
Natural Computing **2:** 47–84, 2003.

Genetic circuit building blocks for cellular computation, communications, and signal processing

RON WEISS*, SUBHAYU BASU, SARA HOOSHANGI, ABIGAIL KALMBACH, DAVID KARIG, RISHABH MEHREJA and ILKA NETRAVALI

*Electrical Engineering Department, Princeton University, Engineering Quad, B312, Princeton, NJ 08540, USA; *Author for correspondence, e-mail: rweiss@princeton.edu*

Abstract. In this paper, we review an emerging engineering discipline to program cell behaviors by embedding synthetic gene networks that perform computation, communications, and signal processing. To accomplish this goal, we begin with a genetic component library and a biocircuit design methodology for assembling these components into compound circuits. The main challenge in biocircuit design lies in selecting well-matched genetic components that when coupled, reliably produce the desired behavior. We use simulation tools to guide circuit design, a process that consists of selecting the appropriate components and genetically modifying existing components until the desired behavior is achieved. In addition to such rational design, we also employ *directed evolution* to optimize genetic circuit behavior. Building on Nature's fundamental principle of evolution, this unique process directs cells to mutate their own DNA until they find gene network configurations that exhibit the desired system characteristics. The integration of all the above capabilities in future synthetic gene networks will enable cells to perform sophisticated *digital* and *analog* computation, both as individual entities and as part of larger cell communities. This engineering discipline and its associated tools will advance the capabilities of genetic engineering, and allow us to harness cells for a myriad of applications not previously achievable.

Key words: cellular computation, cell-cell communications, directed evolution, genetic signal processing, synthetic gene networks

1. Introduction

Genetic engineering with recombinant DNA is a powerful and widespread technology that enables biologists to redesign life forms by modifying or extending their DNA. Advances in this domain allow us to gain insight into the operating principles that govern living organisms, and can also be applied to a variety of fields including human therapeutics, synthesis of pharmaceutical products, molecular fabrication of biomaterials, crops and livestock engineering, and toxin detection with biological sentinels. While already providing great benefits, existing genetic engineering applications only hint at the possibilities for harnessing cells to our benefit.

For the purpose of genetic engineering, it is currently possible to construct DNA fragments that consist of almost any gene sequence. While the synthesis process is well developed, the behavior of the resulting genetic constructs is not easy to predict. Consequently, it is often difficult to design constructs that achieve desired behaviors with reliability and robustness. For example, we would like to instruct cells to reliably make logic decisions based on factors such as surrounding environmental conditions and internal cell state. In this paper, we explore the nascent field of *synthetic gene networks*, whose primary goal is to obtain sophisticated and reliable cell behaviors.

We strive to develop an engineering methodology for creating synthetic gene networks that will allow us to engineer cells with the same ease and capability with which we currently program computers and robots. The first step in making programmed cell behavior a practical and useful engineering discipline is to assemble a component library of genetic circuit building blocks. These building blocks perform computation and communications using DNA-binding proteins, small inducer molecules that interact with these proteins, and segments of DNA that regulate the expression of these proteins. We describe a component library of cellular gates that implement several digital logic functions. These include components for intracellular computation (i.e., NOT and NAND) and devices for external communication (i.e., IMPLIES and AND). The building blocks have already been assembled into several prototype genetic circuits in *Escherichia coli* bacterial cells, with up to three logic gates per cell. These genetic elements can also be configured to process environmental and internal biochemical *analog* signals. The integration of the above capabilities in future synthetic gene networks will enable cells to perform sophisticated *digital* and *analog* computation, both as individual entities and as part of larger cell communities.

An integral part of genetic circuit design is modeling the behavior of proposed circuits prior to their synthesis. In silicon chip fabrication, simulation tools help guide chip design to minimize the time and effort required for circuit construction. Similarly, genetic circuit development can also benefit from modeling tools that can predict characteristics such as the steady state and dynamic behavior of the proposed system. The simulation tools are used to evaluate various network configurations for achieving particular functions, and to help refine existing and proposed designs. A unique and powerful capability of biological circuit engineering, as compared to its silicon chip counterpart, is our ability to exploit one of Mother Nature's fundamental operating principles – the process of evolution. Specifically, through *directed evolution*, cells can be engineered to mutate their DNA sequences with the goal of optimizing circuit characteristics such as signal sensitivities and switching thresholds.

In this paper, we describe the genetic circuit building blocks (Section 2), existing prototype circuits (Section 3), genetic circuit design (Section 4), cell-cell communication (Section 5), and signal processing circuits (Section 6).

2. Genetic circuit building blocks

The first step in programming cells and controlling their behavior is to establish a library of well-defined components that serve as the building blocks of more complex systems. This section describes how certain genetic elements can be configured to implement logic gates and logic circuits. Here, instead of electrical signals representing streams of binary ones and zeros, the chemical concentrations of specific DNA-binding proteins and inducer molecules act as the input and output signals of the genetic logic gates. Within the cell, these molecules interact with other proteins, bind specific DNA sites, and ultimately regulate the expression of other proteins. This regulatory activity can be used to implement digital logic functions, as well as analog processing of signals.

2.1 *Biochemical inverter*

The characteristics of natural gene regulation systems can be exploited to design *in vivo* logic circuits (Weiss et al., 1999). An example of such a system, that illustrates the biochemical process of inversion is seen in Figure 1. Here, the presence or absence of input messenger RNA (mRNA) determines the two possible output states. A more complete picture that explains the role of transcription and translation cellular processes in inversion is seen in Figure 2.

Figure 3 depicts a functional model of the inverter derived from its biochemical reaction phases. The first phase in inversion is the translation stage, denoted as $\mathcal{L}$. The input signal to this stage, and thus the inverter, corresponds to the concentration level of the input mRNA, ϕ_A. Ribosomal RNA (rRNA) translates the input mRNA into the input repressor protein, ψ_A, where $\mathcal{L}$ represents the steady state mapping between the mRNA and protein concentrations. The relationship between the input mRNA and repressor protein is initially linear, with increases in ϕ_A corresponding to increases in ψ_A, until an asymptotic boundary is reached. The properties of this boundary are determined by characteristics of the cell such as amino acid synthesis capabilities, the efficiency of the ribosome-binding site, and mRNA stability. Since cells degrade mRNA as well as protein molecules, constant synthesis of the input mRNA is needed to maintain a steady level of the input repressor protein.

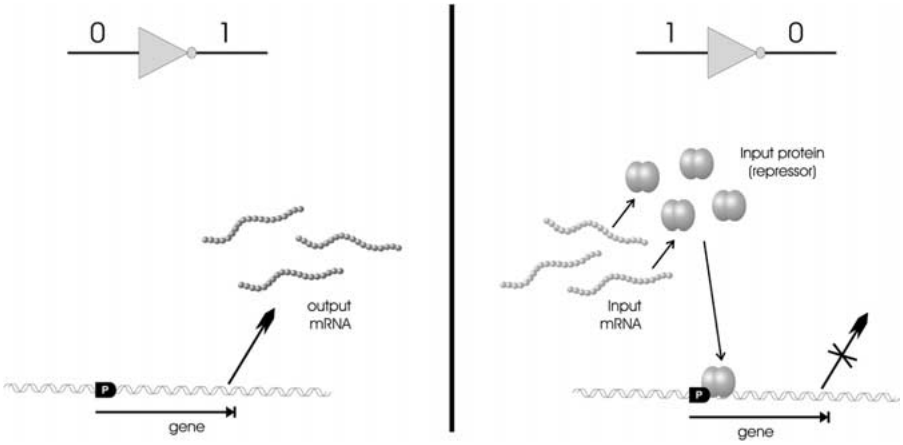


Figure 1. A simplified view of the two cases for a biochemical inverter. Here, the concentration of a particular messenger RNA (mRNA) molecule represents a logic signal. In the first case, the input mRNA is absent and the cell transcribes the gene for the output mRNA. In the second case, the input mRNA is present and the cell translates the input mRNA into the input protein. The input protein then binds specifically to the gene at the promoter site (labeled "P") and prevents the cell from synthesizing the output mRNA.

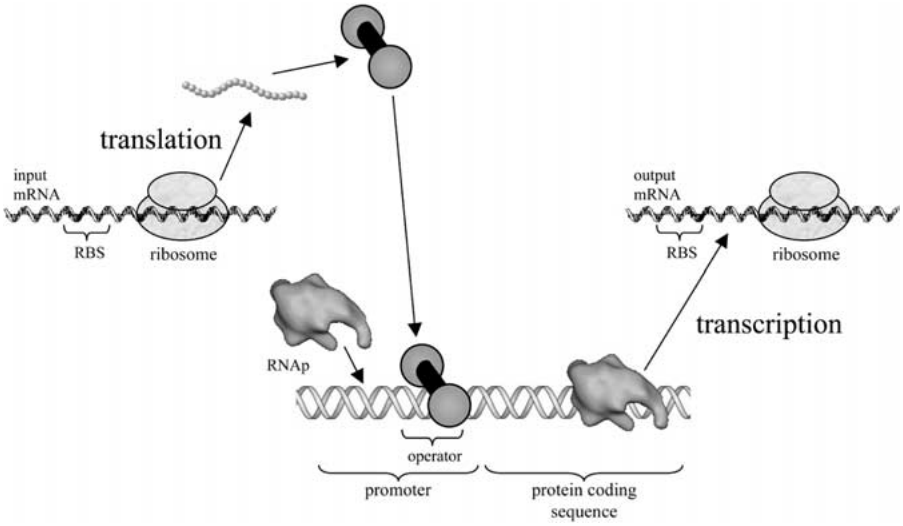


Figure 2. Biochemical inversion uses the transcription and translation cellular processes. Ribosomal RNA translates the input mRNA into an amino acid chain, which then folds into a three-dimensional protein structure. When the protein binds an operator of the gene's promoter, it prevents transcription of the gene by RNA polymerase (RNAp). In the absence of the repressor protein, RNAp transcribes the gene into the output mRNA.

In the second phase, input protein monomers combine to form polymers that bind the operator, and subsequently repress the transcription of the output

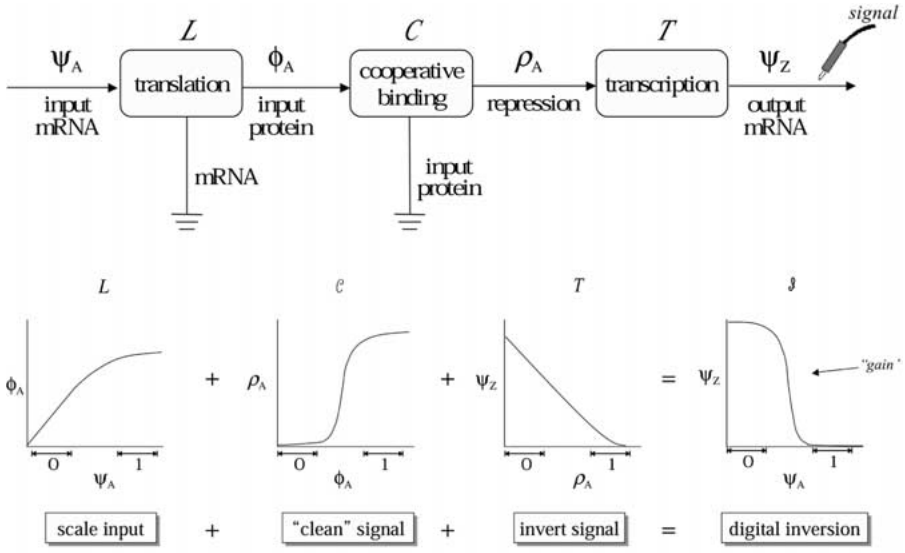


Figure 3. Functional composition of the inversion stages: the translation stage maps input mRNA levels (ψ_A) to input protein levels (ϕ_A), the cooperative binding stage maps input protein levels to bound operator levels (ρ_A), and the transcription stage maps bound operator levels to output mRNA levels (ψ_Z). The degradation of the mRNA and protein molecules is represented with the electrical ground symbol. The degradation of mRNA is part of the translation stage, while the degradation of proteins is part of the cooperative binding stage. The graphs illustrate the steady-state relationships for each of these stages and the overall inversion function that results from combining these stages.

gene. This cooperative binding, which ensures that only dimerized proteins can bind the DNA, decreases the digital noise. Let us define the concentration of operator that is bound to the repressor, or the strength of the repression, as ρ_A. In addition, denote the cooperative binding stage that occurs between ψ_A and ρ_A as $\mathcal{C}$. In steady state, the relationship between ψ_A and ρ_A is sigmoidal. At low levels of ψ_A, the strength of repression does not increase significantly for increases in ρ_A because these concentrations are too low for appreciable dimerization. At higher concentrations of ψ_A, however, considerable dimerization occurs, resulting in a nonlinear increase in repression activity. For values of ψ_A approaching saturation, the operator is mostly bound, and repressor activity is close to maximal. At this point, increasing the concentration of ψ_A does not increase repression, and instead causes the ψ_A/ρ_A curve to move toward an asymptotic boundary. In this way, the cooperative binding stage performs signal restoration in which the analog output signal better represents the appropriate digital meaning than the corresponding analog input signal. Because each stage of the computation reduces the noise in the

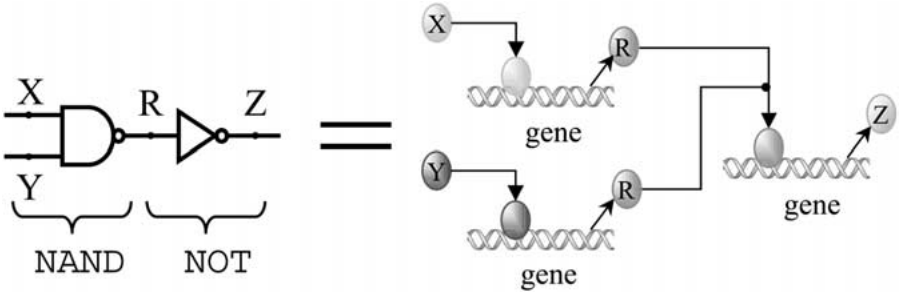


Figure 4. Design of a biochemical NAND logic gate connected to a downstream inverter. The two-input NAND gate consists of two separate inverters, each with a different input, but both with the same output protein. The NAND gate output is always HIGH unless both inputs are present. This output can then be connected to other downstream gates, such as an inverter.

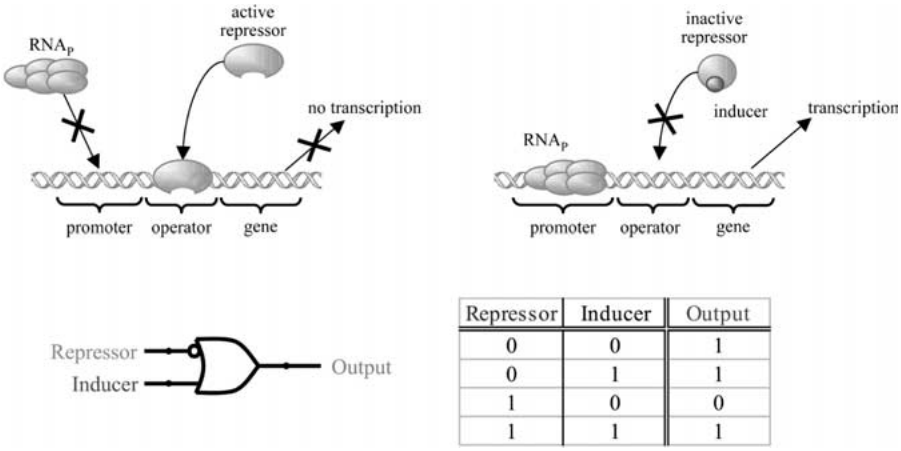
system through signal restoration, multiple inverters can be combined into more complex circuits, while still maintaining or even increasing the overall reliability of the system.

In the final stage of the inverter, the transcription stage, RNA polymerase (RNAp) transcribes the regulated gene and the input signal is inverted. Let us define Z to be the output signal of the inverter and ϕ_Z to be its corresponding mRNA concentration. The transcription stage, with input ρ_A and output ϕ_Z, has a steady state relationship in which increases in ρ_A correspond to monotonic decreases in ϕ_Z. During periods of minimal repression, transcription progresses at rapid rates resulting in maximal concentrations of ϕ_Z. However, for high levels of repression, the transcriptional activity declines and the level of ϕ_Z drops.

Overall, the three stages combine to form a system that behaves as an inverter, negating the input mRNA signal, ϕ_A, to yield the output mRNA signal, ϕ_Z. Furthermore, with efficient signal restoration during the cooperative binding stage of inversion, complex but reliable digital logic circuits are attainable.

2.2 NAND *gate*

Biochemical inverters are used to construct more sophisticated gates and logic circuits. For example, a NAND gate can be designed by "wiring-OR" the outputs of two inverters by assigning them same output gene (Weiss et al., 1999). Figure 4 depicts a circuit in which a NAND gate is connected to an inverter. For simplicity, both mRNA and their corresponding protein products are used to denote the signals, or the circuit wires. Again, the regulation of the promoter and mRNA and protein decay enable the gate to perform computation. The NAND gate protein output is expressed in the absence of either of



Repressor	Inducer	Output
0	0	1
0	1	1
1	0	0
1	1	1

Figure 5. Design of a biochemical gate for the IMPLIES logic function that enables extra-cellular interaction. With no inducer present, the gate behaves similarly to the inverter. However, when the inducer is present, it binds the repressor and modifies its three-dimensional confirmation, thereby preventing the resulting molecular complex from binding the operator. As a result, transcription proceeds even in the presence of the repressor.

the inputs, and transcription of the output gene is only inhibited when both input repressor proteins are present.

Because the performance of a NAND gate depends only on its constituent inverters, well-designed inverters can be engineered into a variety of reliable combinatorial gates. Furthermore, the NAND gate is a universal logic element that can theoretically be used to wire any finite intracellular digital circuit.

2.3 IMPLIES *gate for external interaction*

Cells use the IMPLIES gate to receive control messages sent by humans or to detect certain environmental conditions (Weiss, 2001). The biochemical reactions, the logic symbol, and the truth table for an intercellular construct that implements the IMPLIES gate are shown in Figure 5. The gate has two inputs: a repressor protein and an inducer, that is typically a small molecule that diffuses through the cell membrane. When the input mRNA and its corresponding protein are absent, RNAp binds the promoter and the output mRNA is transcribed, resulting in a logical HIGH output. If only the input repressor protein is present, it binds the promoter and prevents the RNAp from initiating transcription, yielding a logical LOW output. In order to interact with a cell, one can introduce the appropriate inducer molecule to its surrounding environment. If the inducer molecule exists at sufficiently high concentration, it will diffuse through the cell membrane and affect the output of the IMPLIES gate. When both the inducer and repressor are present, the inducer binds

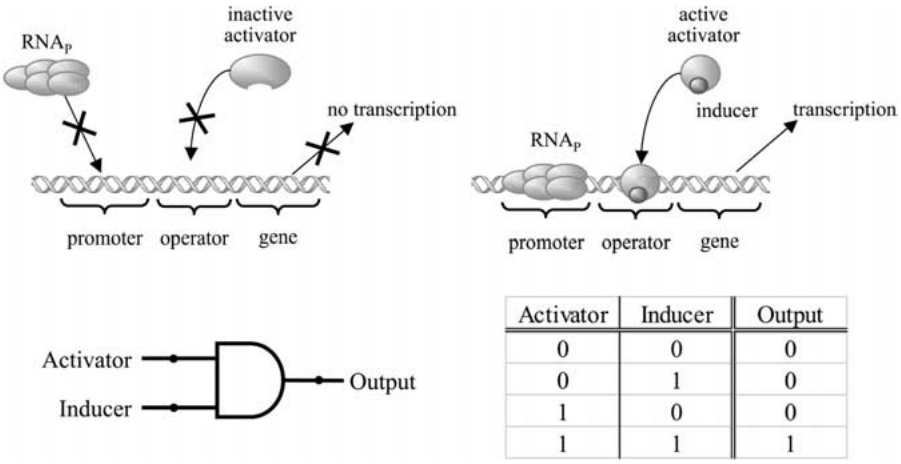
the repressor, changes the repressor's three-dimensional conformation, and renders the resulting molecular complex incapable of binding the promoter. In this case, transcription occurs and a HIGH output ensues.

The IMPLIES gate is comprised of the same three biochemical stages as the inverter: translation, cooperative binding, and transcription. The fundamental difference between the two gates lies in the role of the inducer. The concentration of the inducer acts as an additional input, along with the input repressor mRNA, to the cooperative binding phase, $\mathcal{C}'$. Let the inducer concentration be denoted as υ_A, the concentration level of input mRNA as ϕ_A, and the concentration level of output mRNA as ϕ_Z. In this construct, the repression strength, ρ_A, depends non-linearly on the level of the repressor (as in the case of the inverter), and also on the level of the inducer. Specifically, $\mathcal{C}'$ is determined by the binding affinities of the active repressor to the operator and the affinity of the inducer to the repressor.

Inducers and mRNA/proteins are not interchangeable signals. A protein is coupled to a promoter that contains an operator region that specifically binds the protein. This protein/promoter interaction is the important interaction for characterizing a logic gate. The output of a logic gate A can be connected to the input of a logic gate B by fusing the promoter of A with a DNA coding region for input protein B. If we ignore fan-in and fan-out issues, these protein/promoter gates can theoretically be configured to implement any arbitrary digital logic function. In contrast, the IMPLIES gate represents a hybrid gate that has inputs of two different types. The inducer molecule input is always coupled to the corresponding protein molecule input. When designing circuits with such hybrid gates, it is important to ensure that there are no unintended interactions between various inducer molecules and proteins.

2.4 AND *gate for cell-cell communications*

The AND gate is utilized by cells to detect incoming messages sent by other neighboring cells (Weiss and Knight Jr., 2000). The biochemical reactions, the logic symbol, and the truth table for an intercellular gate that implements the AND function are illustrated in Figure 6. In this construct, RNAp has a low affinity for the promoter and thus, basal transcription activity is minimal. It follows that in the absence of the activator and inducer, the logic output of the AND gate is LOW. When only the activator is present, the output is still LOW, since the activator has little affinity for the operator without its corresponding inducer. The output is HIGH only if both the activator and inducer are present. In this case, the inducer binds the activator and changes its conformation, yielding an activator/inducer complex that binds the promoter.



Activator	Inducer	Output
0	0	0
0	1	0
1	0	0
1	1	1

Figure 6. Detecting cell-cell communications with AND gate.

This complex helps recruit RNAp to the promoter and initiate transcription to yield a HIGH output.

The biochemical reaction stage involved in the AND gate include translation, cooperative binding, and transcription. The translation stage is very similar to those of the NOT and IMPLIES gates. The cooperative binding stage, $\mathcal{C}''$, takes the activator protein, ρ_A, and inducer, υ_A, as inputs. However, unlike the NOT and IMPLIES gates that map the inputs to repression levels, the output of $\mathcal{C}''$ is an activation level, ω_A. As in the IMPLIES gate, the activation level depends non-linearly on the levels of the activator and inducer. In addition, the binding affinities of the inducer to the activator and of the activator/inducer complex to the operator determine the shape of $\mathcal{C}''$. The final transcription stage of the AND gate maps the activation level, ω_A, to output mRNA, ϕ_Z, in a direct relation.

As with the IMPLIES gate, the two inputs to the AND gate are not interchangeable, since the inducer must be coupled to its corresponding activator. Here again, it is imperative that the circuit elements are comprised of only non-interfering activator/inducer pairs and that there are no inadvertent interactions between inducer molecules and activators.

3. Implementation of prototype circuits

This section describes the implementation of three prototype circuits using the inverter and IMPLIES gate genetic circuit building blocks of Section 2. These experimental circuits consist of one, two, or three gates, and exhibit various behaviors such as the reduced noise of gene expression when

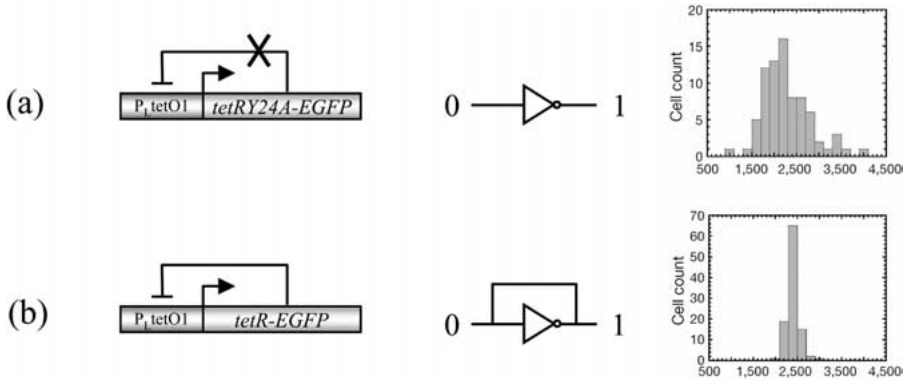


Figure 7. The auto-repressor circuit consists of a $P_L$tetO-1 promoter that regulates a tetR/EGFP fusion protein. In case (a) with a non-functional mutated version of tetR, the circuit behaves as a simple inverter because the tetR mutant protein (tetRY24A) cannot bind the tetO operator site on the promoter. The circuit exhibits wide fluctuations in gene expression. However, when the wild-type tetR is used in case (b), the negative feedback causes gene expression to be more uniform among the cell population.

compared to a simple inverter (Section 3.1), bistability (Section 3.2), and oscillation (Section 3.3).

3.1 *Auto repressor*

All biochemical processes in cells such as transcription, translation, and the protein decay are inherently stochastic. Thus, a population of cells with an identical genetic makeup grown under the same environmental conditions can still show significant cell-cell variation in behavior. The extent of gene expression fluctuations in a single inverter can be reduced by introducing a negative feedback loop, as illustrated by the auto-repressor circuit (Becskei and Serrano, 2000). In Figure 7, a *tetR*-EGFP (tetracycline repressor fused with Enhanced Green Fluorescent Protein) coding sequence and EGFP (Enhanced Green Fluorescent Protein) is placed downstream of $P_L$tetO-1.

In case (a), tetRY24A is a one amino-acid mutant protein that does not bind the tet operator of $P_L$tetO-1, and therefore the circuit behaves as a simple inverter. The corresponding histogram, which relates measured fluorescence intensity to cell count, shows wide variations in output gene expression. In case (b) where the functional wild-type tetR is used, the circuit contains negative feedback that serves to reduce the noise of gene expression. Since the TetR that occupies the tetO operator inhibits RNAp from binding the $P_L$tetO-1 promoter, the TetR protein repressor restricts the production of itself (and the EGFP fusion protein). This negative feedback reduces variations in gene expression from $P_L$tetO-1 as can be seen by the histogram in Figure 7(b).

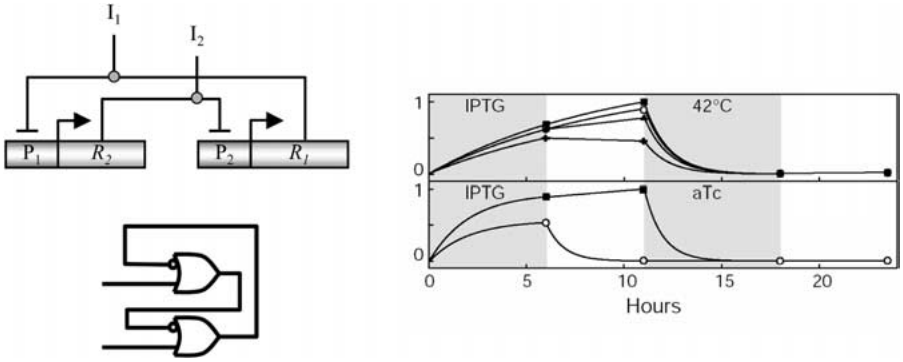


Figure 8. The toggle switch gene network consists of two proteins, R_1 and R_2, that repress the expression of each other from promoters P_1 and P_2. To set the toggle switch to either one of its bistable states, P_1 and P_2 expression is induced externally using I_1 and I_2. This network is represented as a logic circuit with two interconnected IMPLIES gates. The behavior of six different choices of genetic elements is shown in the time-series response of the circuits to external inducers. Five out of six configurations exhibit bi-stability.

3.2 *Toggle switch*

The toggle switch is a synthetic gene network that retains bistable states (Gardner et al., 2000). Chemical or thermal induction allows the switch to move between the stable states. The genetic circuit comprises two repressors, R_1 and R_2, their promoters P_1 and P_2, and inducers I_1 and I_2 that deactivate the repressors (Figure 8). The logic circuit representation consists of two interconnected IMPLIES gates. Two types of this gene network, with a total of six different versions, were synthesized and checked for bistability.

The first type includes four versions of the network that contain *lacI* and a temperature sensitive λ repressor (cIts) downstream of the λ P_Ls1con and Ptrc-2 promoters, respectively. LacI represses Ptrc-2, whereas cIts represses the λ P_Ls1con promoter. The four different configurations differ in their RBS sequences located upstream of LacI. A green fluorescent protein (GFPmut3) downstream of cIts reports the state of the network. When the system is exposed to IPTG for approximately six hours, the four LacI/cIts versions of the circuit switch from the LOW to HIGH state. This is because IPTG bound to LacI can no longer repress the Ptrc-2 promoter. If the cells are diluted into fresh media with no inducers, they still retain the HIGH state after five additional hours of growth, as illustrated in the top graph of Figure 8. After incubation at 42°C for an additional seven hours, all four systems switch back to the LOW state.

The other type of the toggle switch network includes two new versions where TetR replaces cIts and $P_L$tetO-1 replaces λ P_Ls1con. Again, two different RBS sequences were placed upstream of lacI. Of these, only one

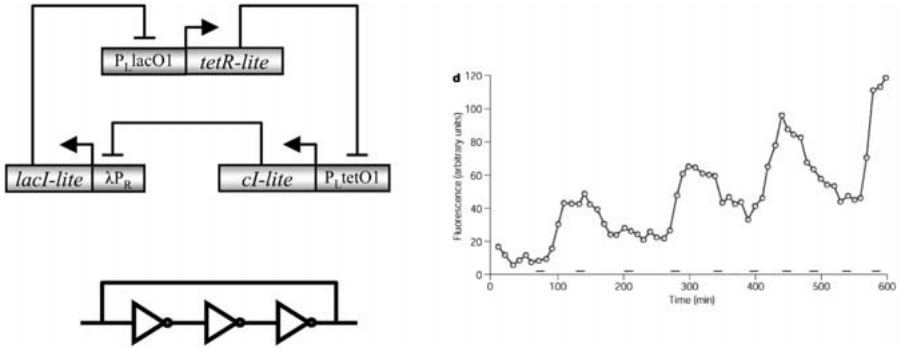


Figure 9. The repressilator circuit consists of three proteins and their three corresponding promoters, arranged such that each protein P_x represses the expression of a different protein P_y which does not repress P_x. These proteins include a synthetic tag, signified by the suffix "lite", that targets the proteins for fast decay in the cell. The gene network configuration corresponds to a ring oscillator logic circuit. A green fluorescent protein placed downstream of a $P_L$tetO-1 promoter reports the approximate concentration of *cI* in any particular cell. The graph shows the time evolution of fluorescence intensity in a particular cell that exhibits oscillation, with the small bars depicting cell division events. Oscillatory behavior of other cells varied considerably.

configuration demonstrated bistability (lower graph in Figure 8). In the non-functional version, it is likely that even in the repressed state, LacI expression was stronger than the repression efficiency of TetR, and eventually LacI concentration increased back to the original high levels.

3.3 *Ring oscillator*

The repressilator is an oscillatory gene network constructed with three repressors that are not part of a natural biological clock network (Elowitz and Leibler, 2000). The oscillation frequency of the genetic network is less than the cell division frequency, and as a result, the oscillations are propagated through the generations. The network comprises three fast-decaying versions of the repressors CI, LacI, and TetR, and their corresponding promoters. CI represses the expression of LacI, which in turn represses the transcription of *tetR*. The correspondng TetR protein subsequently inhibits the expression of CI (Figure 9). When the kinetic characteristics are matched between the different gene components under the appropriate delay conditions, this system will oscillate with regular periodicity and amplitude.

In order to optimize the dynamic behavior of the system, the repressor proteins were fused with amino-acid tails that encode decay tags. These tags reduce protein stability and increase *in-vivo* decay rates. A *gfp* placed downstream of an additional $P_L$tetO-1 promoter (not shown) reports the

approximate level of CI. The graph in Figure 9 shows the oscillation of a single cell as measured by GFP fluorescence over time. Oscillatory behavior was observed by approximately half of the individual cells. However, the period and amplitude of oscillations varied dramatically from cell to cell, even though they contained an identical genetic construct. This variability is likely attributed to the stochastic nature of the biochemical reactions that govern the cellular machinery of gene expression. Various research projects are currently attempting to implement synthetic gene networks that exhibit improved oscillatory behavior. One approach involves synchronizing gene expression among a cell population using cell-cell communications (McMillen et al., 2002).

4. Circuit design

We have already demonstrated that prototype circuits can be used to construct small scale synthetic gene networks with interesting behavior. However, even at this scale, we lack the ability to design a DNA sequence that reliably implements a desired cellular function with quantitative precision. This issue will become more and more important as the size of target synthetic gene network grows.

In this section we explore two seemingly divergent approaches for genetic circuit design. The first methodology employs "rational design" in which an attempt is made to gain accurate knowledge of the behavior of the genetic components and their compositions. It includes both modeling of gene networks and making modifications of genetic elements based on simulation results until the components achieve the desired characteristics. The second approach, *directed evolution*, uses large scale genetic mutations and combinatorial synthesis, combined with high throughput assays, to screen for genetic network variations that yield the desired behavior. Ultimately, the most productive approach for genetic circuit design will likely integrate rational design with directed evolution and combinatorial synthesis.

4.1 *Rational design*

4.1.1 *Modeling*

Computational models are necessary for systematic circuit design and analysis. Biological circuit models are similar to electrical circuit models in that the simulated behavior of the system depends on the characteristics of the components. However, fundamental differences between the different substrates require modeling techniques that optimize different facets of the simulations. For example, in electrical circuits, components are spatially

separated and the connectivity is determined by a fixed conducting pathway or wire. Biological simulations, on the other hand, must consider the sharing of physical space by multiple components and their widely varying kinetic characteristics. An electrical circuit is typically modeled as a deterministic network. Specifically, the circuit state at a given time and the regulatory interactions between components are sufficient for determining the next state. In contrast, biological circuit models often need to take into account the stochastic nature of chemical reactions in cells. Experimental data has revealed that genetic networks tend to exhibit significant noise even for the simplest configurations (Elowitz et al., 2002). While these findings suggest that stochastic models may be better suited for analysis of genetic circuits, other forms of modeling are widely used. Depending on the configuration of the genetic network, deterministic modeling tools may shed insight into important characteristics of the networks (De Jong, 2002).

Many types of models and simulations, with varying degrees of specificity and difficulty, have been used to describe biological circuits. One simple approach, boolean networks, can be used to sort large amounts of biological data, such as from microarrays, in order to gain theoretical understanding of the systems (Kauffman, 1969). The boolean models assume that gene expression is discrete, i.e., ON or OFF, and that state transitions are deterministic and synchronous. In practice, biological transitions are rarely simultaneous, and assuming otherwise can prevent the accurate prediction of many concentration-sensitive behaviors. Boolean networks are resource-efficient but come at the expense of possibly over-simplifying the structure and dynamics of biological networks (De Jong, 2002).

A prevalent method for modeling dynamic systems, such as biological circuits, is to employ nonlinear ordinary differential equations (ODE). The concentrations of RNA, proteins, and other molecules are represented by time-dependent variables. Rate equations describe the biochemical reactions as a function of the concentrations of the various species in the circuit and are of the form

$$\frac{dx_i}{dt} = f_i(x), 1 < i < n$$

where the vector $x = [x_1, \ldots, x_n]' \geq 0$ includes the concentrations of proteins, mRNAs, and other molecules, and $f_i : \mathbf{R}^n \rightarrow \mathbf{R}$ is usually a nonlinear function (De Jong, 2002).

Table 1 lists the chemical reactions that model the inverter in Figure 2. The mechanisms modeled here include translation of the input protein from the input mRNA (reactions 1–2), input protein dimerization and decay (reactions 3–6), cooperative binding of the input protein (reactions 7–12), transcription (reaction 13), and degradation of the output mRNA (reaction 14). From the

Table 1. Biochemical reactions that model an inverter. $mRNA_A$ is the input and $mRNA_Z$ the output

$$mRNA_A + rRNA \xrightarrow[\text{translate}]{k_{\text{xlate}}} mRNA_A + rRNA + A \quad (1)$$

$$mRNA_A \xrightarrow[\text{decay}]{k_{\text{dec(mrna)}}} \quad (2)$$

$$A + A \xrightarrow[\text{dimerization}]{k_{\text{dim(a)}}} A_2 \quad (3)$$

$$A_2 \xrightarrow[\text{single}]{k_{\text{sngl(a)}}} A + A \quad (4)$$

$$A \xrightarrow[\text{decay}]{k_{\text{dec(a)}}} \quad (5)$$

$$A_2 \xrightarrow[\text{decay}]{k_{\text{dec(a2)}}} \quad (6)$$

$$P_Z + A_2 \xrightarrow[\text{repress 1}]{k_{\text{rprs(a2)}}} P_Z A_2 \quad (7)$$

$$P_Z A_2 \xrightarrow[\text{dissociation}]{k_{\text{dis(a2)}}} P_Z + A_2 \quad (8)$$

$$P_Z A_2 \xrightarrow[\text{decay}]{k_{\text{dec(ga2)}}} P_Z \quad (9)$$

$$P_Z A_4 \xrightarrow[\text{decay}]{k_{\text{dec(ga4)}}} P_Z A_2 \quad (10)$$

$$P_Z A_2 + A_2 \xrightarrow[\text{repress 2}]{k_{\text{rprs(a4)}}} P_Z A_4 \quad (11)$$

$$P_Z A_4 \xrightarrow[\text{dissociation}]{k_{\text{dis(a4)}}} P_Z A_2 + A_2 \quad (12)$$

$$P_Z + RNA_p \xrightarrow[\text{transcribe}]{k_{\text{xscribe}}} P_Z + RNA_p + mRNA_Z \quad (13)$$

$$mRNA_Z \xrightarrow[\text{decay}]{k_{\text{dec(mrna)}}} \quad (14)$$

fourteen reactions, seven ordinary differential equations are derived, one for each molecular species in the system (Weiss et al., 1999). Each differential equation describes the time-domain behavior of a particular molecular species based on all the equations in the biochemical model that include that particular molecule. For example, the ordinary differential equation for simulating the time-dependent molecular concentration of the input protein A is:

$$\begin{aligned} d(A) \;=\; & 2 \cdot k_{sngl(a)} \cdot A_2 - k_{dec(a)} \cdot A + k_{xlate} \cdot rRNA \cdot mRNA_A \\ & -2 \cdot k_{dim(a)} \cdot A^2 \end{aligned}$$

while the equation for simulating promoter P_Z bound by protein dimer A_2 is:

$$\begin{aligned} d(P_Z A_2) \;=\; & k_{rprs(a2)} \cdot P_Z \cdot A_2 - k_{dis(a2)} \cdot P_Z A_2 - k_{rprs(a4)} \cdot P_Z A_2 \cdot A_2 \\ & + k_{dec(ga4)} \cdot P_Z A_4 - k_{dec(ga2)} \cdot P_Z A_2 + k_{dis(a4)} \cdot P_Z A_4 \end{aligned}$$

Figure 10 shows an ODE simulation of the dynamic behavior of the inverter circuit with the above chemical reactions in response to an external stimulus. The kinetic constants used in this simulation were obtained from the literature describing the phage λ promoter P_R and repressor (CI) mechanism (Hendrix, 1983; Ptashne, 1986). The top graph represents the input mRNA_A,

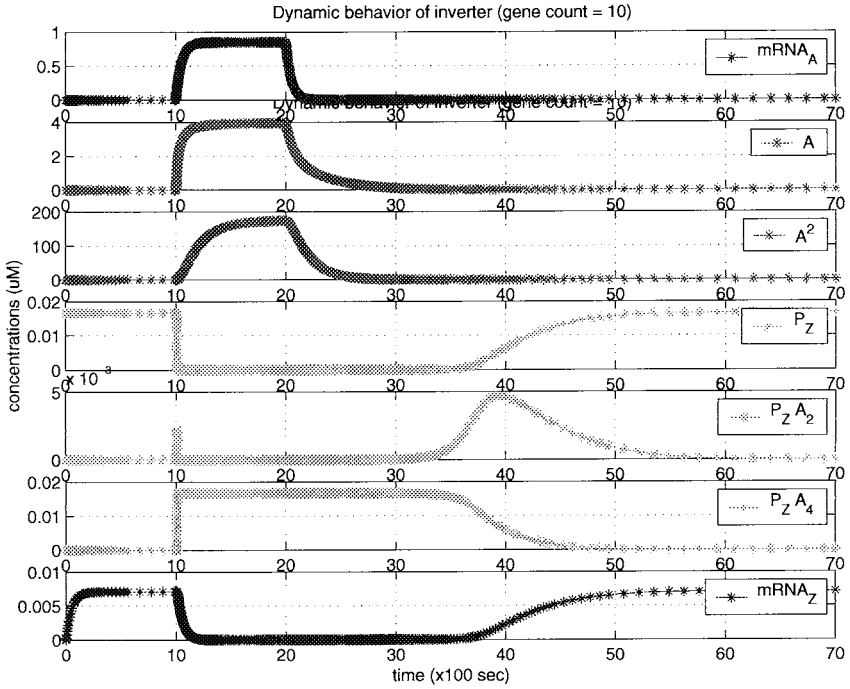


Figure 10. The dynamic behavior of the inverter. The graphs show a time-series of the molecular concentrations involved in inversion, in response to a stimulus of input mRNA.

followed by graphs for the input protein repressor and its dimeric form, free and occupied forms of the promoter, and finally the output mRNA_Z.

The reactions proceed as follows: at first, input mRNA and input protein repressor are absent. As a result, P_Z is active and RNAp transcribes output mRNA_Z. The level of mRNA_Z increases until it stabilizes when the gene expression and decay reactions reach a steady-state. At this point, the input signal is LOW while the output signal is HIGH.

Then, an externally-imposed drive increases the input mRNA_A, which is further translated into the input repressor protein A. This protein begins to form dimers, which bind the promoter's free operators. The system quickly reaches a state where each promoter is essentially completely bound by two dimers. The almost complete inactivation of the promoters occurs at a fairly low concentration of the dimer A_2, and indicates the strong repression efficiency of the CI repressor that is used for this simulation. As a result of the promoter inactivation, transcription stops and the output mRNA_Z decays to zero. At the end of this stage, the input signal is HIGH while the output signal is LOW.

Finally, mRNA_A synthesis is halted and the lack of this external drive results in the decay of existing mRNA_A, A, and A_2. Slowly, the repressor dimers dissociate from the P_Z operators, and the level of the active promoter P_Z increases to the original level. This allows RNAp to resume transcription of P_Z, and the level of the output mRNA_Z rises again. At this stage, the input signal reverts to LOW, while the output signal becomes HIGH.

A significant problem with using differential reaction-rate equations lies in the model's assumption that biochemical systems are both continuous and deterministic. While these assumptions are convenient for simulating systems with a large number of molecules for any given species, they often lead to significant inaccuracies in modeling biochemical systems such as cells that often consist of only a few molecules for a given species. Molecular reactions change the population dynamics discretely, thereby invalidating the former assumption of time continuity. The latter assumption can be refuted with the following argument. Without knowing the exact positions and velocities of every molecule in a system, a prediction of the precise molecular population levels is impossible (Gillespie, 1977). For cases when only the average behavior of a genetic circuit needs to be modeled, deterministic models suffice. This is because the number of each type of molecule is large compared to thermal fluctuations and the number of each type of reaction per unit time in each observation interval is also large (Mcadams and Arkin, 1998). If, however, the model needs to describe population heterogeneity for a small number of reacting molecules, stochastic fluctuations due to random noise need to be taken into account (Rao and Arkin, 2001). The main limitations to stochastic modeling include the large computational resource requirements and the lack of accurate quantitative information about noise in the system.

For networks of modest complexity and small size, it may be worthwhile exercise to model the system both deterministically and stochastically to compare the results, particularly if any experimental data is available to validate one model. The repressillator described in Section 3.3 was simulated using both methods and the results were compared to experimental data (Elowitz and Leibler, 2000). The deterministic model predicted regular oscillation with consistent amplitudes, and the stochastic model showed damped oscillation. The experimental data for one particular cell exhibited periodic oscillations with a baseline drift (Figure 9), but overall cells in a microcolony exhibited large variations in the period and amplitude of oscillations. The noisy experimental data is the result of unaccounted endogenous interactions. In theory, this system will oscillate indefinitely but there will likely be an imbalance in the network that will be amplified by the circuit feedback. Deterministic models average out slight imbalances in the system,

predicting less erratic behavior than what naturally occurs. A stochastic model, however, may not necessarily be more accurate because the regulatory mechanisms may not have been modeled correctly. As a guideline, networks with feedback, such as the ring oscillator, will have a greater accumulation of random reactions as compared to combinatorial networks such as a cascade of inverters.

Certain studies have shown robust deterministic systems that can accurately model network response to variations in initial conditions and parameter values over orders of magnitude. Examples include models of bacterial chemotaxis (Barkai and Leibler, 1997) and a simulation of the segment polarity network in Drosophila (von Dassow et al., 2000). These findings suggest that a simulated system may be stable as a result of the network structure and not because of particular parameter values (De Jong, 2002). In spite of this, the lack of quantitative kinetic parameters for the rate equations and incomplete knowledge of the biochemical reaction mechanisms are significant hindrances to achieving accurate deterministic and stochastic simulations. But as shared databases become more prevalent, mathematical network identification techniques will be used to estimate more parameter values and improve the breadth of our knowledge. Synthetic gene regulatory networks are also playing an important role in achieving precise biological circuit models. Parameters can be measured directly and systematically using methods such as the ones described in the next section.

4.1.2 *Engineering the device physics of gates*

One of the main issues encountered in designing genetic circuits is the ability to match the components into combinations that result in the correct behavior. In practice, this task is harder than what one might expect. Naturally occurring components have widely varying kinetic characteristics and coupling these elements together will most likely not result in desirable behavior. As a result, there is a need for genetic process engineering, in which genetic elements are modified until they are tuned to respond in a desirable fashion.

To study the device physics of these circuits, we start our investigation with a simple inverter circuit where the output is inversely proportional to the input: a HIGH concentration of the input signal results in a LOW concentration at the output and vice versa. In order to observe the response we need to construct the circuit so that the input can be externally controlled to achieve desirable levels. This can be achieved by coupling the inverter to an IMPLIES gate. Figure 11 depicts the construct of this inverter circuit.

In the first part of the circuit, promoter P_1 regulates the expression of gene R_2. Since the cell does not have a repressor for P_1 (LOW input), R_2 is expressed constitutively and the gate's output is always HIGH. R_2 serves as

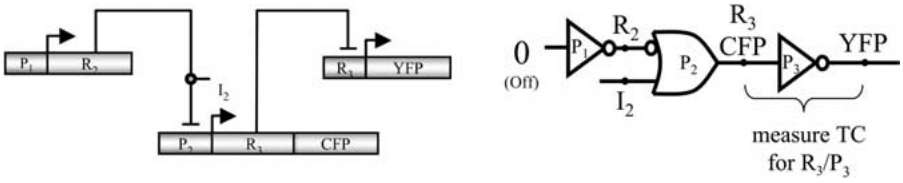
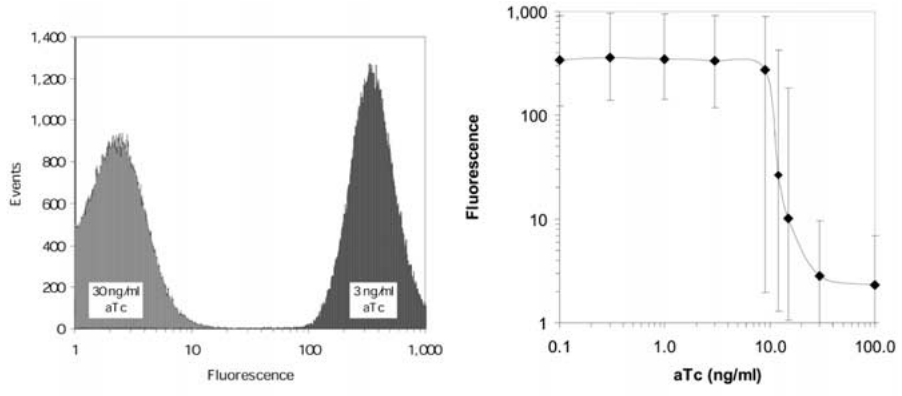


Figure 11. Genetic circuit diagram to measure the device physics of an R_3/P_3 inverter: digital logic circuit and the genetic regulatory network implementation (P_x: promoters, R_x: repressors, CFP/YFP: reporters).

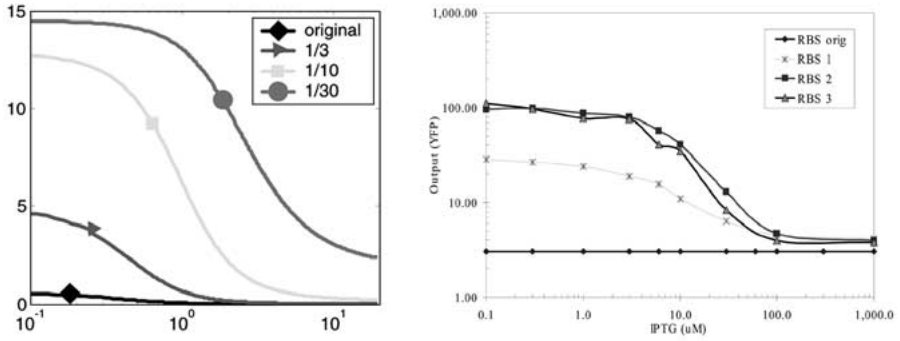


(a) Digital computation noise margins (b) Gain / signal restoration

Figure 12. Transfer curve gain and noise margins for the lacI/p(lac) circuit.

the input signal to the next stage of the circuit and it also acts as a repressor for promoter P_2. This promoter regulates gene R_3. Using external interaction, an inducer molecule I_2 can bind R_2 and prevent it from binding the operator site of P_2, hence allowing the expression of R_3. This part of the circuit thus acts as a buffer such that when I_2 is present, the output is R_3 and when I_2 is absent, the expression of R_3 is repressed. A reporter cyan fluorescent protein (CFP) is placed downstream of R_3 to probe for the signal at this stage. R_3 is the input to the next stage of the circuit and it acts to repress the transcription of *YFP*. In terms of the overall response of this system, the presence of I_2 (HIGH input) corresponds to the repression of the YFP (LOW output) and the absence of I_2 (LOW input) allows the production of the fluorescent protein (HIGH output).

This circuit can be configured using different repressor/promoter pairs. Figure 13 shows the experimental results of the system where P_1, P_2 and P_3 are $\lambda_{P(R-O12)}$, $P_L$tetO-1, and p(lac), respectively. Inducer I_2 is anhydrotetracycline (aTc), R_2 is the TetR protein, and R_3 is the LacI protein. Figure 12(a) shows Fluorescence-Activated Cell Sorting (FACS) (Shapiro,



(a) Simulations (b) Results

Figure 13. Simulations and experimental results demonstrating the effect of modifying RBS efficiencies on circuit performance.

1995) cell population data of EYFP in two separate experiments where the cells were exposed to HIGH and LOW concentrations of the inducer input (aTc). For a LOW aTc level of $3\frac{ng}{ml}$ the cells are highly fluorescent (HIGH output). For a HIGH aTc input level of $30\frac{ng}{ml}$, the output is LOW. Figure 12(b) illustrates the transfer curve of the circuit with respect to different aTc input levels. The error bars denote the range that includes 95% of the fluorescent intensities of individual cells measured for corresponding aTc levels. The signal to noise ratio and the sharp transition between the two states demonstrate the potential of such circuits for digital logic computation.

While the *lacI*/p(lac) inverter exhibits the correct behavior for a digital inverter, other combinations of repressor/promoter can also be used to construct additional components for the cellular gate library. For the purpose of assembling a component library, we built another version of this circuit with p(lacIq), p(lac), and $\lambda_{P(R)}$ as the promoters P_1, P_2, and P_3, and LacI and CI as the repressors R_2 and R_3. The input to this circuit is IPTG.

It is worth mentioning that the CI repressor binds cooperatively to λ promoter's two operator sites, O_R1 and O_R2 very efficiently, resulting in a high gain for such a system. The dimeric form of CI has a high affinity for the O_R1 region of the operator and it is to this site, to which it first binds. In addition, the binding of CI to O_R1 increases the affinity of a second dimer for the O_R2 site, due to the interaction with the bound dimer. Therefore dimers bind O_R1 and O_R2 almost simultaneously. This cooperative binding is highly desirable in the design of logic circuits as it provides a high gain and quick change in output response over a small change in input concentrations.

This circuit is expected to exhibit an inverse correlation between input IPTG levels and output EYFP. However the initial experimental results showed no response in this case and the system seemed to be insensitive to

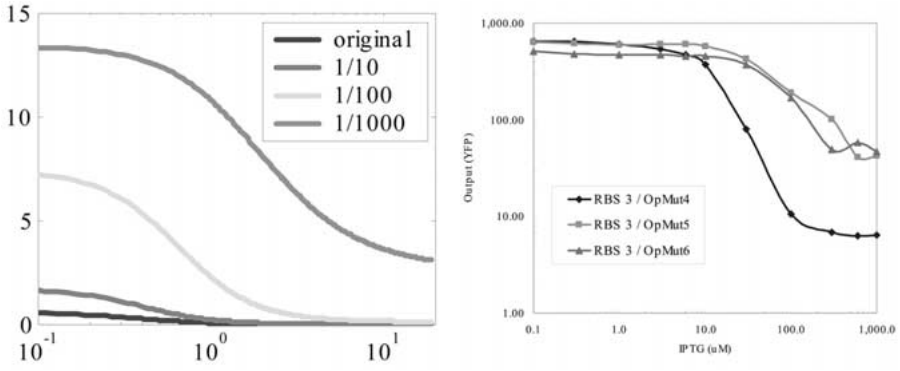
variations in the levels of input IPTG. This lack of response stems from the mismatch between the kinetics characteristics of LacI/p(lac) gate versus the CI/$\lambda_{P(R)}$ inverter. Even in the absence of IPTG and a maximum repression of the p(lac) promoter, a low basal level of CI is expressed. Because CI is a very efficient repressor, it fully represses the $\lambda_{P(R)}$ promoter even at low concentrations and prevents the expression of the output fluorescent protein.

This kinetic mistmatch highlights the importance of understanding the device physics of the cellular logic gates. It also necessitates the construction of a mechanism to measure and fine tune these kinetic parameters in order to be able to construct complex logic circuits from theses simple components.

In order to overcome the mismatch problem one can mutate genetic elements that have a significant effect on the system dynamics until the desirable response is obtained. Two elements that were modified in the genetic inverter were the RBS for *cI* and the λ promoter's O_R1 operator site. The RBS plays an important role in determining the response of the system because it controls the rate of translation of the input mRNA. RBS sequence aligns the ribosome onto the mRNA in the correct reading frame in order to initiate the translation of the first codon. For a given input mRNA, a reduction in translation efficiency results in a lower protein synthesis rate and corresponding change in the system response. Simulation using BioSPICE (Weiss et al., 1999), our internally developed genetic network modeling tool, revealed the effect of modifying the RBS efficiencies, as illustrated in Figure 13(a).

Experimental work in which the RBS was mutated confirmed the effect of reducing RBS strength for improving the circuit response (Weiss and Basu, 2002), as shown in Figure 13(b). Here three different RBS sequences with weaker translation efficiencies than the original RBS were synthesized and tested in our laboratory. The graph shows that all weaker ribosome binding sites exhibit improved response to the IPTG levels. These results demonstrate that the mechanism of genetic process engineering can convert a non-functional circuit with a flat response into a functional one.

In addition to RBS sequences, the operator sites of promoters can be mutated to optimize system response. BioSPICE simulations shows how a reduction in the affinity of the repressor CI to the operating site of the promoter $\lambda_{P(R)}$ influences the inverter's response (Figure 14(a)). We constructed additional circuits where the O_R1 site was modified using site-directed mutagenesis in order to verify these predictions and to continue assembly of the cellular gate library. In order to get the optimum response, these modifications were combined with the weakest RBS from above, as shown in Figure 14(b). OpMut4, a single base mutation to O_R1, resulted in the best system performance, while additional base pair mutations to O_R1 resulted in responses where the circuits were unable to fully repress the



(a) Simulations (b) Results

Figure 14. Simulations and experimental results demonstrating the effect of modifying repressor/operator binding affinities on circuit performance.

output. These results reveal that the RBS-3/OpMut4 combination yields a highly optimized circuit suitable for digital logic computation.

4.1.3 *Directed evolution*

Constructing a genetic circuit with desirable behavior is a challenging task because the response of the system depends on many kinetic parameters within the cell that are unknown and difficult to control. The process of optimizing circuit performance can be quite labor intensive, as described in the previous section. Yet biological systems offer a powerful design feature: the ability to evolve and be optimized under the pressure of natural and artificial selection. This process of *directed evolution* can be used as a tool to construct well-behaved logic elements and circuits. In the previous section, it was shown that a logic inverter can be tuned through rational design and modification of RBS sequences and operator sites. Here we describe an evolutionary method to achieve the same goal. Directed evolution is performed by placing random mutations in the DNA sequences of a genetic network. By restricting mutations to only a specific region, one can rapidly test how the modified component of the circuit contributes to the overall response of the system.

In order to investigate directed evolution, we mutated the non-functional version of the CI/λ circuit discussed in the previous section (Yokobayashi et al., 2002). Specifically, random mutations were incorporated into the *cI* gene and its RBS to determine whether a functional system could result from mutations in this gene and its RBS. About 50% of the colonies of the mutated strain exhibited fluorescence in the absence of IPTG, which is the desired behavior for this input condition. This behavior can be explained by a mutation that

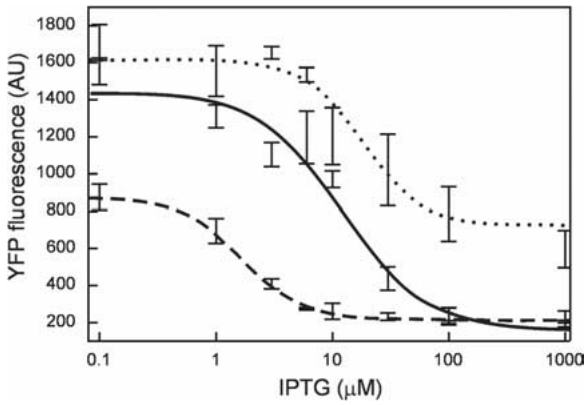


Figure 15. Experimental results demonstrating the ability to optimize system response using directed evolution. Curves are best fit to data points using a standard Hill equation, and error bars represent the results of triplicate experiments.

possibly inactivates the *cI* gene, thus rendering it incapable of repressing the λ promoter. We would like to distinguish between mutations that completely inactivate *cI* versus those that partially reduce its repression efficiency, where the latter is the desired response. To ensure correct response of the system for all input conditions, we further investigated the fluorescent colonies by transferring them to a new plate and exposing them to a high level of IPTG. About 5–10% of the colonies were not fluorescent, which implies that the mutations did not fully disrupt CI's ability to repress the λ promoter.

DNA sequencing of the colonies with the evolved ability to switch state from HIGH to LOW revealed a variety of mutations in the system. One of the most abundant mutations was the introduction of a stop codon into the *cI* gene. The resulting partial CI protein is expected to have reduced dimerization and decreased repressor/operator affinity. These reductions in repression efficiency enable the output to switch states as desired. Other colonies evolved a different set of mutations in the *cI* start codon that reduced the translation efficiency. This latter effect is similar to the one explored by rational design (Figure 13).

Of the colonies that exhibited the correct behavior, two were chosen for further optimization in a second generation of evolution. Sequencing selected second-round clones with the correct fluorescence response revealed that amino acids substitutions in *cI*. Figure 15 depicts the transfer curve for two of the clones that were tested in this experiment (dotted and solid lines) and compared to the response of the circuit with the weak RBS. As depicted in the graph, the system response is sharper with the introduction of mutations.

These results allow us to conclude that evolved mutants can adjust the kinetic characteristics of the genetic network to result in correct behavior.

While rational design proved to be effective, as demonstrated by the effect of modifying operator sites and RBS sequences, our experience also reveals the power of directed evolution due to the discovery of effective amino acid substitutions that would have been hard to develop rationally. Ultimately, the most efficient approach for genetic circuit design will likely integrate rational design and directed evolution.

4.1.4 *Combinatorial circuit synthesis*

In a recent project, a genetic combinatorial library was constructed to study how gene network connectivities determine function (Guet et al., 2002). The combinatorial library consisted of small set of regulatory genes and their promoters with varying connectivity. Three well characterized regulatory genes (*lacI*, *tetR*, and *cI*) were randomly inserted downstream of three promoters picked randomly from a set of five promoters that regulate these genes. The network was constructed with following structure: P_i-lacI-P_j-cI-P_k-tetR, where P_i, P_j and P_k are chosen from the five promoters. Diverse connectivities resulted from the fact that the three genes activated and repressed one another. The behavior of these circuits was monitored with a GFP reporter, while the levels of the aTc and IPTG inducer molecules served as the inputs. The plasmid library was transformed into two *E. coli* wild types strains, and each clone was grown under four separate conditions: with or without IPTG and aTc. GFP fluorescence was monitored to determine the behavior of each clone. One of the goals of this work was to search the combinatorial circuit library for versions that behave as logic gates in which the output is a binary logic function of both inducers.

To determine network phenotypes, 30 clones with different characteristics were transformed and examined. Figure 16 shows all the combinations that were generated and their corresponding GFP expression under the four different input conditions. Sequencing results determined thirteen different network topologies that were generated depending on the connectivities of the elements. As expected, the variety in network connectivity appears to be a source of phenotype differences within the library. Single changes in network connectivity, such as replacing a promoter with another one, altered the system response. However, the behavior of the circuits also depended on the particular strain used for the phenotype experiment. Specifically, certain network arrangements behaved differently when incorporated into the two different strains (Guet et al., 2002). This implies that the operational context of synthetic gene networks, i.e., the genetic makeup of the host, plays an important role in determining behavior. Typically, however, such information is not used in gene network simulations, largely due to the fact that we still cannot encode such information.

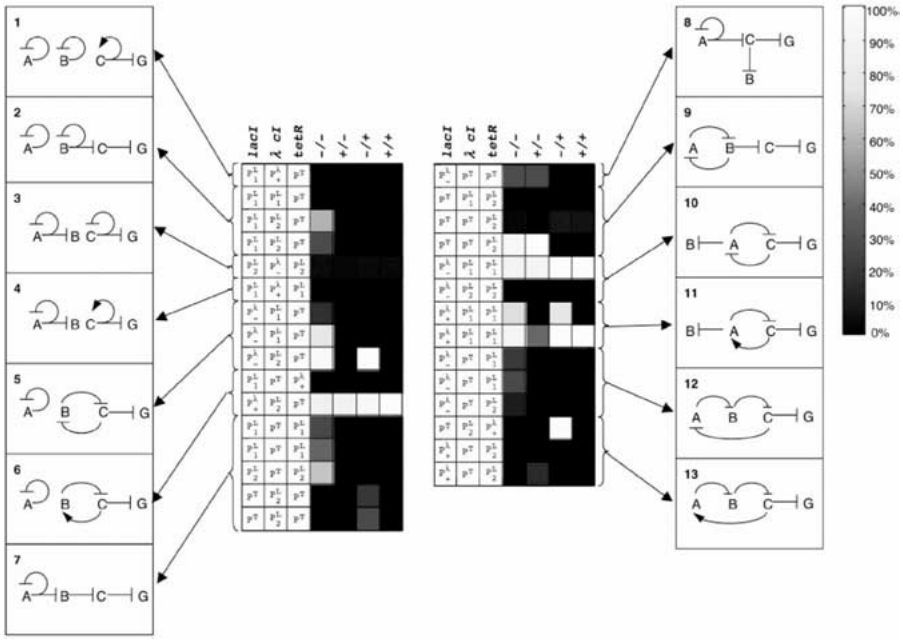


Figure 16. Thirteen network topologies generated by combinatorial synthesis and experimental results showing the response of the different topologies to variations in IPTG and aTc. The gray level of a box represents fluorescence intensity of a colony observed on a plate. Figure adapted from Guet et al. (2002).

The discussion thus far has focused on issues in engineering synthetic gene networks to control the behavior of individual cells. While many challenges still exist in this realm, it is important to also consider the issues of how large ensembles of individual cells can be engineered to work in harmony towards a common goal. Programming cell aggregates will allow us to achieve tasks that are simply not feasible with individual cells. Next, we turn to engineering cell-cell communication systems.

5. Cell-cell communication

While clearly an integral part of eukaryotic multi-cellular systems, cell-cell communication was also discovered in bacteria about three decades ago (Hastings and Nealson, 1977). The ability to engineer both prokaryotic and eukaryotic communication systems with new cell-cell interaction capabilities will be central to the future engineering of multicellular structures. In this section, we report the successful biological implementation of a controlled cell-cell communication system in *E. coli*. The system allows us to control

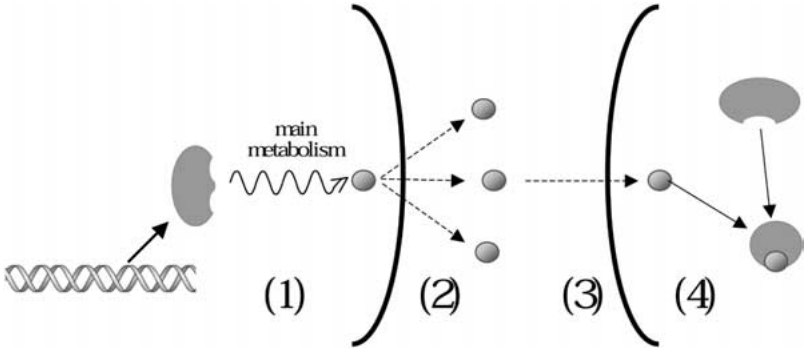
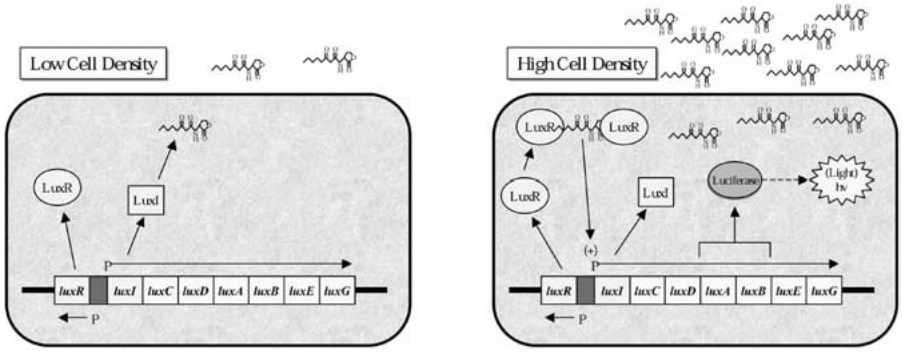


Figure 17. Cell-cell communication schematics: (1) The sender cell produces small signal molecules using certain metabolic pathways. (2) The small molecules diffuse outside the membrane and into the environment. (3) The signals then diffuse into neighboring cells and (4) interact with proteins in the receiver cells, and thereby change signal values.

the extent of a chemical message that a sender cell transmits to a receiver cell, which subsequently activates a remote transcriptional response (Figure 17). Thus far, we have isolated and engineered components from the naturally existing photobacterium *Vibrio fischeri* for the construction of cell-cell signaling systems and their integration with synthetic gene networks in other bacterial species. In the remainder of this section, we describe the quorum sensing mechanisms for cell-cell signaling in *Vibrio fischeri* and discuss our experimental circuits that send and receive messages and are also able to respond to multiple signals.

Quorum sensing is a bacterial communication system that allows cells to sense their own population density through the diffusion of a chemical signal encoded by their genes (Bassler, 1999). The quorum sensing system of certain marine prokaryotes (e.g., *Vibrio fischeri*) is responsible for light organ symbiosis with other animals. The bacteria bioluminesce in a cell density dependent fashion that relies on autoinduction (Engebrecht et al., 1983). They produce a species specific chemical signal molecule, an *autoinducer*, that diffuses into the surrounding media as it is produced (Kaplan and Greenberg, 1985). The autoinducer permeates the cells, and as the cells grow, its concentration increases within the media as well as within the cells.

Vibrio fischeri contains a 9-kb construct comprised of two divergently transcribed operons that contain the genes responsible for bioluminescence (Greenberg, 1997). The left operon, controlled by a constitutive $luxP_L$ promoter is responsible for the production of LuxR protein. The right operon is comprised of a $luxP_R$ promoter followed by the luciferase and *luxI* genes. The $luxP_R$ contains a lux box, a 20-bp inverted palindromic motif, that plays a key role in the transcriptional activation of the operon. Luciferase genes



free living, 10 cells/liter
< 0.8 photons/second/cell

symbiotic, 10^{10} cells/liter
800 photons/second/cell

Figure 18. The lux Operon and quorum sensing: density dependent bioluminescence.

are responsible for light production. They consist of *luxA* and *luxB* genes that encode the α and β subunits of luciferase, as well as the *luxC*, *luxD*, and *luxE* genes, which maintain the aldehyde substrate necessary for light production (Hanzelka and Greenberg, 1996). The structure of autoinducer in *Vibrio fischeri* is *N-(β-ketocaproyl)homoserine lactone* ($3OC_6HSL$) and combines homoserine lactone, an intermediate in amino acid metabolism, with β-ketocaproic acid, a molecule similar to intermediates in fatty acid metabolism.

When the concentration of $3OC_6HSL$ reaches a threshold level of $1 - 10\frac{\mu g}{ml}$, the molecules bind the N-terminal domain of a LuxR protein that consequently unmasks its highly conserved C-terminal DNA binding domain (helix-turn-helix motif). The LuxR/$3OC_6HSL$ complex then binds the lux box within the $luxP_R$ promoter to activate transcription of the luciferase and *luxI* genes. The activity of luciferase results in light production and the increased production of LuxI results in greater concentrations of $3OC_6HSL$. Thus, at low cell densities, only low $3OC_6HSL$ concentrations exist, whereas at high culture densities, for example within a light organ, $3OC_6HSL$ builds up, ultimately resulting in a density dependent induction of bioluminescence (Figure 18).

5.1 *Genetic circuits for engineered communications*

We have isolated, sequenced, and transfered the quorum sensing genetic constructs from *Vibrio fischeri* into *E. coli* in order to engineer communication between bacterial cells (Weiss and Knight Jr., 2000). We constructed genetic circuits that direct one set of cells (the "sender" cells) to synthesize the autoinducer and other circuits that direct another set of cells (the "receiver"

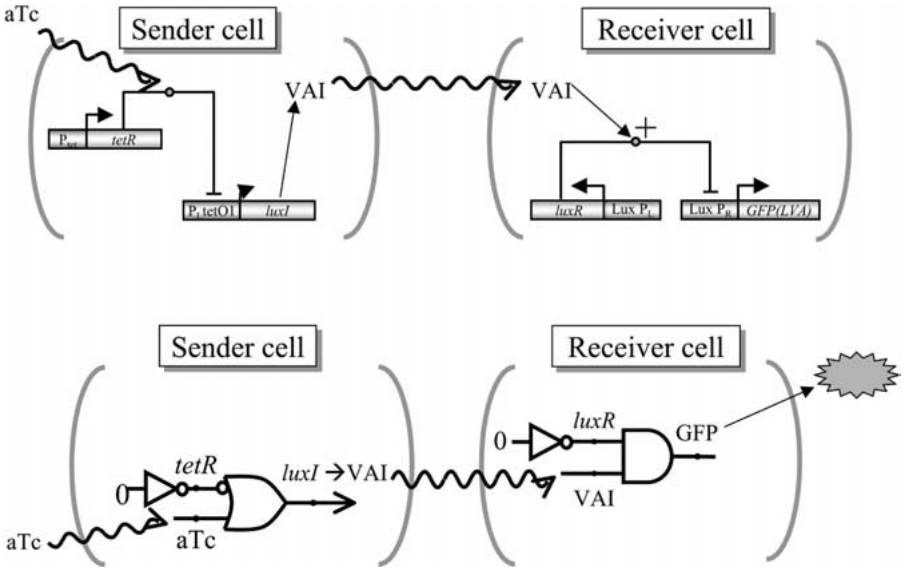


Figure 19. Genetic and corresponding logic circuits for cell-cell communications.

cells) to express a GFP in response to the incoming message (Figure 19). The sender cells contain a *luxI* gene regulated by an aTc-induced $P_L$tetO-1 promoter. The receiver cells contain a *luxR* gene expressed constitutively by the luxP_L promoter and a GFP located downstream of the luxP_R promoter. The LuxR-3OC_6HSL binding complex activates luxP_R.

Figure 20 illustrates the response of the receiver's genetic circuit to increasing levels of 3OC_6HSL. Receiver cell cultures with different levels of purified 3OC_6HSL were incubated at 37°C for five hours, and the median fluorescence data for each culture sample was measured. As expected, increasing levels of autoinducer resulted in corresponding increases in GFP until saturation is reached at approximately $3\frac{\mu g}{ml}$.

5.2 *Visual observation of communication*

We observed the interactions between sender and receiver cells using a fluorescence microscope in order to verify that communication was taking place and to understand the diffusion characteristics of 3OC_6HSL. A small droplet of sender cells was placed in the vicinity of receiver colonies and a brightfield image was captured to mark the location of the various colonies. The communication that followed between the sender and receiver cells was captured by a series of time-lapsed green fluorescence snapshots. The three sample green fluorescence images in Figure 21 illustrate the communication gradient as the 3OC_6HSL diffuses from the sender to the different receiver

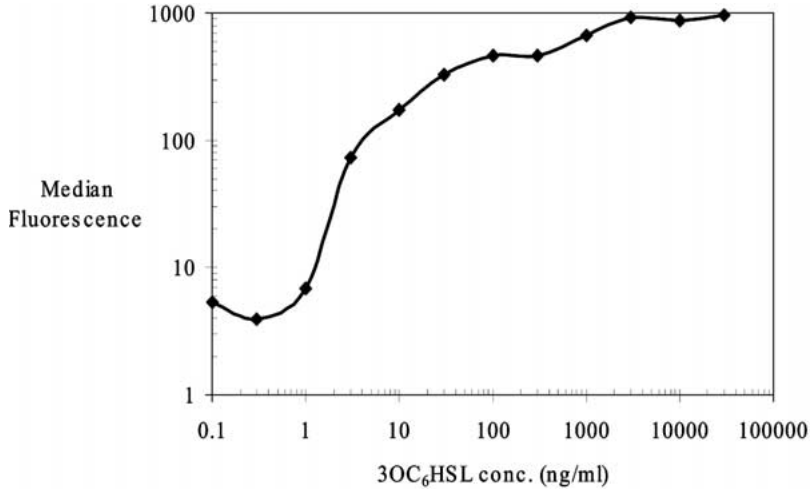
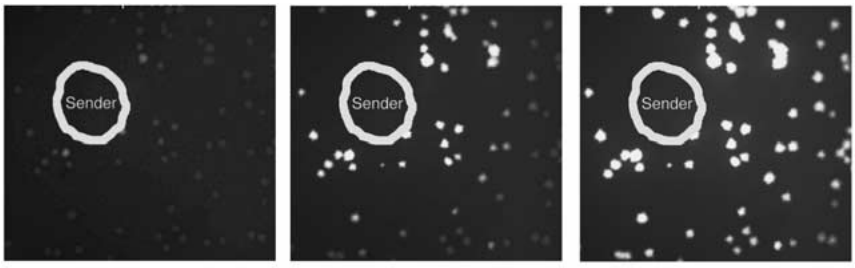


Figure 20. Response of receiver cells to $3OC_6HSL$.



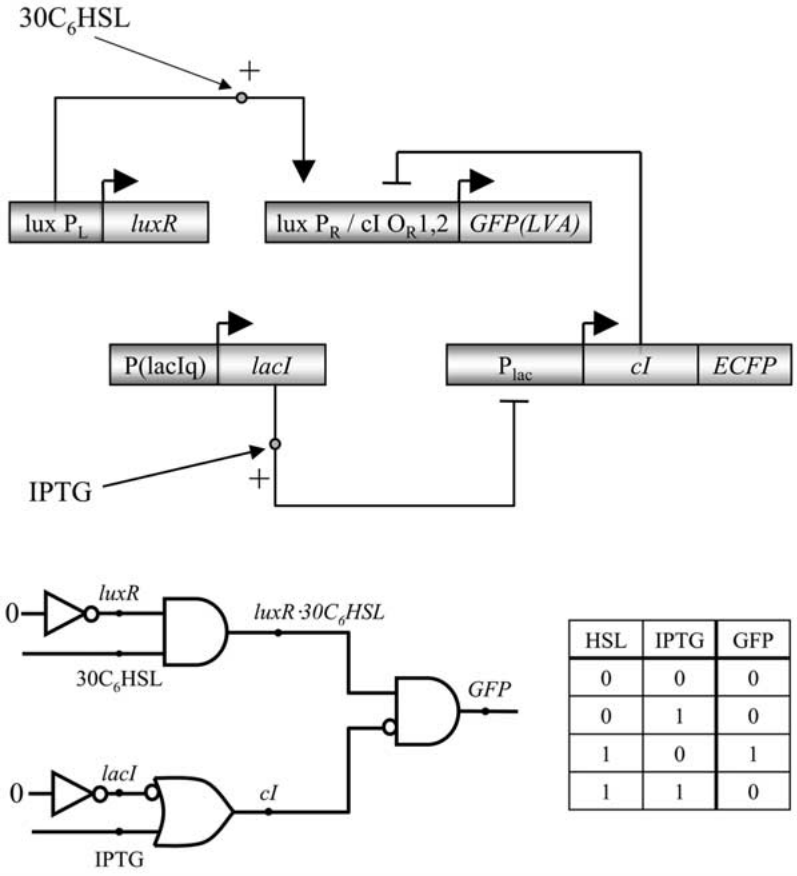
40 minutes 5:00 hours 7:30 hours

Figure 21. Time-series fluorescence images illustrating the response of micro colonies of receivers to communication from nearby senders on a plate.

colonies. The position of the sender cells is denoted by an artificial marker. Based on the fluorescence response, it appears that the autoinducer diffused at a rate of approximately $1\frac{\text{cm}}{\text{hour}}$ through the agar plate.

5.3 *Multi-signal communication*

Multicellular systems use numerous signals to coordinate the behavior of an organism. Even though we have successfully engineered simple communication between bacterial cells, we are still far from developing complex multicellular systems. In order to increase the capabilities of the engineered systems, we constructed genetic circuits that respond to two input signals simultaneously. Specifically, the genetic network responds to both IPTG and $3OC_6HSL$ as shown in Figure 22.



HSL	IPTG	GFP
0	0	0
0	1	0
1	0	1
1	1	0

Figure 22. Circuit to decode two incoming signals, showing the genetic network and corresponding logic circuit and truth table.

To design a circuit that responds to multiple signals, we extended the genetic circuit from Figure 19. *LacI* was inserted downstream of a constitutive promoter p(lacIq) and *cI* was placed under the control of the *lacI* regulated p(lac) promoter. We also constructed a synthetic promoter where the *cI* operator site O_R1 was inserted at the luxP$_R$ +1 transcription site. As in the genetic circuit of Figure 19, increases in 3OC$_6$HSL result in the LuxR-3OC$_6$HSL complex binding the lux box of luxP$_R$, typically resulting in activation. However, when IPTG induces p(lac) expression, CI binds O_R1 and O_R2 downstream of luxP$_R$ and represses transcription. This circuit can therefore process two incoming signals such that GFP is expressed only when 3OC$_6$HSL is present and IPTG is absent. Figure 23 shows median fluorescence values of cells cultures induced with different IPTG and 3OC$_6$HSL concentrations. The

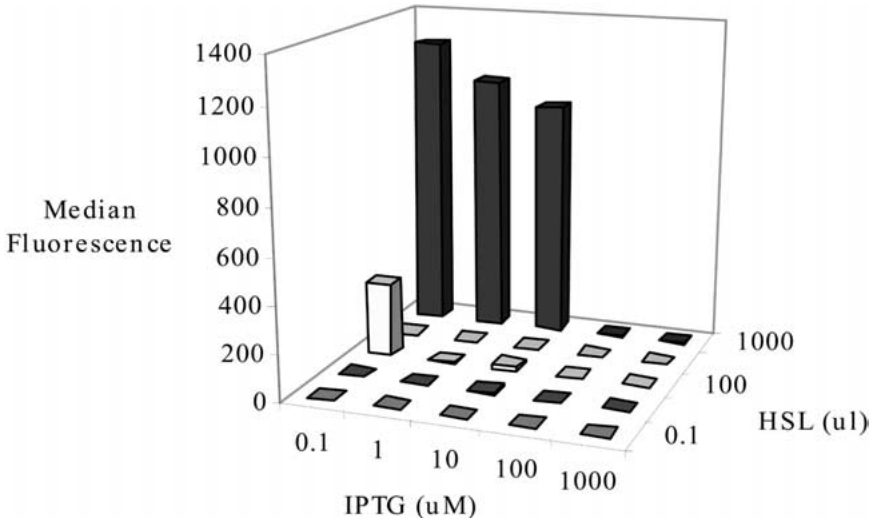


Figure 23. Processing two incoming signals: Median fluorescence values for the multi-signal processing circuit incubated with varying concentrations of IPTG and $3OC_6HSL$.

graphs indicate that the maximum intensity occurs only for low IPTG and high $3OC_6HSL$ concentrations, demonstrating the ability of the circuit to process multiple signals.

6. Signal processing: Band detect

The ability of cells to detect and subsequently react to environmental and internal signals is a principal component of many biological phenomena. Examples include the movement of bacteria toward higher concentrations of nutrients through the process of chemotaxis, detection of photons by the rhodopsin in retinal cells and their conversion to electrical nerve signals, release of fuel molecules in response to hormones that signal hunger, coordinated secretion of virulence factors and degradative enzymes by bacterial cells using quorum sensing molecules, and cell differentiation based on cell gradients. This section describes an example of how cells can be artificially programmed to respond to specific internal and environmental information with the incorporation of synthetic gene networks.

Previous work has demonstrated that bacterial cells can be engineered to detect traces of trinitrotoluene (TNT) in landmines (Burlage et al., 2000). When the bacterial strain *Pseudomonas putida* encounters TNT, it activates genes for the proteins that digest TNT. Researchers inserted a GFP reporter downstream of the promoter regulated by TNT. When the engineered cells

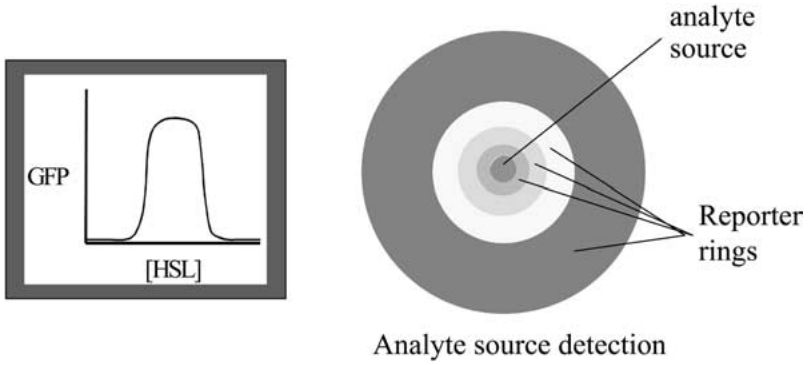


Figure 24. Detecting chemical gradients can be accomplished using a genetic circuit that expresses GFP only when a particular analyte (e.g., HSL) falls within a particular range. If multiple circuits detect different chemical concentration ranges with unique fluorescent colors, they will form separate rings centered around the source.

come in contact with TNT, they express GFP and emit bright fluorescence when excited by ultraviolet light.

We have designed a new genetic signal processing circuit that can be configured to not only detect the presence of a ligand molecule, but also to distinguish between various chemical concentration ranges of the molecule (Basu et al., 2002). For example, consider an analyte source whose location in the environment is unknown. The analyte will be secreted and form a chemical gradient centered around the source. One way to determine the location of the source is to spread the suspected environment with engineered cells that can detect prespecified chemical concentrations. For a prespecified concentration range, these cells will fluoresce in a ring pattern around the source. When detecting multiple ranges, each ring represents a different analyte concentration forming a bullseye pattern (Figure 24).

A proposed circuit reports GFP only when the amount of an analyte $3OC_6HSL$ signal falls within a specified concentration range (Figure 25). The circuit consists of four components: analyte detection, low threshold component, high threshold component, and a negating component. Specifically, $3OC_6HSL$ molecules bind LuxR protein and activate the multicistronic gene transcription of $mRNA_{XY}$. Protein X binds the operator site of P(X) and represses the transcription of $mRNA_{Za}$. This subcircuit acts as the low threshold component since transcription of $mRNAZ_a$ from P(X) is inversely proportional to the $3OC_6HSL$ concentration. Simultaneously, Protein Y binds the operator site of P(Y) and represses the transcription of $mRNA_W$, as illustrated in Figure 26(a). The regulation of W on $mRNA_{Zb}$ from P(W) determines the high threshold component, which exhibits a sigmoidal response to the input $3OC_6HSL$ concentration. The synthesis of Protein Z

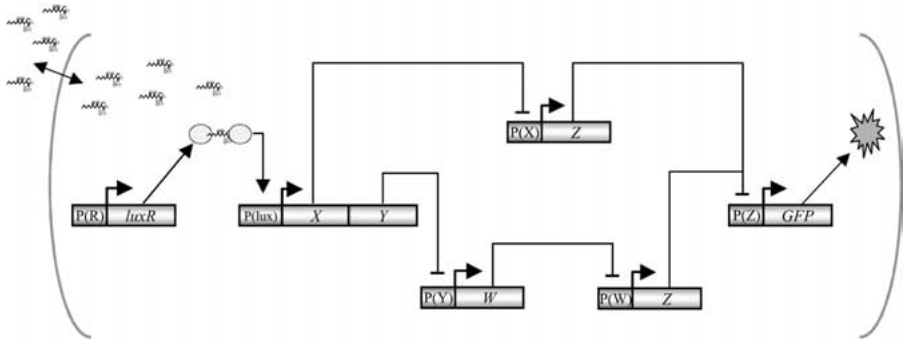


Figure 25. Gene Network for a chemical concentration band detector.

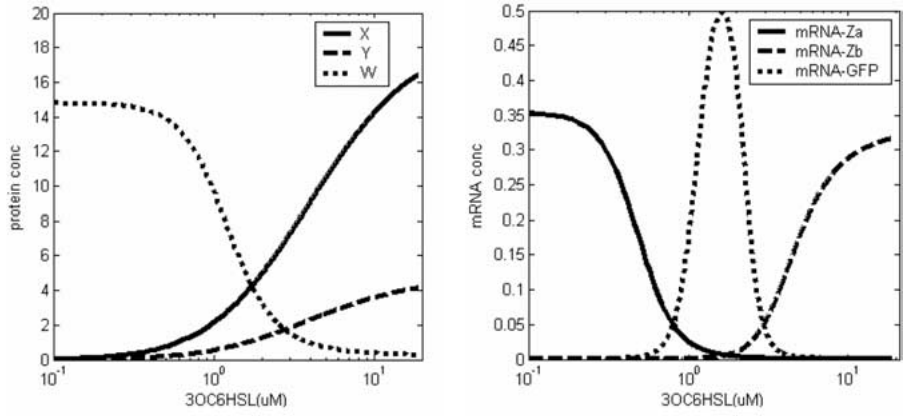


Figure 26. Simulation of the steady state behavior of the band detect circuit components.

from the low and high threshold components determines the band reject component. Protein Z represses the P(Z) promoter and thereby regulates GFP expression, which results in the final band detect response. Hence, GFP expression will be high only when $3OC_6HSL$ is within the specified detection range (Figure 26(b)).

We simulated the dynamic behavior of cells with the band detection circuit in order to predict the diffusion of $3OC_6HSL$ secreted by a sender cell into the media and to characterize the response of the engineered cells to this analyte. The simulations consist of a single sender cell located in the center of a grid and 35 neighboring detector cells. Ordinary differential equations are used to model the intracellular genetic circuit and the intercellular $3OC_6HSL$ diffusion. Figure 27 illustrates the simulated response of the detector cells over time. At $t = 0$, the detector cells are not fluorescing and their positions are shown with light gray squares. At $t = 1$, $3OC_6HSL$ starts to diffuse from the sender cell (shown in dark gray). At $t = 2$, the detector cells closest to

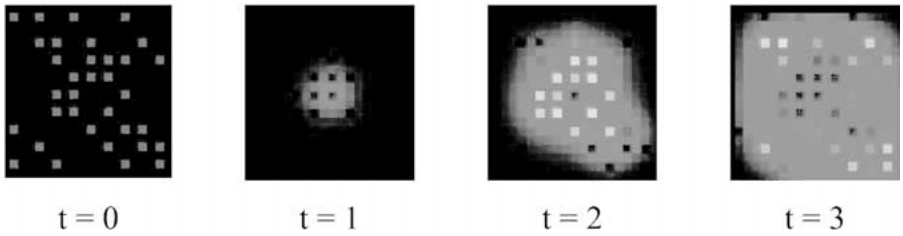


Figure 27. Simulation of the band detect circuit with a sender cell in the center, 35 receiver cells, and diffusion of $3OC_6HSL$.

the sender start to fluoresce in response to the $3OC_6HSL$, as indicated by the black squares. In the last time frame, the distant detector cells fluoresce, while the nearest neighbors to the sender cell have stopped fluorescing. This occurs because the $3OC_6HSL$ concentration in the cells nearest the sender has exceeded the high threshold of detection.

In order to detect a variety of chemical concentration ranges, one can tune the parameters of the genetic components using the same circuit design mechanisms discussed in Section 4 (Weiss and Basu, 2002; Yokobayashi et al., 2002). Figures 13 and 14 reveal how a digital logic inverter can be tuned to exhibit different responses by varying ribosome binding sites and repressor binding affinities. The sigmoidal shape of the response can also serve as the low threshold detection component.

For an experimental prototype that we are working on, we chose CI to be protein X and LacI to be Protein Y. In the first construct, a strong RBS regulates the translation of *CI(LVA)*, a fast decaying version of CI. When $3OC_6HSL$ binds LuxR and the resulting complex activates P(lux) transcription, CI is expressed. In this construct, *gfp(lva)* is inserted downstream of the $\lambda_{P(R)}$ promoter, which is regulated by CI. In the second construct, $luxP_R$ regulates LacI that subsequently inhibits GFP(LVA) expression by repressing the p(lac) promoter. In both constructs, GFP expression is inversely proportional to the concentration of $3OC_6HSL$ (Figure 28). In these experiments, the two different constructs were integrated into different cells and analyzed independently. As seen in the figure, the expression of GFP(LVA) is inhibited in the CI(LVA) system at a much lower $3OC_6HSL$ threshold than in the LacI system. The separation between these responses will be used to determine the final detection range of the complete circuit.

We visually inspected the response of the CI low threshold component to $3OC_6HSL$ secreted by the sender cells. For the experiment, detector cells were plated on a petri dish and incubated until microcolonies were seen under the microscope. A tiny droplet of sender cells that also constitutively expressed a red fluorescent protein was then placed on the plates. A time-

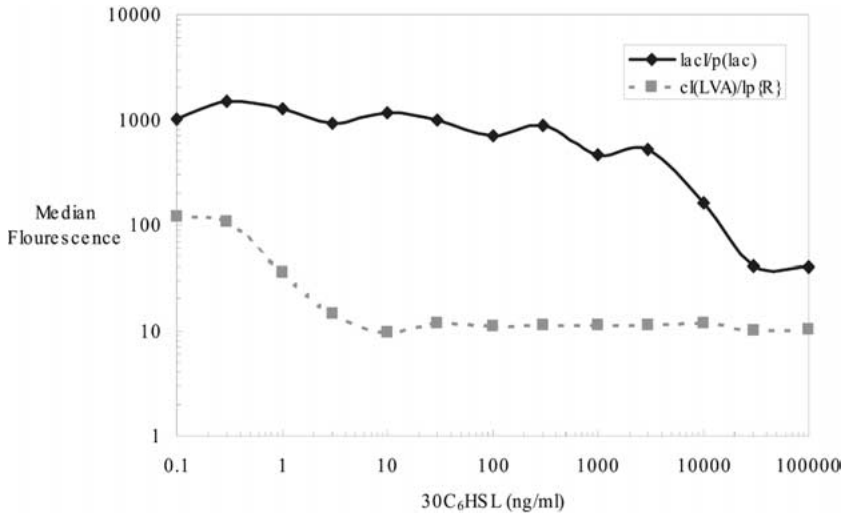
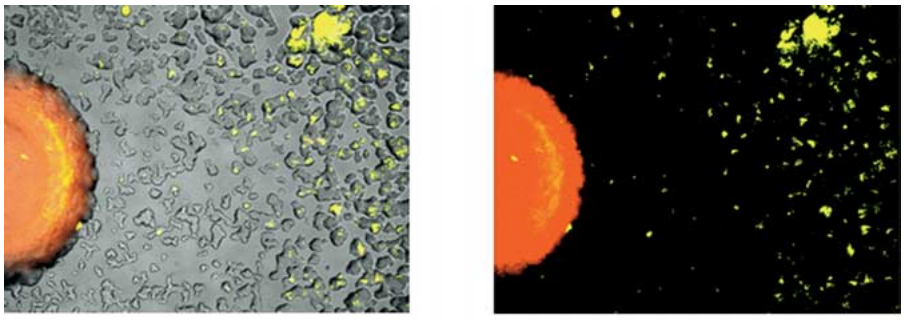


Figure 28. Different low thresholds for band detection, where the less sensitive curve will serve to determine the high threshold.



(a) brightfield and fluorescence (b) fluorescence only

Figure 29. Microscope images of low threshold detection.

lapsed movie of brightfield and fluorescence images was taken to monitor the interaction between the sender and receiver cells. Figure 29(a) shows the brightfield image of the sender cells colony (large semi-circle on the lefthand side) and neighboring receiver cell colonies after a 24 hour incubation. Figure 29(b) is a fluorescence image of the same field of view, superimposing the observed yellow fluorescence of the receiver cells and the red fluorescence of the sender cells. Initially, all receiver cell colonies were fluorescing yellow (not shown). Based on the fluorescence response, it appears that the receiver colonies nearest the sender cells acquired sufficiently high $3OC_6HSL$ concentrations to repress YFP transcription. Colonies farther away from the sender cell maintained their fluorescence because their $3OC_6HSL$ concentra-

tion never reached high levels. The differences in receiver fluorescence based on their distance from sender colonies demonstrate the desired behavior of the low threshold component.

7. Conclusions

We have embarked on a path to develop an engineering discipline for creating synthetic gene networks for modifying and extending the behavior of living organisms. Progress to date includes the characterization and assembly of a genetic circuit component library. These building blocks have been used to implement several prototype circuits that provide the foundation for creating more sophisticated networks. Various design strategies, such as rational design and directed evolution, have been defined and explored. It is likely that the best approach will be a hybrid of several methodologies bringing together expertise from a variety of fields. We will harness our abilities to systematically engineer other types of complex systems such as computers and robots, while exploiting the unique features provided by biological substrates (e.g., evolution). We have also begun to engineer multi-cellular systems using cell-cell communication. In the future, we will program cells to implement *digital* and *analog* computation, both as individual entities and as part of larger cell communities.

While the field holds great promise, we still face a number of challenges. One of the major obstacles is our current inability to devise models and perform simulations that can accurately predict the quantitative behavior of genetic networks. Thus, the difficulty of inferring function from DNA sequence limits our circuit design efficiency. In addition, we must delineate the limits of the new information processing capabilities that can be embedded into cells. Beyond the engineering of individual cells, new programming paradigms are needed to achieve the coordinated behavior of cell aggregates. Biological substrates are constrained by factors such as unreliable computing elements, significant noise, and imperfect communication with limited range. Therefore, our designs must strive to achieve sufficient reliability and reproducibility. Such a myriad of challenges leave much opportunity for future work.

The genetic regulatory structures that are currently being developed will have significant impact once they are integrated with other cell capabilities, including bio-secretion, sensing, and the development of multi-cellular structures for tissue engineering. Much promise also lies in shifting to new cell platforms (i.e., prokaryotic to eukaryotic) that will expand the available arsenal of biological machinery. In addition, improved understanding of the operating principles of naturally occurring systems will guide and inspire our

own designs. As the field of synthetic gene networks develops, we will surely witness its impact on various future applications.

Acknowledgements

Financial support has been provided by the Defense Advanced Research Projects Agency (DARPA) under award No. N66001-02-1-8929 and the National Science Foundation, Biological Information and Storage (BITS) Grant EIA-0130613.

References

Barkai N and Leibler S (1997) Robustness in simple biochemical network. Nature 403: 168–171

Bassler BL (1999) How bacterial talk to each other: regulation of gene expression by quorum sensing. Current Opinion in Microbiology 2: 582–587

Basu S, Karig D and Weiss R (2002) Engineered processing of cells: towards molecular concentration band detection. In: Eighth International Meeting on DNA-Based Computers. Sapporo, Japan

Becskei A and Serrano L (2000) Engineering stability in gene networks by autoregulation. Nature 405: 590–593

Burlage RS, Fisher RL, DiBenedetto J and Maston J (2000) Gene System for the field detection of explosive. Second International Symposium on Biotechnology for Conservation of the Environment

De Jong H (2002) Modeling and simulation of genetic regulatory systems: a literature review. Journal of Computational Biology 9(1): 67–103

Elowitz M and Leibler S (2000) A synthetic oscillatory network of transcriptional regulators. Nature 403: 335–338

Elowitz MB, Levine AJ, Siggia ED and Swain P (2002) Stochastic gene expression in a single cell. Science 297: 1183–1186

Engebrecht J, Nealson KH and Silverman M (1983) Bacterial bioluminescence: isolation and genetic analysis of the functions from *Vibrio fischeri*. Cell 32: 773–781

Gardner T, Cantor R and Collins J (2000) construction of a genetic toggle switch in *Escherichia coli'*. Nature 403: 339–342

Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. Journal of Physical Chemistry 81(25): 2340–2361

Greenberg E (1997) Quorum sensing in gram-negative bacteria. In: ASM News, Vol. 63, pp. 371–377

Guet CC, Elowitz MB, Hsing W and Leibler S (2002) Combinatorial synthesis of genetic networks. Nature 296: 1466–1470

Hanzelka BL and Greenberg EP (1996) Quorum sensing in *Vibrio fischeri*: evidence that S-adenosylmethionine is the amino acid substrate for autoinducer synthesis. J. Bacteriol. 178: 5291–5294

Hastings JW and Nealson KH (1977) Bacterial bioluminescence. Annual Review of Microbiology 31: 549–595

Hendrix RW (1983) Lambda II. Cold Spring Harbor Press, Cold Spring Harbor, NY
Kaplan HB and Greenberg EP (1985) Diffusion of autoinducer is involved in regulation of the *Vibrio fischeri* luminescence system. J. Bacteriol. 163: 1210–1214
Kauffman SA (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22: 437
McAdams HH and Arkin A (1998) Simulation of Prokaryotic genetic circuits. Annu. Rev. Biophys. Biomol. Struc. 27: 199–224
McMillen D, Kopell N, Hasty J and Collin JJ (2002) Synchronizing genetic relaxation oscillators by intercell signaling. Proceedings of the National Academy of Science USA 99: 679–684
Ptashne M (1986) A Genetic Switch: Phage Lambda and Higher Organisms, 2 ed. Cell Press and Blackwell Scientific Publications, Cambridge, MA
Rao CV and Arkin AP (2001) Control motifs for intracellular regulatory networks. Annu. Rev. Biomed. Eng 3: 391–419
Shapiro HM (1995) Practical Flow Cytometry, 3rd ed. Wiley-Liss, New York
von Dassow G, Meir E, Munro E and Odell G (2000) The segment polarity network is a robust development model. Nature 406: 188–192
Weiss R (2001) Cellular computation and communications using engineered genetic regulatory networks. Ph.D. thesis, Massachusetts Institute of Technology
Weiss R and Basu S (2002) The device physics of cellular logic gates. In: NSC-1: The First Workshop of Non-Silicon Computing. Boston, MA
Weiss R, Homsy G and Knight TF Jr. (1999) Toward in-vivo digital circuits. In: Dimacs Workshop on Evolution as Computation. Princeton, NJ
Weiss R and Knight TF Jr. (2002) Engineered communications for microbial robotics. In: DNA6: Sixth International Workshop on DNA-based computers, DNA2000, pp. 1–16. Springer-Verlag, Leiden, The Netherlands
Yokobayashi Y, Weiss R and Arnold FH (2002) Directed evolution of a genetic circuit. Proceedings of the National Academy of Science USA 99: 16587–16591