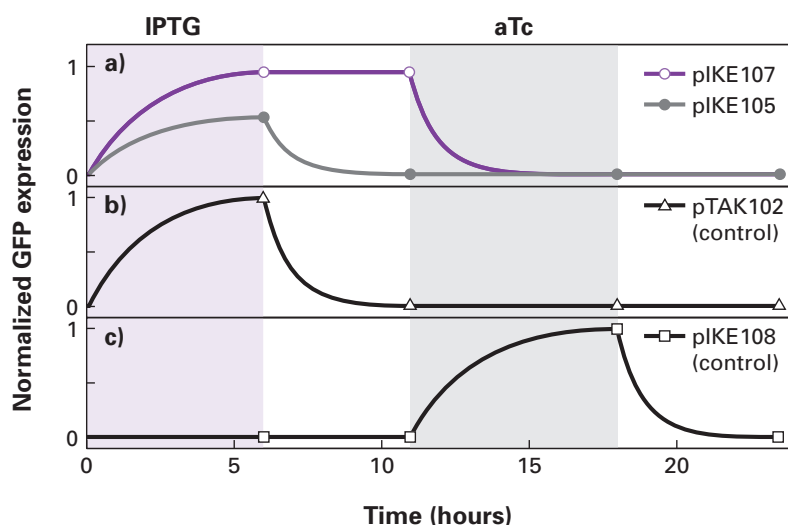
experiment was well controlled (Figure 11.25b–c), since each half of the toggle switch was able to induce GFP production but neither of these “half-switches” was stable. Gardner’s research demonstrated it is possible to take a theory (see Figure 11.1a) and construct a biologically functional switch in a test tube that works inside living cells.

**LINKS**   
indicated plasmids  
**METHODS**  
inducible promoter

### DISCOVERY QUESTIONS

34. Explain why pIKE105 was considered a failure and the control plasmid pTAK102 was deemed a success.
35. Based on the data in Figure 11.25, determine which drug (IPTG or aTc) was the gray inducer and which was the black inducer in Figure 11.24.
36. Design a bistable toggle switch that could be used in gene therapy to produce a protein on demand (e.g., insulin). Include in your design how the protein of interest could be turned on and off.

The success of Gardner’s work is encouraging to those who want to understand genomic circuits. Synthetic circuits can be designed and validated experimentally. It also represents a success in “forward engineering,” in which simple genetic circuits serve as models for more complex systems. In a more applied sense, the bistable toggle switch



**Figure 11.25** Experimental two-gene bistable toggle switch.

*E. coli* were transformed with the indicated plasmids and exposed to the inducer drugs as indicated by the shading. **a)** The only difference between pIKE107 and pIKE105 was different ribosomal binding sites that affected the rate at which the encoded proteins were translated. **b)** pTAK102 control plasmid contained only the IPTG-inducible promoter upstream of the GFP gene. **c)** pIKE108 control plasmid contained only the aTc-inducible promoter upstream of the GFP gene.



## LINKS

Michael Elowitz  
Stan Leibler

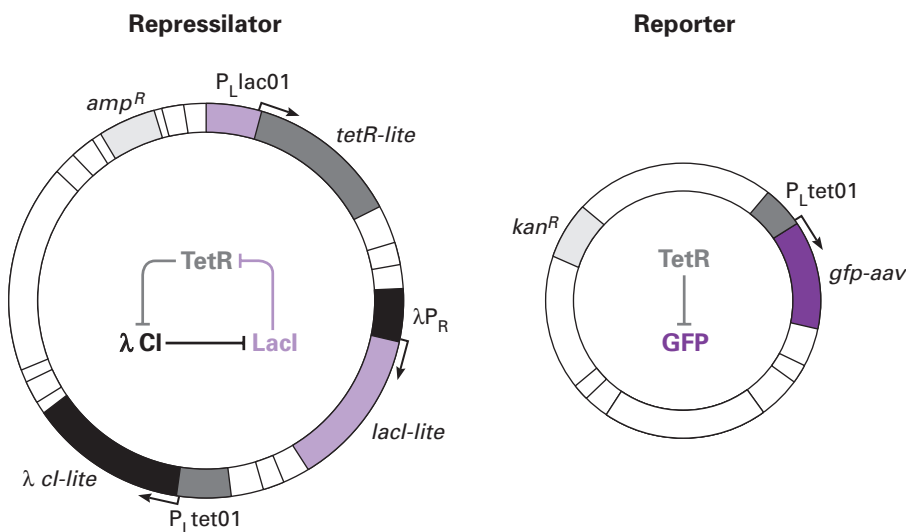
may prove useful in gene therapy and other biotechnology methods that require a silent gene to be activated at a certain point and then sustained. Finally, Gardner's team produced a bistable toggle switch, which is considered the first **genetic applet**, (a term derived from small computer programs called Java applets). Applets are self-contained programs, and Gardner's genetic applet is capable of being programmed (turned on or off, similar to the digital version of 1 or 0). Although the genetic applet is a long way from becoming a biological computer, the potential to store information in DNA is intriguing.

### Can We Build a Synthetic Oscillating Clock?

One lesson from genomics is that interdisciplinary collaborations are the norm rather than the exception. Biologists need to collaborate with physicists, mathematicians, chemists, even graphic artists. While still a biology graduate student in Princeton University, **Michael Elowitz** teamed up with physicist **Stan Leibler** to construct a synthetic oscillating clock. To create a self-perpetuating cycling toggle switch required more genes than a two-gene genetic applet (Figure 11.26). First, you will notice that two plasmids were used. The larger **repressilator** plasmid controlled the cyclical nature of the output, while the smaller reporter plasmid

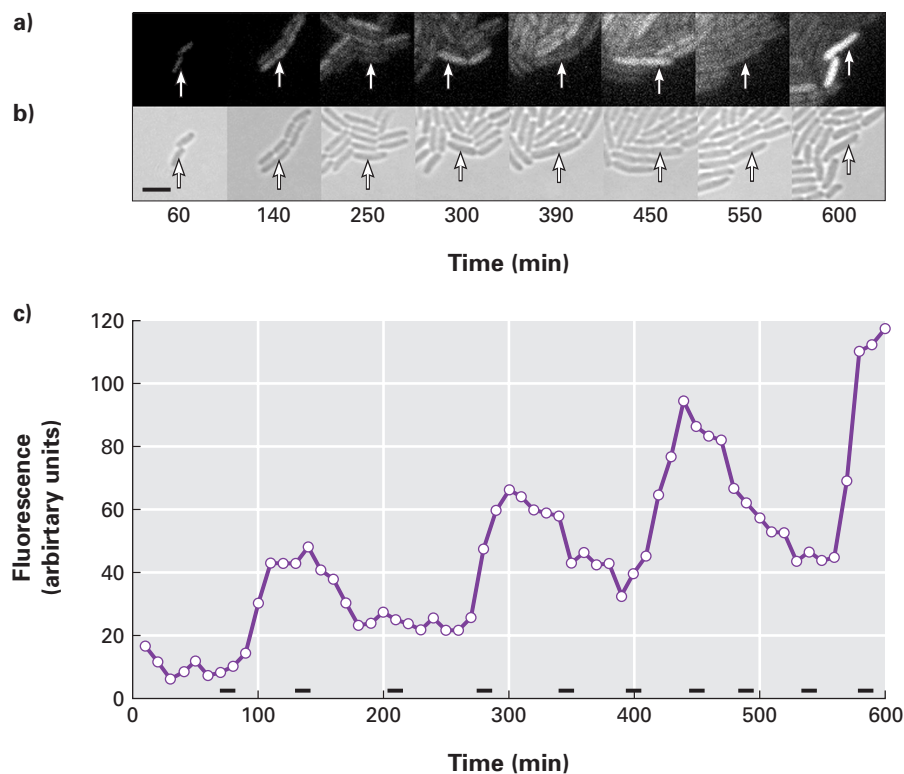
produced GFP. Three promoters and three protein repressors were used in the repressilator plasmid: the  $\lambda$  CI protein represses the production of LacI; the LacI protein represses the production of TetR protein; TetR represses the production of CI. You can see that each gene was repressed by one of the three repressors encoded on the repressilator plasmid. Therefore, the repressilator was a genetic closed circuit or triple negative feedback loop. Since it was impossible to directly observe the repressilator circuit in action, the reporter plasmid encoding GFP displayed the activity of the repressilator. GFP production was constitutive except when repressed by TetR that was part of the repressilator cycle; the *E. coli* cells lost their fluorescence every time the synthetic cycle produced TetR.

Figure 11.26 is a nice theoretical model, but does it actually work? Figure 11.27 dramatically illustrates how well the theory worked inside growing *E. coli*. The photos allow us to follow a single cell (Figure 11.27a–b) through its oscillations of GFP production (Figure 11.27c). The amount of GFP in one cell was measured, which revealed a fluorescence periodicity of about 150 minutes. What was so striking about the periodicity was that 150 minutes is longer than the *E. coli* cell cycle (about 65 minutes). The graph indicates cells divide more rapidly than the repressilator cycles; the investigators produced a cyclical circuit that out-lived its cellular host.



**Figure 11.26** Two plasmids needed to make and see the output of the repressilator synthetic circuit.

The repressilator plasmid contains a cyclic negative feedback loop composed of three repressor genes and their corresponding promoters.  $P_{Lac01}$  and  $P_{tet01}$  are strong promoters that can be tightly suppressed by LacI and tetracycline, respectively. The third promoter,  $\lambda P_R$ , is repressed by CI (see Figure 11.2). The three repressor genes are appended with the suffix "lite" to indicate the encoded proteins degrade rapidly. The *gfp-aav* protein encoded on the reporter plasmid is also degraded rapidly, and its production is regulated by the  $P_{tet01}$  promoter.



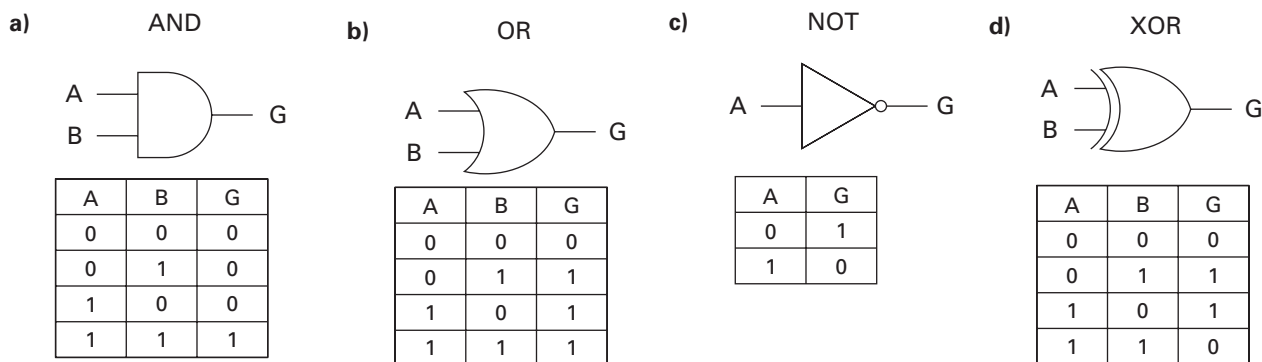
**Figure 11.27** Cyclical toggle switching in live bacteria.

**a)** Fluorescence and **b)** phase contrast microscopy images of cells, revealing the time course of GFP expression and cell growth beginning with a single bacterium containing the repressilator and reporter plasmids (see Figure 11.26). Scale bar in **b)** indicates 4  $\mu\text{m}$  in the photographs. **c)** The pictures in **a)** and **b)** correspond to peaks and troughs in the time course of GFP fluorescence intensity of the selected cell. Bars at the bottom of panel **c)** indicate the timing of cell division, as estimated from the phase contrast photomicrographs in panel **b)**.

### Math Minute 11.3 How Can You Visualize Gene Regulation Logic?

The schematic representation of the repressilator (Figure 11.26) contains information about how the device was constructed and how it works. However, schematic representations require you to follow the logic in your head to understand the behavior of the system. In this Math Minute, we explore how two other representations, truth tables and logic gates, are used to summarize gene regulation logic and behavior of systems like the repressilator.

Truth tables and logic gates describe a system of binary variables, each of which has the value 0 or 1. Gene regulation can be modeled with a binary system by representing gene expression as either “on” (= 1) or “off” (= 0) and regulatory elements as either “present” (= 1) or “absent” (= 0). For example, the simple binary function AND, which has the value 0 unless both input variables have the value 1, was used in statement 2 of Table 10.3 to model the toggle between control by module A and module B. The logic gate and truth table for the AND function are shown in Figure MM11.3a. The purpose of a truth table is to list the value of output variables for each possible combination of input variable values. A logic gate is a standardized symbol that encapsulates the same information for elementary functions; some of the most common logic gates and the corresponding truth tables are shown in Figure MM11.3.



**Figure MM11.3 Elementary logic functions.**

Logic gate and truth table are given for each function. All gates, except NOT, receive two binary inputs (A and B) and return a single binary output (G). **a)** AND activates the output only if both inputs are present. **b)** OR activates the output if either (or both) of the inputs are present. **c)** NOT inverts the input. **d)** XOR activates the output if either (but not both) of the inputs is present.

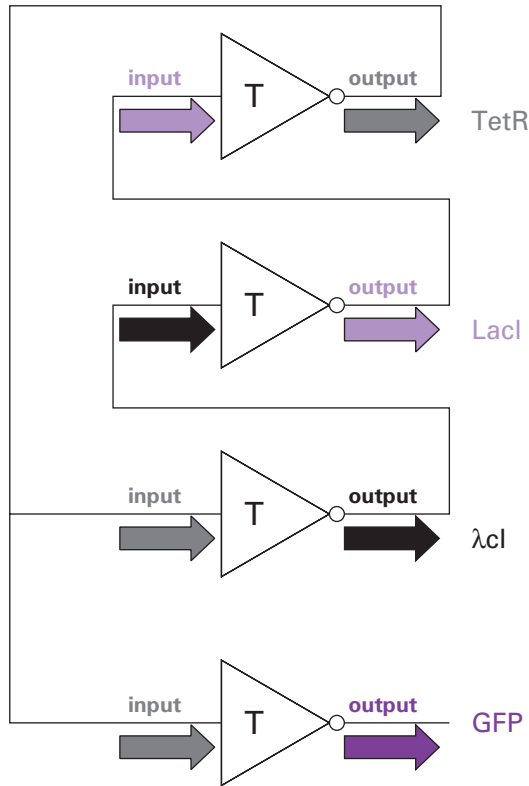
**Table M11.1 Repressilator truth table.**

Input State				Output State			
$\lambda cI$	LacI	TetR	GFP	$\lambda cI$	LacI	TetR	GFP
0	0	0	0	1	1	1	1
0	0	1	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	0	0	1	1	0	1
1	1	0	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	0	0	1	0
1	1	1	1	0	0	0	0

A truth table can represent more complex functions that have many input and output variables. For example, the repressilator can be thought of as a function with four binary input and output variables:  $\lambda cI$ , LacI, TetR, and GFP. The state of the repressilator at a particular time is represented by an ordered list of variable values; e.g., the state “1, 0, 0, 1” means that  $\lambda cI$  is present, LacI is absent, TetR is absent, and GFP is present. Because the state of the system at the current time point determines the state of the system at the next time point, we can represent the behavior of the repressilator by describing the state of the system at regular time intervals.

In our example, the time interval is the time required to transcribe and translate a single copy of each gene. We also assume that all gene products are degraded in a single time interval. Thus, for example, if the input state is “1, 0, 0, 1,” the output state will be “1, 0, 1, 1” because the input presence of  $\lambda cI$  will prevent expression of LacI over the next time interval; the input absence of LacI will enable expression of TetR over the next time interval; and the input absence of TetR will enable expression of  $\lambda cI$  and GFP over the next time interval. The truth table in Table MM11.1 summarizes input (current time point) and output (next time point) variable values for the repressilator.

To represent complex functions like the repressilator with logic gates, several gates must be combined into a circuit in which the output of one or more gates serves as input into other gates. A logic gate circuit model of the repressilator is shown in Figure MM11.4. Borrowing tools like truth tables and logic gates from the field of computer science gives us new ways to visualize complex information and helps us gain a deeper understanding of systems like the repressilator.



**Figure MM11.4** Logic gate circuit model of repressilator.

The time interval for propagating values from one variable to another is represented by the “T” in each NOT gate. The input state of the system is indicated by the labeled arrows to the left of each NOT gate. The output state of the system is indicated by the labeled arrows to the right of each NOT gate.

### MATH MINUTE DISCOVERY QUESTION

1. Why doesn't the repressilator get locked into a cycle of alternating between all 0's and all 1's?

The very regular periodicity of the repressilator helps you appreciate your own circadian rhythm and how it too can be controlled by a small number of genes that produce cyclical amounts of protein. However, there is a fly in this ointment of perfect timing. Noise and stochastic behavior of proteins are a fundamental property of gene regulation and protein production, so you might expect some problems with the regularity of the repressilator.

### DISCOVERY QUESTIONS

37. How can a biological clock outlive its host cell?
38. Graph the production of CI, LacI, and TetR proteins on top of the graph in Figure 11.27c.

39. Why did the investigators choose proteins that are rapidly degraded by cells? What would have happened if the proteins were long-lived?
40. Predict what might happen to the repressilator periodicity inside sister cells after division.
41. What would be required for sister cells to maintain the exact same periodicity of GFP production? In other words, does the repressilator have any means for cooperation through communication or checkpoints (see pages 376)?

Given the clarity of the data in Figure 11.27c, you might think humans can construct a clock that is unaffected by



**LINKS**  
Christina Smolke

the inherent noise of gene regulation. Still, a clock can only be as consistent as its component parts (Figure 11.28). We can follow one cell (purple trace in Figure 11.28c) and its sisters from two rounds of division (in gray and black) and see that although oscillation was retained in all three cells, timing and amplitude were not maintained through the generations. Interestingly, when cells stopped dividing due to nutritional limitations (**stationary phase**), the repressilator stopped working. Therefore, cell growth was required for proper functioning of the repressilator.

### DISCOVERY QUESTIONS

42. Predict what would happen if IPTG were added to the repressilator (IPTG disrupts the function of LacI).
43. Hypothesize why some sibling cells altered periodicity and others altered amplitude.

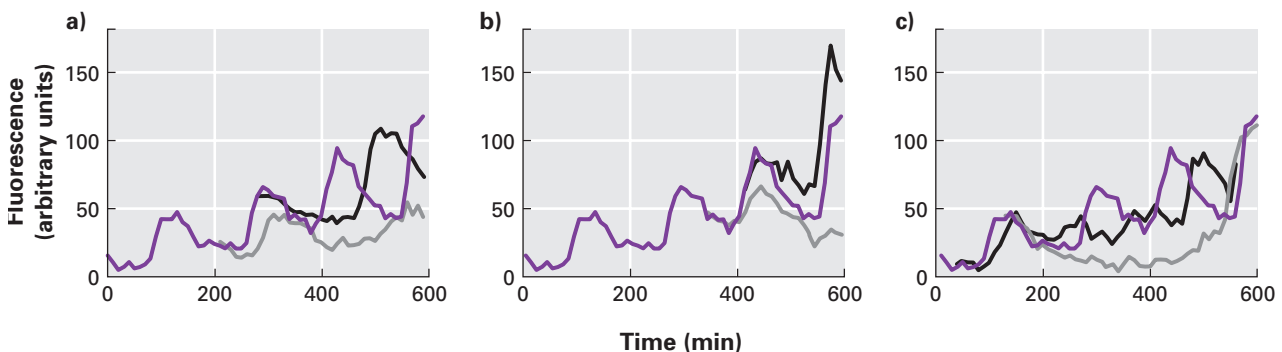
## Can Synthetic Devices Alter Gene Expression?

A chemical engineering graduate student in Christina Smolke's lab at Caltech, Travis Bayer, wanted to synthesize small devices that could predictably alter the activity of reporter genes. However, Bayer and Smolke wanted a faster response time than exhibited by the repressilator, so they chose to design RNA molecules that could respond quickly to environmental changes. They called their new molecules **antiswitches** because they were bistable toggle switches that performed part of their function by acting like antisense RNA. The term antiswitch should not be confused with another new term, **riboswitch**, which describes the 5' untranslated regions that regulate translation of the attached coding regions of some mRNAs.

An antiswitch contains an **aptamer**, a nucleic acid that can bind to a small ligand. Aptamers can be composed of either RNA or DNA. Bayer and Smolke found an RNA aptamer sequence in the literature that bound a drug called theophylline (used to treat respiratory ailments). Theophylline is structurally very similar to caffeine, but the aptamer can distinguish between the two related compounds (Figure 11.29a). Onto the aptamer, the investigators fused two different stems of bases. The short, single-stranded stem was called the aptamer stem and the longer, double-stranded one was called the antisense stem (Figure 11.29b). In the absence of theophylline, the antisense stem binds to itself and has no effect on the target mRNA. When theophylline binds to the aptamer, the entire molecule experiences a conformational change such that the antisense stem becomes single-stranded and the aptamer stem binds to a portion of the antisense stem. With this toggle switch of conformations, the single-stranded antiswitch stem is now able to bind to, and silence, the targeted mRNA.

The design looks functional, but how did the antiswitch perform *in vivo*? A plasmid encoding the antiswitch was transformed into yeast and the cells were incubated with different concentrations of theophylline (Figure 11.29c). The antiswitch functioned as predicted, with a discrete theophylline concentration for binding to the aptamer and time required for silencing the GFP signal. In subsequent experiments, Bayer and Smolke designed and built antiswitches that worked at different concentrations of theophylline, thereby demonstrating the flexibility of their devices.

Now that they could turn off GFP translation, the investigators wanted to test the specificity of their new device. Could they design and construct a system that would distinguish two different ligands to silence GFP and YFP (yellow fluorescent protein; Figure 11.30)? To test their constructs, they transformed yeast cells with two plasmids, each encoding a different antiswitch, and then exposed the cells to each

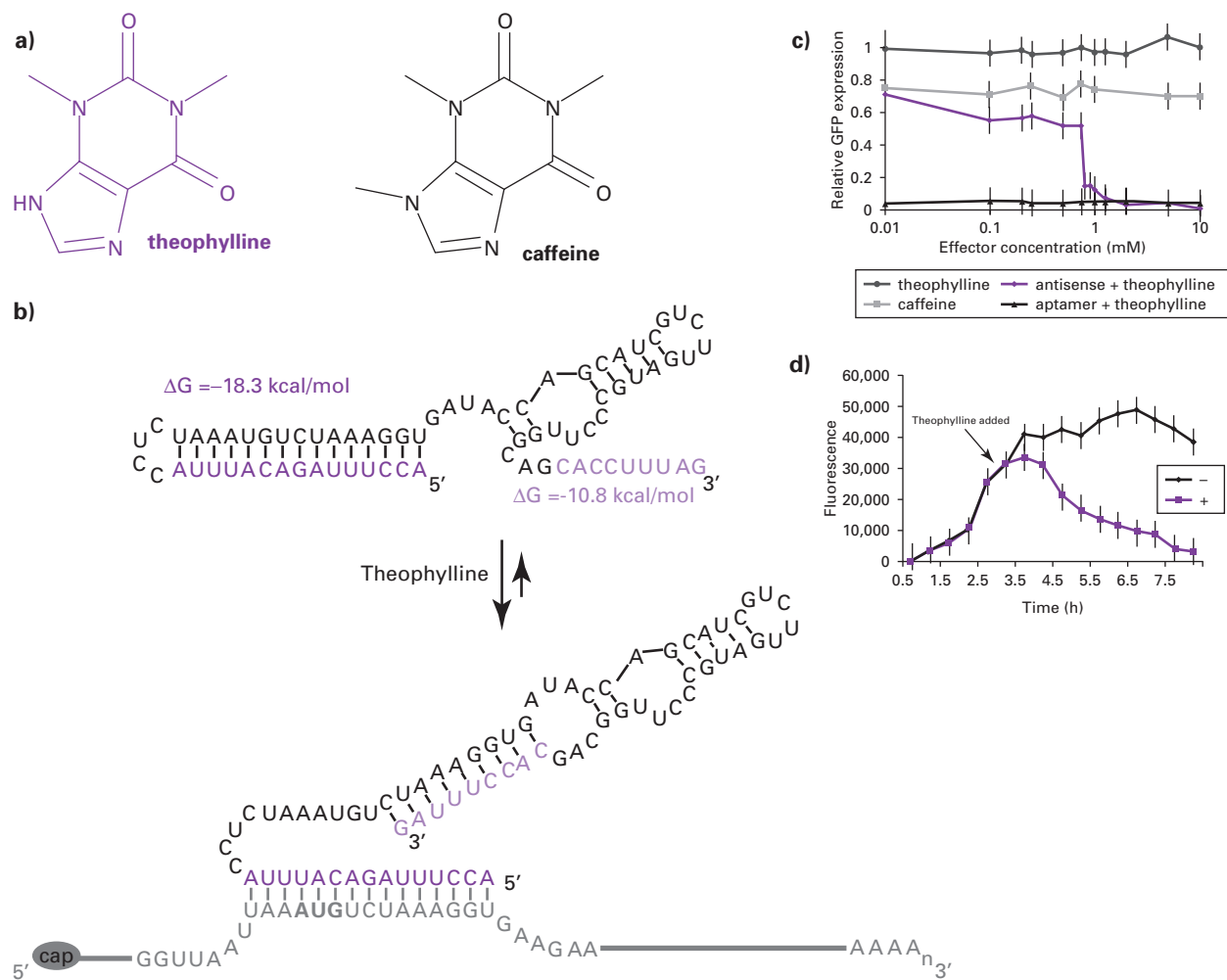


**Figure 11.28** Examples of sister cells that maintain periodicity but not synchronicity.

- a) to c)** In each case, the fluorescence time course of the cell depicted in Figure 11.27 is redrawn in purple as a reference, and two of its siblings are shown in black and gray.
- a)** Siblings exhibiting delays in the phase after cell division, relative to the reference cell.
  - b)** Phase is approximately maintained but amplitude varies significantly after division.
  - c)** Reduced period (black) and long delay (gray).

ligand separately and then in combination. Once again, the synthetic devices worked exactly as designed.

Having mastered antiswitches that turn off active genes, the investigators designed and built a new antiswitch that repressed GFP production until the cells were incubated with ligand (Figure 11.31). They tested cells that contained one or the other of the two antiswitches and, as predicted, found that the two opposing antiswitches turned on or off GFP production at about 1 mM theophylline. Given this ability to turn a gene on or off, and the ability to distinguish different concentrations of ligand, it seems that Bayer and Smolke have designed and constructed very clever devices that should prove useful in future synthetic biology research.



**Figure 11.29 Design and function of first antiswitch construct.**

- Theophylline and its close relative, caffeine, which does not bind to the aptamer.
- Sequence and conformational changes of antiswitch upon binding theophylline. Without ligand, the antiswitch antisense arm is closed and the aptamer arm is open. Theophylline binding causes the antiswitch arm to open and bind to the GFP mRNA (including start codon) while the aptamer arm closes. The stability of each arm when closed is shown as change of free energy ( $\Delta G$ ; larger negative number is more stable).
- Three control experiments demonstrate the function meets the design expectations.
- Two flasks of identical cells produce GFP, but one is incubated with theophylline to block further GFP production. All antiswitch experiments were performed in yeast cells.

## DISCOVERY QUESTIONS

- Explain why antiswitches might react faster than protein-based mechanisms for gene silencing.
- Riboswitches, aptamers, and antiswitches are all new terms. Try a search on PubMed to see how many citations include these terms. It would be nice to know what aptamers are available. Try searching [RNABase.org](http://RNABase.org) using the word “aptamer.” Then try searching [Aptamer Database](http://Aptamer Database) to find particular ligands. Can you find the two used in the antiswitch case study? How could this database be improved?

**LINKS**   
Aptamer Database  
RNABase.org



examine fitness is to see what has evolved over millions of years of selective pressure. Imagine some original cell on earth, some tiny robust prokaryote that was capable of surviving mutations to its own genome (RNA or DNA) and giving rise to a second species on earth. These two species continued to diversify as new genes evolved, became duplicated, modified, duplicated again, etc. All this genetic experimentation led to new arrangements of genes. For example, species 4 might have genes A, B, C, D, E, F, G, H, I, J, whereas species 5 evolved with a gene order of A, H, G, F, E, D, C, B, B', I, J. At some point during this time of species diversification, sex evolved and the order of genes became scrambled even more with chromosomal rearrangements.


### DISCOVERY QUESTIONS

47. What kind of changes occurred to the DNA of species 5 compared to 4 (see preceding text)? What new opportunities does species 5 have by virtue of its two copies of gene B?
48. What would you predict about the degree of reliability (see pages 377–378) of the process encoded by genes B and B'?
49. The observational approach has limitations based on available genomes. Can you imagine a way to experimentally test the effects of altering gene order within a genome?

Evolutionary biologists assume that what works best must be what is still in existence today. Some look at morphology, some at behavior, and others at DNA sequences. As a result of genome comparisons, investigators have identified many gene complexes that are highly ordered. The most famous are the *Hox* genes, which are conserved not only in DNA sequence but also in gene order on chromosomes. DNA sequence and gene order are conserved from flies to humans, with genes arranged from head to tail along the length of the chromosomes in humans, mice, worms, and flies. To molecular biologists, however, there is something unsatisfying about this kind of research: it leads to correlations that have not been experimentally tested. Every molecular biologist accepts that the arrangement of *Hox* genes is evolutionarily advantageous, but it is impossible to test this by altering history.

Recent work by many researchers who analyze DNA sequences has produced some interesting observations. Elizabeth Williams and Laurence Hurst found that linked genes often evolve at similar rates. The coevolution of linked genes might seem obvious, but remember that after a few billion years of recombination and mutation, the probability that two genes would remain as neighbors and mutate at similar rates is very small. The caveat about this type of analysis is that you never can be sure if the

conserved gene order is due to chance or selective pressures. To address this concern, a group from the European Molecular Biology Laboratory (EMBL) in Germany analyzed the genomic sequences of nine different taxa of prokaryotes (proteobacteria, Gram-positive bacteria, and Archaea). They found, in many cases, that gene pairs were conserved in their order on chromosomes across diverse taxa. Clusters of two to seven genes were conserved in order and orientation for transcription. It is hard to imagine that chance alone could explain conserved gene order and orientation. Nevertheless, this observation is a correlation and not a causative analysis, and thus is not completely satisfying.

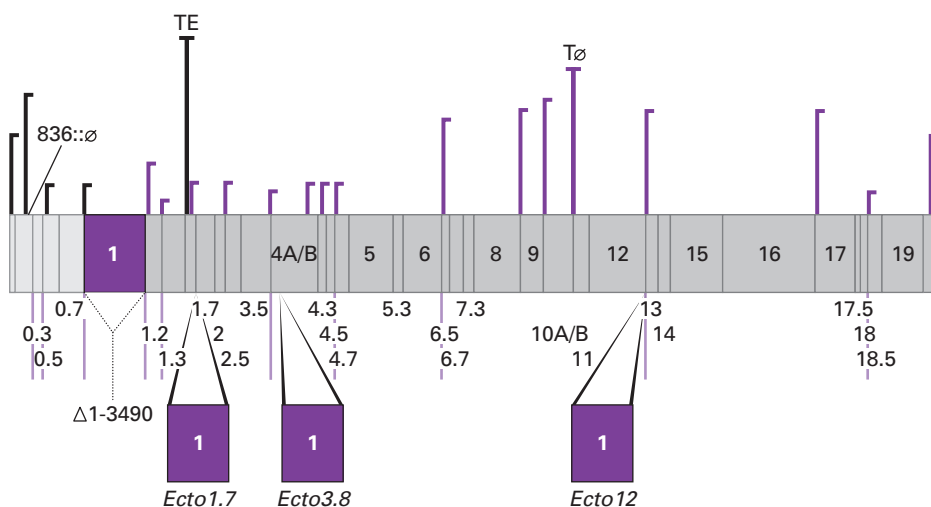
**LINKS**   
Drew Endy  
Elizabeth Williams and  
Laurence Hurst  
EMBL

### Computational Approach

There is a second approach that is more satisfying to the growing number of biologists exploring genomes, though this approach also has weaknesses. If we know all the sequences of the genes in a genome, and we know the roles for most of these genes, then why not build a computer simulation, make predictions, test them, refine the model, etc.? This computational approach to evaluating gene order for entire genomes is very new. One of the pioneers in this area is Drew Endy, currently working at MIT, and his whole genome approach to evolution is a new trend in biology.

Endy approached this experimentally difficult question of gene order by choosing a simple system (i.e., a model organism). Rather than selecting a species like yeast, which has about 6,200 genes, he chose to start smaller, with a viral genome of only 56 genes. Endy chose the virus T7 as his model organism (Figure 11.32). T7's double-stranded DNA genome is only 39,937 bp long. This is so small that it is hard to imagine the genome includes much complexity, but even this simple organism has some hidden secrets. Of the 59 proteins produced by T7's 56 genes, only 33 have known functions. That could present some problems with a simulated circuit, but computer models are not intended to be perfect; rather, they should incorporate current knowledge and allow predictions that can be tested experimentally to improve our understanding. Clearly, there is room for improvement when only 56% of the proteins have known functions.

Twenty-five minutes after T7 infects an *E. coli* bacterium, about 100 T7 progeny will be released when the host cell lyses. Unlike viruses that inject their genomes quickly, it takes about 10 minutes for the T7 genome to enter its host. The first 850 bp are inserted by the virus, and the rest is pulled into the host as it is being transcribed. Initially, the *E. coli* RNA polymerase binds to one or more of the 5 *E. coli*-recognized promoters in the T7 genome (spanning the first 15% of the genome in Figure 11.32). *E. coli* RNA polymerase pulls the T7 genome into the cell at about 45 bp



**Figure 11.32** Wild-type T7 ( $T7^+$ ) genome contains 56 genes encoding 59 proteins.

Numbers represent coding regions (genes were numbered as space permitted). Vertical lines with half-bars represent *E. coli* (black) and T7 (purple) promoters, with bar height proportional to the strength of each promoter. *E. coli* RNAP (polymerase; TE) and T7 RNAP ( $T\emptyset$ ) terminators are shown as vertical lines with full bars. RNase III recognition sites are shown as vertical lines below the genome. All transcription moves from left to right. The positions of the *gene 1* (encoding the T7 RNA polymerase) in  $T7^+$  and in the three experimental strains constructed and characterized in the laboratory are shown below the genome and labeled *Ecto1.7*, *Ecto3.8*, and *Ecto12*; *wt gene 1* was deleted in the three Ecto-mutants.

per second. The first essential T7 gene to produce a protein is conveniently called *gene 1*. *gene 1* encodes the T7 RNA polymerase, which takes over the role of transcription and genome movement into *E. coli*. T7 RNA polymerase uses 17 different promoter sequences (in the remaining 85% of the genome) and pulls the DNA in at a rate of 200 bp per second—a fourfold increase in genome movement. Therefore, it is easy to see why T7 would evolve to have *gene 1* as the first coding sequence in its genome.

### Does Gene 1 Have to Be First?

T7 is a clear example that gene order might matter even for a small genome. How much flexibility is there for *gene 1* position within the genome? Could it be second? Third? If the promoters were moved around, would the change alter T7's evolutionary outcome? Endy started his computer simulation with the question about the optimum position for *gene 1* (Figure 11.32). Using *in silico* methods, Endy placed *gene 1* in every possible position (except as very first and very last) and calculated how long it would take T7 to double its numbers in a simulated flask of *E. coli*. He used this formula to determine the growth rate of T7:

$$\text{maximum doubling rate} = \mu_m = \max_t \{ \log_2 [Y(t)] / t \}$$

where  $Y(t)$  is the computed number of intracellular progeny as a function of time ( $t$ ).

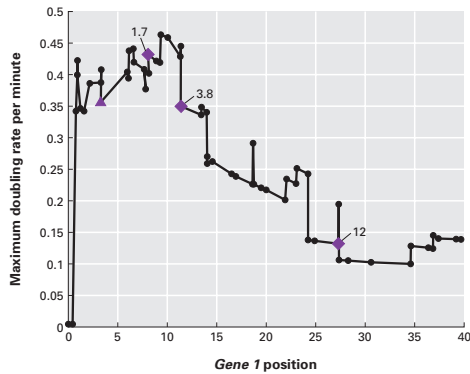
The term  $\mu_m$  defines the optimal time for phage-induced lysis in environments containing infinite uninfected hosts. For each of the 72 *in silico gene 1* positional mutants,  $\mu_m$  was calculated and graphed. When he ran his simulation, Endy discovered some genotypes were better than the simulated *wt* strain (see Figure 11.33). The *wt* genotype (triangle) has a doubling rate of about 0.35 per minute, but there are several genotypes with higher rates—some as much as 30% higher. This simulated improvement on evolution flies in the face of evolutionary theory. Surely the  $T7^+$  genome is not suboptimal in its evolutionary fitness! But remember, this was a computer simulation. How did Endy's simulations compare with reality?

Three **ectopic** (not in their normal genomic location) strains, where copies of *gene 1* were placed into new locations, were generated in the lab. *Ecto1.7* had *gene 1* inserted into a nonessential gene called *1.7*. *Ecto3.8* had *gene 1* inserted into the coding region of the nonessential *gene 3.8*. In *Ecto12*, *gene 1* was inserted between a promoter and *gene 12*. Endy also constructed two control viruses which had a *gene 1*-sized piece of  $\lambda$  phage DNA inserted into genes *1.7* and *3.8*, and a third control strain in which a T7 late promoter was inserted at base 836.

### DISCOVERY QUESTIONS

- 50.** Think of reasons why the controls Endy used were not ideal. Design a different control for these experiments.

51. It is difficult to understand what would make a mutant strain more efficient, but easier to understand less efficient mutants. Explain why the mutations with *gene 1* positioned late in the genome would be the least efficient.



**Figure 11.33** Predicted consequences of altered *gene 1* position.

Computed maximum phage doubling rate,  $\mu_m$ , as *gene 1* was repositioned on the T7<sup>+</sup> genome. Black circles mark the 5' end of the in silico ectopic (inserted) *gene 1* and the  $\mu_m$  for 72 positional mutants. The purple diamonds indicate the computed  $\mu_m$  for the ectopic *gene 1* strains that were constructed and characterized experimentally. The purple triangle indicates the  $\mu_m$  computed for T7<sup>+</sup>.

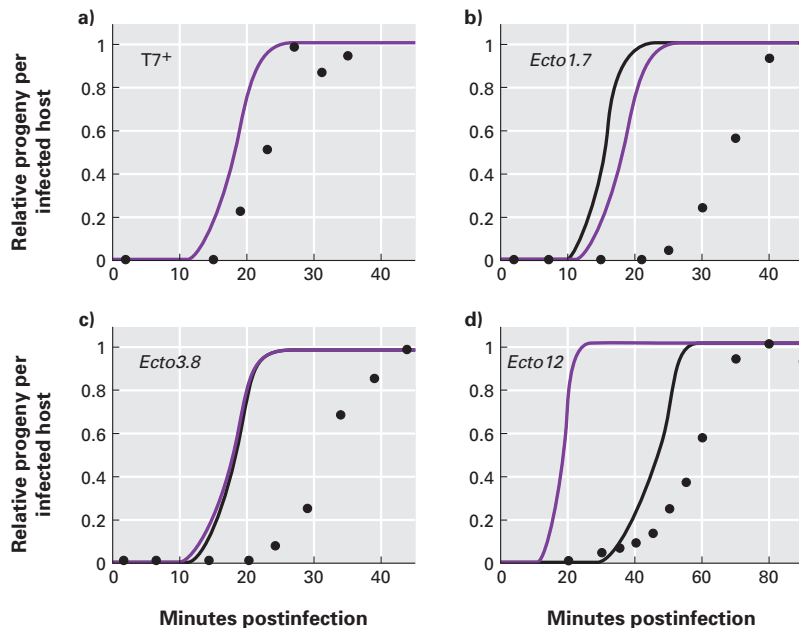
52. In the T7 computer simulation, identical viruses were programmed to behave identically. In reality, would you expect each virus to perform exactly the same? Do you think stochastic events might occur in natural settings? If so, what effect might randomness have on the difference between computer growth curves and real growth curves?

**DATA**  
T7 simulation



Endy compared the simulated viruses to his three genetically engineered mutant strains of T7. Each real strain was added to a separate population of *E. coli* in a flask, and the real doubling rates for these T7 mutants were experimentally measured (Figure 11.34). The simulated growth curve for T7<sup>+</sup> was four minutes faster than the experimentally determined growth curve for T7<sup>+</sup>. Compare the slopes of computer growth rates and the real growth rates for T7, and you will see they are very similar. The simulated growth curve for *Ecto12* was ten minutes faster than the experimentally determined curve. Live *Ecto1.7* and *Ecto3.8* strains showed little agreement with predicted growth rates.

As part of his research, Endy simulated and measured plaque sizes, DNA replication, protein production, and lysis rates for his three real mutants. Go to the web site to view the [T7 simulation](#) (with a link to the 80-MB movie) that shows the rate of transcription and translation for every gene in the T7 genome. It also shows the *E. coli* and



**Figure 11.34** Computed (solid lines) and observed (black dots) intracellular one-step growth curves for T7<sup>+</sup> (*wt*) and the ectopic *gene 1* strains.

Note the different time scale for *Ecto12*. **a)** *wt*, **b)** *Ecto1.7*, **c)** *Ecto3.8*, and **d)** *Ecto12* are shown (black), with T7<sup>+</sup> growth curve (purple line) for reference.



**DATA**  
BioBricks Registry  
T7 genes

T7 RNA polymerases in action. All this work indicated that Endy was successful in the scientific sense. He constructed a model,

made predictions, tested them, and now he will refine his model based on new experimental evidence.

### DISCOVERY QUESTIONS

53. Which part of the real curve differs the most from the simulated curve (see Figure 11.34)? Explain why this part of each curve might be so different, given what you know about stochastic protein kinetics and noisy toggle switches.
54. Reevaluate the data in Figure 11.33 after analyzing the experimental data from Figure 11.34. What can you say about the optimum gene order for T7?
55. In the **T7 simulation**, which is produced in greater quantities, mRNA or proteins? By the end of infection, which genes are most highly transcribed? Which proteins are most abundant? How many copies are present inside the infected cell for the most abundant protein? At the end of the infection, the earliest **T7 genes** are no longer transcribed. Explain why this may be biologically adaptive and why this may indicate why the order of T7 genes is functionally significant.
56. Can the T7 RNA polymerase transcribe all the genes, or only the last 85%? In this simulation, what happens when a T7 RNA polymerase catches an *E. coli* RNA polymerase from behind? Do you think this is biologically accurate? Explain your answer.
57. Initially, what two proteins are the only ones present? What happens to each of them during the infection?

Endy's research illustrates how difficult it will be to understand more complex genomes. Part of the problem with T7<sup>+</sup> is that some genes overlap each other, which makes it impossible to mutate one gene but not the other. To address this challenge, Leon Chan and Sriram Kosuri generated a synthetic T7 (called T7.1) in which every gene is completely separated from its neighbors. T7.1 sounds like a good idea that is bound to fail because synthetic biologists cannot fully understand the selection pressures that have driven maximization of fitness. However, T7.1 was viable and produced viable progeny. The plaque size and morphology were similar to T7<sup>+</sup>, but the time to lysis was delayed for T7.1. As with computer models designed to approach biological reality, T7.1 is not perfect, but it does improve our previous understanding of the naturally evolved T7<sup>+</sup>. With T7.1, we will be able to dissect each component of the genome to understand its function and later build an improved model of T7 that will test our

newest insights. Over several iterations, T7.*n* will lead to better predictions and a better understanding of genetic toggle switches and synthetic circuits.

### DISCOVERY QUESTIONS

58. You may not realize it, but you can conduct synthetic biology research too. Go to the MIT **BioBricks Registry** to see a community-based effort to understand biological circuits and switches. The concept behind BioBricks is that DNA parts could be interchangeable, similar to Legos. If you could snap one part onto any other part, you could design and construct new devices and systems to test your understanding of how genomes perform some of their functions. Try these tasks to learn your way around.
- Click on “Parts, Devices & Systems” to learn the vocabulary, and “About Parts” in the left frame to learn what is in the BioBrick Registry.
  - From the main page, click on the “reporter” icon to see a full list of reporter components. Find GFP and GFP-AAV. What is the difference between these two components?
  - From the main page, enter “BBa\_M0044” in the View Part search box on the bottom left side. What is AAV, and what information would you like to know that is missing?
  - Click on the “Search” link on the left side, then click on the word “Example” above the middle search box. The system you have searched for is composed of four devices. Below it are the intermediate construction phases that are available in the BioBricks freezer. Mouse over the icons and see if you can understand the four devices, as well as the intended function of the overall system. To test your prediction, enter the term “BBa\_I13907” into the search box and read the description.
59. Information in the Registry may be read by anyone. You can sign up as a guest member and assemble parts in the “sandbox” area of the Registry. Your part numbers have to be in the range designated for nonregistered visitors. You can create parts, but they will not be maintained on the database as a part of the full collection. However, if you find this process interesting, you might want to join the iGEM competition next year to conduct real research in synthetic biology. Although the field is cutting edge, the equipment requirements are modest.

## Summary 11.2

By applying engineering principles to toggle-switch design, we can construct biologically functional genetic switches and experimentally test their reliability. It might seem uncomfortable to treat genomes like machines with interchangeable parts, but by designing and building synthetic devices and systems, we can test our understanding more completely. Whether we assemble three-gene circuits, produce antiswitches to regulate the expression of proteins, or reengineer a model genome as a way of teasing apart the function of component parts, we are always trying to understand biology and evolution more completely. Eventually, synthetic biology may lead to new medical treatments, but for now, research using designed DNA parts is proving to be challenging and insightful.

## Chapter 11 Conclusions

Understanding a genome requires more than just listing the component genes and defining their roles. We need to understand how the proteins interact, how genes regulate each other's transcription, and how positive and negative feedback loops produce synergistic properties that affect a cell's response to its changing needs. Toggle switches enable genomes to make "choices," and the inherent noise in toggle switches is a necessary part of how they function. Proteins exhibit stochastic behavior, so it may be impossible to produce a model that can predict with certainty how a particular cell will respond to a change in its environment. Complex circuits have unique properties, and as we construct increasingly complex circuit diagrams, we will discover more emergent properties that improve our understanding of cells and organisms. A cell web is composed of many interconnected components, and the ultimate challenge is to model the combined information of genome sequences, variations in the population, gene expression profiles, proteomics, metabolomics, and genomic circuits. These challenges are intimidating but exciting, too. In Chapter 12, we will use one case study to see what is possible now and where we want to go in the future.

### References

Bayer, T. S., & C. D. Smolke. 2005. Programmable ligand-controlled riboregulators of eukaryotic gene expression. *Nature Biotechnology*. 23(3): 337–343.

Bhalla, U. S., & R. Iyengar. 1999. Emergent properties of networks of biological signaling pathways. *Science*. 283: 381–387.

Bhalla, U. S., P. T. Ram, & R. Iyengar. 2002. MAP kinase phosphatase as a locus of flexibility in a mitogen-activated protein kinase signaling network. *Science*. 297: 1018–1023.

Bonifer, C. 2000. Developmental regulation of eukaryotic gene loci. *Trends in Genetics*. 16: 310–315.

Chan, L. Y., Sriram, K., & D. Endy. 2005. Refactoring bacteriophage T7. *Molecular Systems Biology* 64–73 doi: 10.1038/msb4100025.

Cho, R. J., & M. J. Campbell. 2000. Transcription, genomes, function. *Trends in Genetics*. 16: 409–415.

Clayton, D. F. 2000. The genomic action potential. *Neurobiology of Learning and Memory*. 74: 185–216.

Dandekar, T., M. B. Snel, & P. Bork. 1991. Conservation of gene order: A fingerprint of proteins that physically interact. *Trends in Biochemical Sciences*. 23: 324–328.

Edwards, J. S., & B. O. Palsson. 2000. The *Escherichia coli* MG1655 in silico metabolic genotype: Its definition, characteristics, and capabilities. *PNAS*. 97: 5528–5533.

Edwards, J. S., & B. O. Palsson. 2000. Metabolic flux balance analysis and the in silico analysis of *Escherichia coli* K-12 gene deletions. *BioMed Central Bioinformatics*. 1(1): 1. <<http://biomedcentral.com/1471-2105/1/1>>

Elowitz, M. B., & S. Leibler. 2000. A synthetic oscillatory network of transcriptional regulators. *Nature*. 403: 335–338.

Endy, D., & R. Brent. 2001. Modelling cellular behavior. *Nature*. 409: 391–395.

Endy, D., L. You, et al. 2000. Computation, prediction, and experimental tests of fitness for bacteriophage T7 mutants with permuted genomes. *PNAS*. 97: 5375–5380.

Frankland, P. W., C. O'Brien, et al. 2001.  $\alpha$ -CaMKII-dependent plasticity in the cortex is required for permanent memory. *Nature*. 411: 309–313.

Gardner, T. S., C. R. Cantor, & J. J. Collins. 2000. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*. 403: 339–342.

Genoux, D., U. Haditsch, et al. 2002. Protein phosphatase 1 is a molecular constraint on learning and memory. *Nature*. 418: 970–975.

Gladwell, M. 2000. *The tipping point: How little things can make a big difference*. Boston: Little, Brown.

Hartwell, L. H., J. J. Hopfield, et al. 1999. From molecular to modular cell biology. *Nature*. 402 Supplement: C47–C52.

Hasty, J., J. Pradines, et al. 2000. Noise-based switches and amplifiers for gene expression. *PNAS*. 97: 2075–2080.

Hurst, L. D., Williams, E. J. B., & Pál, C. 2002. Natural selection promotes the conservation of linkage of co-expressed genes. *Trends in Genetics*. 18: 604–606.

Jeong, H. B., R. Tombor, et al. 2000. The large-scale organization of metabolic networks. *Nature*. 407: 651–654.

Kelley, B. P., R. Sharan, et al. 2003. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *PNAS*. 100(20): 11394–11399.

Kitami, T., & J. H. Nadeau. 2002. Biochemical networking contributes more to genetic buffering in human and mouse metabolic pathways than does gene duplication. *Nature Genetics*. 32: 191–194.

Kohn, K. 1999. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molecular Biology of the Cell*. 10: 2703–2734.

Kuang, Y., I. Biran, & D. R. Walt. 2004. Simultaneously Monitoring Gene Expression Kinetics and Genetic Noise in Single Cells by Optical Well Arrays. *Analytical Chemistry*. 76: 6282–6286.

Lee, J. F., J. R. Hesselberth, et al. 2004. Aptamer database. *Nucleic Acids Research*. 32(1): D95–100.

Legrain, P., J.-L. Jestin, & V. Schächter. 2000. From the analysis of protein complexes to proteome-wide linkage maps. *Current Opinion in Biotechnology*. 11: 402–407.

Lercher, M. J., J.-V. Chamary, & L. D. Hurst. 2004. Genomic regionality in rates of evolution is not explained by clustering of

- genes of comparable expression profile. *Genome Research*. 14: 1002–1013.
- McAdams, H., & A. Arkin. 1999. It's a noisy business! Genetic regulation at the nanomolar scale. *Trends in Genetics*. 15: 65–69.
- McAdams, H., & A. Arkin. 1997. Stochastic mechanisms in gene expression. *PNAS*. 94: 814–819.
- McAdams, H. H., & L. Shapiro. 1995. Circuit simulation of genetic networks. *Science*. 269: 650–656.
- Miller, P., A. M. Zhabotinsky, et al. 2005. The stability of a stochastic CaMKII switch: Dependence on the number of enzyme molecules and protein turnover. *PLoS Biology*. 3(4): 0705–0717.
- Milo, R., S. Shen-Orr, et al. 2002. Network motifs: Simple building blocks of complex networks. *Science*. 298: 824–827.
- Murthy, V. L. 2005. The RNA structure database. <[www.RNABase.org](http://www.RNABase.org)>. Accessed 20 May 2005.
- Pirson, I., N. Fortemaison, et al. 2000. The visual display of regulatory information and networks. *Trends in Cell Biology*. 10: 404–408.
- Ptashne, M. 1987. *A genetic switch: Gene control and phage  $\lambda$* . Cambridge, MA: Cell Press & Blackwell Scientific Publications.
- Rzhetsky, A., T. Koike, et al. 2000. A knowledge model for analysis and simulation of regulatory networks. *Bioinformatics*. 16(12): 1120–1128.
- Suyama, M., & P. Bork. 2001. Evolution of prokaryotic gene order: genome rearrangements in closely related species. *Trends in Genetics*. 17(1): 10–13.
- Weng, G., U. S. Bhalla, & R. Iyengar. 1999. Complexity in biological signaling systems. *Science*. 284: 92–96.
- Wray, G. A. 1991. Promoter logic. *Science*. 279: 1871–1872.