DIMACS Workshop on Evolution as Computation

Toward *in vivo* Digital Circuits

Ron Weiss, George E. Homsy, and Thomas F. Knight, Jr.

Abstract. We propose a mapping from digital logic circuits into genetic regulatory networks with the following property: the chemical activity of such a genetic network *in vivo* implements the computation specified by the corresponding digital circuit. Logic signals are represented by the synthesis rates of cytoplasmic DNA binding proteins. Gates consist of structural genes for output proteins, fused to promoter/operator regions that are regulated by input proteins. The modular approach for building gates allows a free choice of signal proteins and thus enables the construction of complex circuits. This paper presents simulation results that demonstrate the feasibility of this approach. Furthermore, a technique for measuring gate input/output characteristics is introduced. We will use this technique to evaluate gates constructed in our laboratory. Finally, this paper outlines automated logic design and presents BioSpice, a prototype system for the design and verification of genetic digital circuits.

1. Introduction

We seek to design and build biochemical reaction networks *in vivo* that implement the digital logic abstraction, and are thus capable of carrying out computational functions. This would allow us to fit biological cells with digital "prostheses" that enable the cells to perform user-specified computational processes. Programmable computation in living cells would be an enabling technology for a host of applications such as drug and biomaterial manufacturing, nanomachine assembly, sensor/effector arrays, programmed therapeutics, and as a tool for studying genetic regulatory networks,

Our approach is to use synthesis rates of DNA binding proteins as logic signals. Since DNA binding proteins can function as transcriptional repressors, the effect of one protein on the transcription rate of another can represent the flow of logical information. The simplest logic gate, the inverter, is built from a single operator/promoter region that can be bound by an (input) repressor, fused to a structural gene coding for the output protein. Since the input protein represses

Ron Weiss is supported by DARPA/ONR under contract number N00014-96-1-1228.

George E. Homsy is supported in part by an NSF Graduate Research Fellowship, and by a Merck/MIT Graduate Research Fellowship.

Thomas F. Knight, Jr. is supported in part by DARPA/ONR contract number N00014-96-1-1228 and by DARPA contract number DABT63-95-C130.

transcription of the gene for the output protein, this system implements a digital inverter, provided the steady-state input-output transfer function is sufficiently sigmoidal. Since transcription rates are additive, we can build combinatorial gates from inverters with the same output protein.

In the remainder of this paper we describe related work (Section 2), present and discuss the general approach for implementing gene expression based digital logic (Section 3), describe an example of a chemical reaction model for the digital abstraction and show simulation results (Section 4), introduce a mechanism for quantifying the steady state behavior of gates *in vivo* (Section 5), discuss some issued in microbial circuit design (Section 6), describe BioSpice, a prototype simulator for designing and verifying genetic digital circuits (Section 7), and offer conclusions and avenues for future work (Section 8).

2. Related Work

At least as early as 1974, Roessler and others [**18, 19, 21, 20**] noted the possibility of building universal automata by coupling bistable chemical reactions, and that chemical reaction kinetics share a formal relationship with electronic circuit action. Okamoto *et al.* studied a cyclic enzyme system and showed that it had some properties of a McCulloch-Pitts neuron. In 1991, Hjelmfelt *et al.* [**6**] showed in principle how to construct neural networks from coupled chemical reactions, and determined specific connections for the construction of chemical logic gates. Later, Arkin and Ross [**1**] refined this method to allow use of enzymes with lower binding cooperativity, and applied their model to an in-depth analysis of a portion of the glycolytic cycle.

Recently, McAdams and others [**12, 10, 11**] have constructed mathematical models of various genetic regulatory networks *in vivo*.

Neidhardt and Savageau [**14**] have noted the need for useful high-level logical abstractions to improve our understanding of the integrative molecular biology of the cell.

Monod and Jacob [**13**], Sugita [**22**], Kauffman [**9**], and Thomas [**23**] have all made various and partially successful attempts at describing the global qualitative dynamics of genetic regulatory systems, by simplifying those systems to binary signal levels and pursuing a treatment in terms of boolean networks.

3. Gene Expression Based Logic

3.1. General Approach. Based on the model proposed by Knight and Sussman [**8**], we are developing an engineering discipline for designing and implementing digital logic *in vivo*. We seek a mapping from digital logic circuits into genetic regulatory networks with the following property: the chemical activity of such a genetic network *in vivo* expresses the computation specified by the corresponding digital circuit. Our approach uses the translation rates of repressor proteins as signals, and constructs genetic regulatory elements that constrain the signals to realize the desired logic function.

To build an inverter, select an existing promoter with an operator (repressor binding site), and fuse it to a structural gene for a distinct repressor protein. The steady state translation rate of the "output" protein will decrease monotonically with increasing concentration of the "input" protein. And given the assumptions in Section 3.2, the concentration of the input is linear in its translation rate. Then

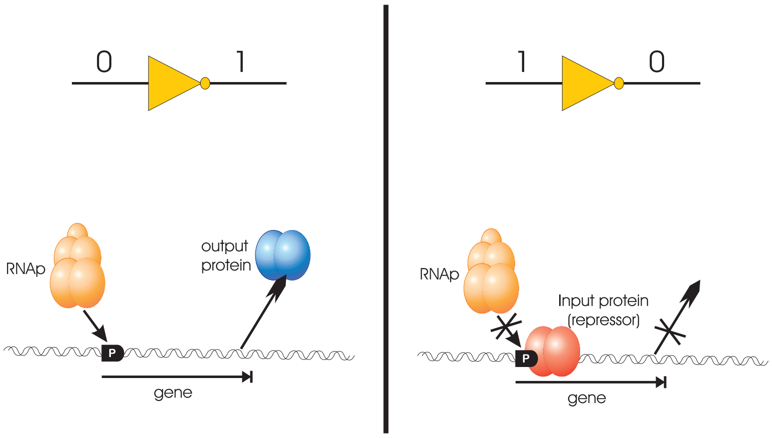


FIGURE 1. The two idealized cases for a biological inverter. If input repressor is absent, RNAP (RNA polymerase) transcribes the gene for the output protein and enables its synthesis. If input repressor is present, no output protein is synthesized.

the two signals, defined by translation rates, are related by a monotone decreasing function. Figure 1 shows the two ideal cases in the truth table of a biological inverter.

NAND gates are built by combining inverters with common output genes. These NAND gates can serve as building blocks for any desired finite state machine, within practical limitations such as the number of distinct signal proteins available.

3.2. The Module Abstraction. Consider a logic element consisting of an input protein, A, acting on an operator, O_A, associated with a promoter P. Let P be fused to a structural gene G_Z coding for the output protein Z.

For abstraction purposes, decouple the transfer function of this logic element into *synthesis* and *decay* stages. The synthesis stage, denoted by S, is the mapping from the input protein concentration, π_A, to the reference translation flux ϕ_P, when the system is in steady state:

$$S : \pi_A \longrightarrow \phi_P \tag{3.1}$$

The reference translation flux is the rate at which a protein coded for by a single structural gene would be synthesized from promoter P. The decay stage that follows, denoted by D, is the mapping from the reference translation flux ϕ_P to the output protein concentration π_Z:

$$D : \phi_P \longrightarrow \pi_Z \tag{3.2}$$

The characteristics of S depend on the thermodynamics of A binding to O_A, the promoter strength of P, and the effectiveness of the ribosome binding site. These are all properties of O_A and P. In general, S will be nonlinear.

In turn, D depends on the degradation rate of the mRNA for Z and the degradation rate of Z itself. These are properties of G_Z. If we assume that mRNA

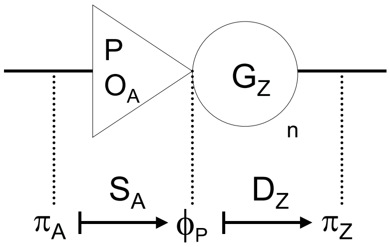


FIGURE 2. Synthesis and decay components of an inverter. Concentrations are denoted by π and translation fluxes by ϕ. S_A is the synthesis mapping determined by operator O_A, and D_Z is the decay mapping of Z determined by G_Z.

degradation and protein degradation are first-order kinetic processes, and that one of them is rate limiting, we can conclude that D is linear. This will be useful in our subsequent analysis.

Figure 2 shows the components of this abstraction. The structural gene G_Z carries a subscript, n, denoting the *cistron count* of G_Z. This count represents the number of distinct ribosome binding sites and copies of the structural gene for Z that are fused to the same promoter. S is subscripted by A to indicate that the translation rate is a function of π_A, the concentration of A. And respectively, D is subscripted by Z to indicate that the concentration of Z is a function of ϕ_Z, the (aggregate) translation rate of structural genes for Z.

Since there are n cistrons coding for Z, the aggregate translation rate of structural genes for Z (the output signal) is just n times the reference translation rate ϕ_P. This yields the transfer function of the inverter:

$$\phi_Z = n \cdot \phi_P = n \cdot D(S(\phi_A)) \tag{3.3}$$

Figure 2 clarifies several points:

- The distinction between protein concentrations and translation fluxes,
- Fluxes determine concentrations, and concentrations determine fluxes,
- S depends only on O_A and P,
- D depends only on G_Z.

The second fact above illustrates "flux/concentration duality": the logical action of the circuit can be characterized either by fluxes or by concentrations. Either forms a complete description of the system, since each determines the other. This notation allows for the fusion of distinct structural genes to a single promoter by connecting the promoter/operator stage to multiple gene stages, one for each output protein of interest.

3.3. Gates: Implementation of Combinatorial Logic. The approach to combinatorial logic is to "wire-OR" the outputs of multiple inverters by assigning them the same output gene. Since the output protein will be expressed in the absence of either input protein, this configuration implements a NAND gate (Figure 3). Since the performance of a NAND gate relies solely on that of its constituent inverters, well-engineered inverters will yield well-engineered combinatorial gates.

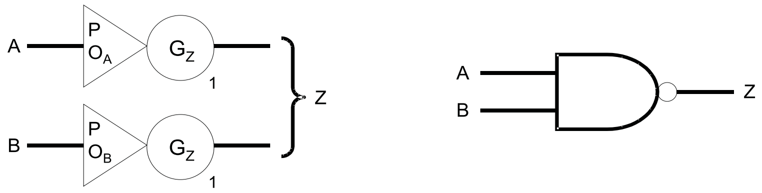


FIGURE 3. Wire OR-ing the outputs of two inverters yields a NAND gate.

3.4. Advantages. Modularity of the network design affords a free choice of signal proteins. Any suitable repressor protein can be used for any signal, where the issue of "suitability" is discussed in Section 6.3. This modularity is necessary for implementing a "bio-compiler": a program that consults a library of repressor proteins and their associated operators and generates genetic logic circuits directly from gate-level descriptions. Contrast this modularity with the method of Hjelmfelt *et al.*, that requires proteins that modify other proteins, and where all signals are protein concentrations. The resulting physico-chemical interdependence of successive logic stages makes simple modularity almost impossible.

In addition, the library of naturally available signal proteins is large. Any repressor protein with sufficiently cooperative DNA binding and that does not interfere with normal cell operation should be appropriate. In our first set of experiments, CI proteins from lambdoid phages will serve as signals. The protein signal library will soon be as large as the family of lambdoid phages. In the future, combinatorial chemistry techniques, combined with a method such as phage display, should yield large libraries of novel DNA binding proteins and corresponding operators.

4. Modeling and Simulation

This section presents a chemical model of a reaction system implementing an inverter, and provides a simulation of its dynamic behavior. The feasibility of the model is explored by testing whether non-trivial circuits composed of such inverters exhibit the desired logical behavior.

4.1. Chemical Reactions Implementing an Inverter. Natural gene regulation systems exhibit characteristics useful for implementing *in vivo* logic circuits. These include transcriptional control, repression through cooperative binding, and degradation of proteins and mRNA transcripts. Table 1 presents one possible chemical model of the reactions involved in such a system. In particular, this model reflects the characteristics of the lambda CI repressor operating on the lambda $\mathrm{O_R}1$ and $\mathrm{O_R}2$ operators.

The repressor protein A represents the input signal, and protein Z represents the output signal. In contrast to Section 3.2, G_Z now denotes the concentration of the *active* form of the structural gene for Z. A structural gene is active only when its associated operator is *unbound* by a repressor.

A_2 and Z_2 denote the dimeric forms of A and Z respectively, and G_ZA_2 and G_ZA_4 represent the repressed (i.e. inactive) forms of the gene. $mRNA_Z$ is the gene transcript coding for Z. RNA_p is RNA polymerase, and $rRNA$ is ribosomal RNA

$$A + A \xrightarrow[\text{dimerization}]{k_{dim(a)}} A_2 \tag{4.1}$$
$$A_2 \xrightarrow[\text{single}]{k_{sngl(a)}} A + A \tag{4.2}$$
$$Z + Z \xrightarrow[\text{dimerization}]{k_{dim(z)}} Z_2 \tag{4.3}$$
$$Z_2 \xrightarrow[\text{single}]{k_{sngl(z)}} Z + Z \tag{4.4}$$
$$G_Z + A_2 \xrightarrow[\text{repress 1}]{k_{rprs(a2)}} G_Z A_2 \tag{4.5}$$
$$G_Z A_2 \xrightarrow[\text{dissociation}]{k_{dis(a2)}} G_Z + A_2 \tag{4.6}$$
$$G_Z A_2 + A_2 \xrightarrow[\text{repress 2}]{k_{rprs(a4)}} G_Z A_4 \tag{4.7}$$
$$G_Z A_4 \xrightarrow[\text{dissociation}]{k_{dis(a4)}} G_Z A_2 + A_2 \tag{4.8}$$
$$A \xrightarrow[\text{decay}]{k_{dec(a)}} \tag{4.9}$$
$$A_2 \xrightarrow[\text{decay}]{k_{dec(a2)}} \tag{4.10}$$
$$Z \xrightarrow[\text{decay}]{k_{dec(z)}} \tag{4.11}$$
$$Z_2 \xrightarrow[\text{decay}]{k_{dec(z2)}} \tag{4.12}$$
$$G_Z A_2 \xrightarrow[\text{decay}]{k_{dec(ga2)}} G_Z \tag{4.13}$$
$$G_Z A_4 \xrightarrow[\text{decay}]{k_{dec(ga4)}} G_Z A_2 \tag{4.14}$$
$$mRNA_Z \xrightarrow[\text{decay}]{k_{dec(mrna)}} \tag{4.15}$$
$$G_Z + RNA_p \xrightarrow[\text{transcribe}]{k_{xscribe}} G_Z + RNA_p + mRNA_Z \tag{4.16}$$
$$mRNA_Z + rRNA \xrightarrow[\text{translate}]{k_{xlate}} mRNA_Z + rRNA + Z \tag{4.17}$$

TABLE 1. Chemical reactions that implement an inverter. A is the input protein and Z the output.

that participates in translation of the $mRNA$. [1] Important aspects of the model include dimerization of the proteins (reactions 4.1 - 4.4), cooperative binding (reactions 4.5 - 4.8), transcription and translation (reactions 4.16, 4.17), and degradation of proteins and mRNA (reactions 4.9 - 4.15).

This relates to the abstraction explained in section 3.2 as follows: If π_A (the *total* amount of A in the system) is fixed, then equations 4.1, 4.2, 4.5-4.8, 4.13, and 4.14 determine G_Z. By reactions 4.16 and 4.17, G_Z determines the reference translation rate ϕ_P from the promoter. This results in the synthesis mapping S. Respectively, ϕ_P determines π_Z (the total amount of Z in the system), given equations 4.3, 4.4, 4.11, 4.12, and 4.15. This results in the decay mapping D.

The kinetic constants used in this simulation (Table 2) are based on the literature describing the phage λ promoter P_R and repressor (cI) mechanism [**17, 5**].

Figure 4 shows the dynamic behavior of the inverter as modeled with the above chemical reactions. The three graphs show the concentrations of the input protein

[1] The simulations in this section assume that the concentrations of RNA_p and $rRNA$ are fixed. Section 5 discusses how to measure the effect of fluctuations in these concentrations, as well as other factors, on the inverter's behavior. Once these effects have been quantified, robust gates can be designed.

$k_{dim(a)}$	8.333	$k_{rprs(a2)}$	66.67	$k_{dec(a)}$	.5775	$k_{dec(ga2)}$	.2887
$k_{sngl(a)}$	.1667	$k_{dis(a2)}$	.2	$k_{dec(a2)}$	.5775	$k_{dec(ga4)}$	.2887
$k_{dim(z)}$	8.333	$k_{rprs(a4)}$	333.3	$k_{dec(z)}$	.5775	$k_{dec(mrna)}$	2.0
$k_{sngl(z)}$	.1667	$k_{dis(a4)}$	.25	$k_{dec(z2)}$	.5775	$k_{xscribe}$	.0001
						k_{xlate}	.03

TABLE 2. Kinetic constants used in the simulations. The units for the first order reactions are 100 sec^{-1} and the units for the second order reactions are $\mu M^{-1} \cdot$ 100 sec^{-1}.

A, the active gene G_Z, and the output protein Z. The concentrations include both the monomeric and dimeric forms.

The reactions proceed as follows: First, π_Z increases until it stabilizes when the expression and decay reactions reach a balance. Then, an externally-imposed drive increases π_A. As a result, the concentration of G_Z decreases as A binds free operator. Then π_Z decreases as no additional Z is synthesized and existing Z is degraded. Finally the drive π_A decreases, A degrades, G_Z recovers, and π_Z once again reaches the HIGH signal range. Note that the gate switching time (measured in minutes for this mechanism) is governed by the rate of recovery of G_Z.

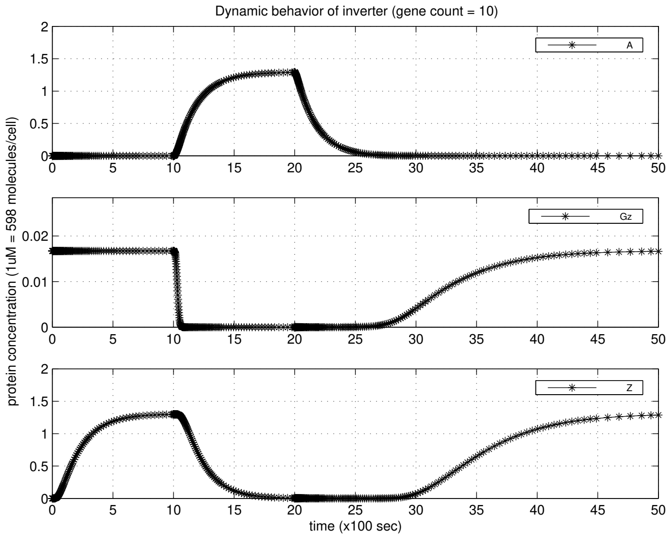


FIGURE 4. The dynamic behavior of the inverter. The top is the input protein, the middle is the active (unrepressed) form of the output gene, and the bottom is the output protein.

4.2. Connections: Analysis of a Ring Oscillator. A ring oscillator is a simple circuit that can help determine the utility of our inverters for building complex logic circuits. The oscillator consists of three inverters connected in a series loop. The simulation results in Figure 5 depict the expected oscillation in protein concentrations, as well as a phase shift between the values. Note, however,

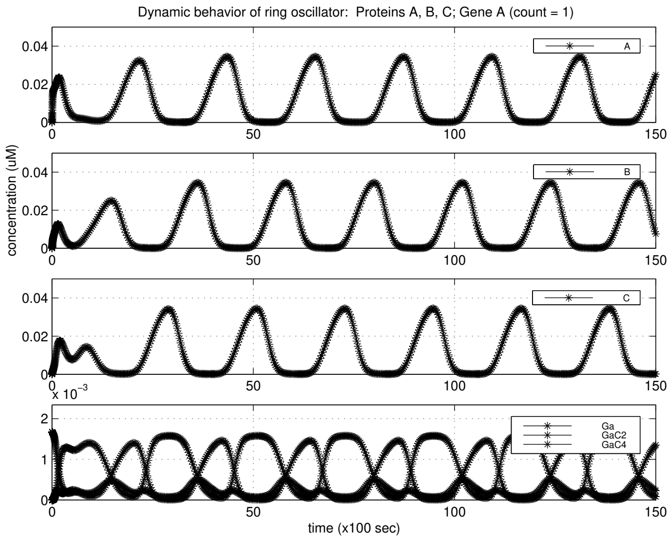


FIGURE 5. Dynamic behavior of a ring oscillator. The top three curves are the outputs of the three inverters. Note the 120° phase shift between successive stages. The bottom shows various repression states of the first inverter's output gene.

that oscillation occurs close to the LOW end of the signal values. This results from the skewed transfer curve that describes the steady state characteristics of the inverter. Sections 5 and 6 discuss this issue in depth.

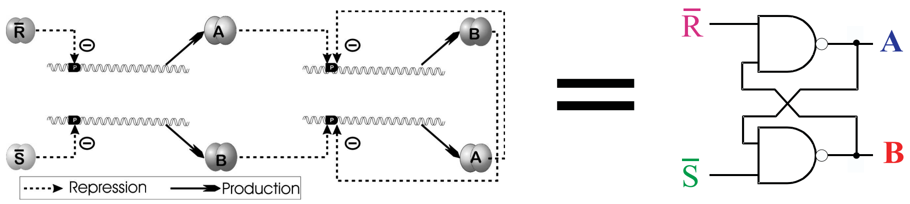


FIGURE 6. Gene logic and schematic representations of an RS latch, used for storage.

4.3. Storage: Analysis of an RS Latch. Another good test circuit is the RS latch, a component for persistently maintaining a data bit. It consists of two cross coupled NAND gates, with inputs $\overline{S}$ and $\overline{R}$ for setting and resetting the complementary output values A and B (Figure 6). The inverters with inputs $\overline{R}$ and B and common output A constitute one of the NAND gates, while the inverters with inputs $\overline{S}$ and A and common output B constitute the other NAND gate. Figure 7 shows the dynamic behavior of this RS latch. As expected, both long and short pulses effectively set and reset the latch.

4.4. Caveats. Published data on kinetic constants is scarce and often imprecise. In several cases, the constants were guessed from published equilibrium

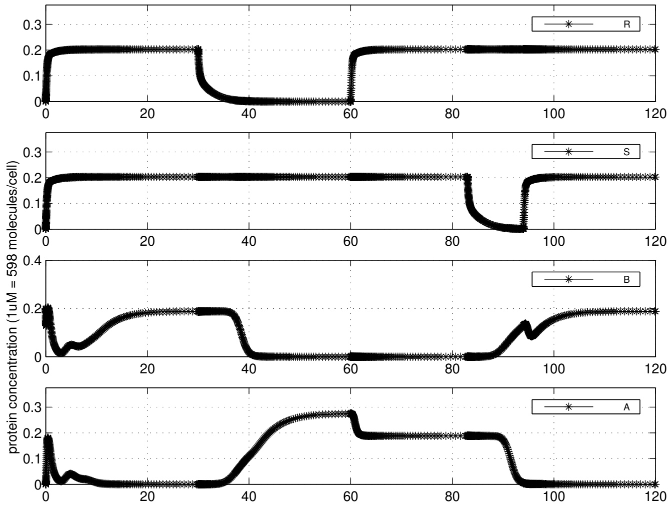


FIGURE 7. Dynamic behavior of the RS latch. The top two curves are the $\overline{\text{reset}}$ and $\overline{\text{set}}$ inputs, respectively. The bottom two curves are the complementary outputs. The initial behavior shows the system settling into a steady state.

constants. This situation is rapidly getting better, and we expect to have more accurate and complete data in the near future.

In cells, typical promoter copy counts correspond to very low concentrations. Therefore, the stochastic noise in concentrations resulting from the discreteness of the transcription reactions can be significant (see *e.g.* Arkin and Ross [**1**]). To decrease this stochastic variance, we will use medium to high promoter copy numbers in our experiments.

5. Measuring Transfer Functions

A transfer function is the relation between the input signal and the output signal of a gate or circuit in steady state. Section 5.1 describes how to measure an individual signal in a genetic circuit, by constructing a probe that measures expression activity *in vivo*. Section 5.2 introduces a mechanism for estimating the transfer function by measuring many different points of the transfer curve. Finally, Section 5.3 discusses how to account for systematic fluctuations and noise inherent in biological systems, by generalizing the transfer function to a *transfer band*. We will use the techniques outlined in this section to measure and verify the characteristics of biological logic circuits *in vivo*.

5.1. Measuring a Signal. Recall that the translation flux ϕ_Z of all genes coding for a protein Z represents a logic signal. Thus the relevant chemical reaction for a signal is the rate of translation of the $mRNA$ product of G_Z into the protein product Z:

$$mRNA_Z + rRNA \xrightarrow[\text{translation}]{k_{xlate}} mRNA_Z + rRNA + n \cdot Z \tag{5.1}$$

where G_z is the active (unrepressed) form of the gene, k_{xlate} represents the rate of translation from $mRNA$ into the protein product, and n denotes the cistron count. Then, assuming this is the only production of Z in the system,

$$\phi_Z \equiv n \cdot k_{xlate} \cdot [mRNA_Z] \cdot [rRNA] \tag{5.2}$$

To measure the signal, insert a reporter protein RP as an additional structural gene, and assume that for the concentrations of interest, RP remains mostly in monomeric form:

$$mRNA_Z + rRNA \xrightarrow[\text{translation}]{k_{xlate}} mRNA_Z + rRNA + n \cdot Z + RP \tag{5.3}$$

$$RP \xrightarrow[\text{decay}]{k_{dec(rp)}} \tag{5.4}$$

Then, the time derivative for the reporter concentration is:

$$\frac{d[RP]}{dt} = k_{xlate}[mRNA_Z][rRNA] - k_{dec(rp)}[RP] \tag{5.5}$$

At equilibrium:

$$0 = k_{xlate}[mRNA_Z][rRNA] - k_{dec(rp)}[RP] \tag{5.6}$$

Since $k_{xlate}[mRNA_Z][rRNA] = k_{dec(rp)}[RP]$, by substitution into 5.2,

$$\phi_Z = n \cdot k_{dec(rp)} \cdot [RP] \tag{5.7}$$

We know n, and can measure $[RP]$ (up to an unknown multiplicative factor) by picking for example a fluorescent protein for RP and measuring its fluorescence. By using the same reporter for each measured signal in the circuit, we obtain approximations of signals all scaled by the same factor.

5.2. Measuring the Transfer Curve of an Inverter. Once an individual signal can be measured, the transfer function of a gate is estimated by measuring many points on the curve. A point on the transfer curve is a steady state relation between the translation flux ϕ_A of the input protein and the translation flux ϕ_Z of the output protein. A point is measured by constructing a system with an unknown but fixed ϕ_A, and measuring ϕ_A and ϕ_Z.

To obtain many points, construct multiple systems yielding various fixed values of ϕ_A, and observe the corresponding values of ϕ_Z. Let ${P_D}^j$ represent a constitutive promoter (i.e. "drive") resulting in a fixed value of ϕ_A, say ${\phi_A}^j$. Let $\mathcal{I}$ denote the transfer function of inverter I. Then, for each drive ${P_D}^j$, the value pair $\left(\frac{{\phi_A}^j}{k_{dec(rp)}}, \frac{\mathcal{I}({\phi_A}^j)}{k_{dec(rp)}}\right)$ can be measured with the reporter RP as described above. This requires two separate experiments, one to measure the drive and one to measure the output. With a set of these points, we obtain the transfer curve of a gate, where all points are expressed in the same (albeit unknown) units.

There are at least three mechanisms for obtaining a variety of drives. First, one can choose different promoters. Second, one can modify the strength of a given promoter region through base-pair substitutions, resulting in different transcription initiation rates. Third, for a given promoter, increasing the cistron count of the gene for the drive yields a multiplicative increase in the drive. Figure 8 illustrates

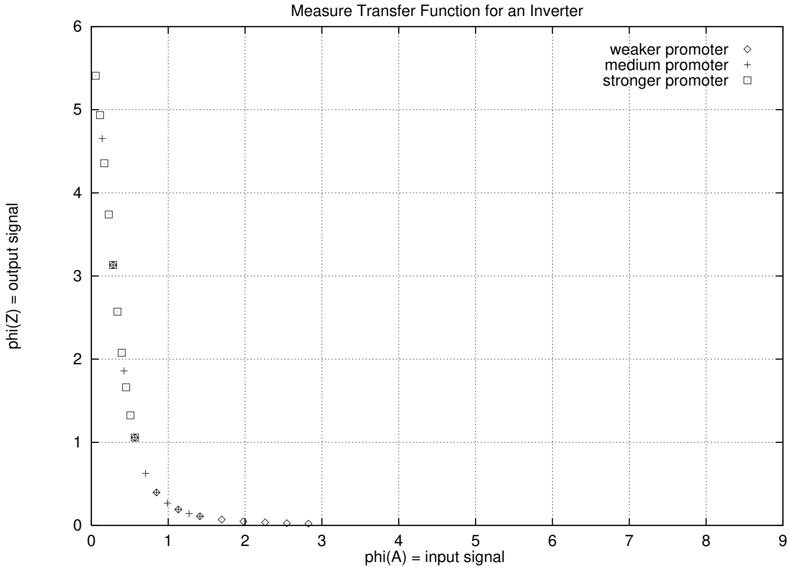


FIGURE 8. Measuring points on the transfer function of an inverter using three promoters, each with ten different cistron counts.

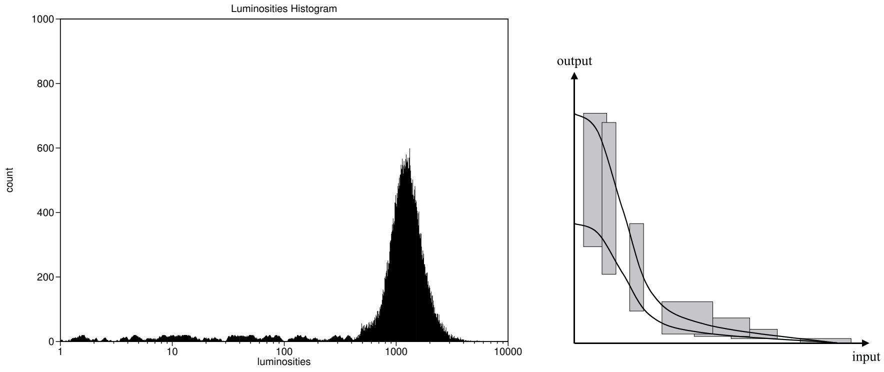
points on an inverter's transfer curve, obtained by simulating thirty different drives. The simulation computes ten points for each of three different promoters with different $\mathrm{RNA_p}$ affinities. For each promoter, the different points indicate the effect of including between one and ten cistrons.

To measure a complete transfer curve, the range of inputs must cover both the LOW and HIGH input ranges. This will require drives with both strong and weak promoters. One does not need to know *a priori* about the characteristics of ${P_D}^j$ to use it for measuring points on the transfer curve. Also, drives with similar characteristics simply add redundancy to the measurements.

To measure more complex circuits, measure the values of the relevant signals by inserting the structural gene for the reporter at the appropriate promoters.

5.3. The Transfer Band: Models *vs.* Reality. While this simple model yields idealized discrete points on the transfer function, actual biological systems exhibit both systematic fluctuations and noise that are not modeled in Section 4. The *transfer band* is a concept capturing these fluctuations. Specifically, it is the mapping from individual input values to the output value ranges exhibited by viable cells.

Flow cytometry [**4**] is a technology useful for quantifying gene expression activities of individual cells. First, a gene coding for a fluorescent reporter (*e.g.* GFP) is fused to the same promoter as the gene of interest. Then luminosity values are measured for individual cells as they flow through a cytometer's capillary, yielding a histogram of luminosities. Figure 9(a) shows such a histogram for a large number of "identical" cells (*i.e.* with the same promoter/reporter construct). The variance results from systematic fluctuations in protein expression rates (*e.g.* due to cell growth cycle and environmental factors), idiosyncratic fluctuations due to



(a) (b)

FIGURE 9. **(a)** A typical flow cytometry histogram of scaled luminosities, showing one dominant peak. **(b)** Approximation of a transfer band from several distinct drives. The measurement of each drive yields a shaded rectangle. The band lies between the two bold curves.

stochastic noise in gene expression, measurement error, and non-viable or damaged cells.

We expect our experiments to yield histograms with one clearly dominant peak. For a drive ${P_D}^j$, ${\phi_A}^j$ is now a distribution. Then, let ${\underline{\phi}_A}^j$ be the minimum value of the drive distribution's peak, and ${\overline{\phi}_A}^j$ be the maximum value of that peak. Cells with values in this range are said to be *operational*. In the same manner, let ${\phi_Z}^j$ represent the corresponding distribution of output values, and let ${\underline{\phi}_Z}^j$ and ${\overline{\phi}_Z}^j$ be the minimum and maximum value of the output distribution's peak.

Then the measurement of the input and output distributions for each drive yields a rectangular region with the lower left corner at $\left({\underline{\phi}_A}^j, {\underline{\phi}_Z}^j\right)$ and the upper right corner at $\left({\overline{\phi}_A}^j, {\overline{\phi}_Z}^j\right)$. If the sample density is high, then the transfer band lies within the union of all such rectangles. Figure 9(b) illustrates the approximation of a transfer band from several flow cytometry measurements.

6. Microbial Circuit Design

The objective of *microbial circuit design* is to take a desired logic circuit and a database of kinetic rates as input, and produce a genetic network that implements the circuit. The design process requires searching the database and assigning suitable proteins to each gate, where the dynamic behavior of the gate depends on these choices. The gates must be robust enough to use a wide variety of proteins with different reaction kinetics.

This section outlines some of the key implementation choices in the design process, defines how to match gate input/output threshold levels, and describes mechanisms to modify the steady-state characteristics of an inverter in order to achieve these levels.

6.1. Implementation Choices. In subsequent publications, we will present detailed analysis of how to design microbial circuits. Some of the key issues are:

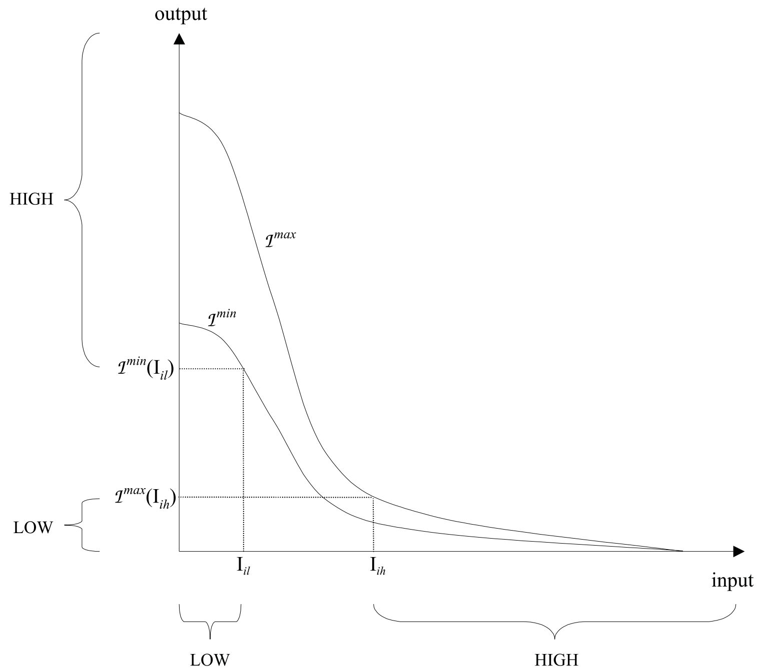


FIGURE 10. HIGH and LOW input ranges for a hypothetical inverter. The transfer band is defined by the two curves.

- *Global gene copy number:* The circuits will be implemented on one or several plasmids. Since high copy number plasmids place a metabolic burden on the cell, while low copy number plasmids may result in large stochastic noise, we intend to use medium copy number plasmids such as *pBR322*.
- *Output proteins:* An output protein must be soluble, bind some known operator site(s), and be inessential for normal cell function. To ensure sufficient gain and noise margins, binding should be highly cooperative (*e.g.* Lambda CI represses using two dimers).
- *Promoter/operator regions:* Operators should bind repressors cooperatively, and promoters should be weak enough to not saturate subsequent gate inputs.
- *Signal threshold levels:* The gate input thresholds must be chosen to provide high gain near the switching threshold, adequate noise margins at the HIGH and LOW signal levels, and balanced transition times.
- *Per-gate cistron count:* The cistron count can be adjusted for each output protein to match threshold levels.

6.2. Matching Thresholds. Transfer functions suitable for implementing digital gates must have LOW and HIGH ranges such that signals in the LOW range map strictly into the HIGH range, and *vice versa.* The strictness of the inclusion reduces noise from input to output. For electronic digital circuits, the LOW and HIGH signal ranges are the same for all gates because the circuit is composed of transistors with identical threshold voltages, spatially arranged. However, in biological digital circuits, the gate components (proteins) have different characteristics depending on their reaction kinetics. Therefore, the designer of biological digital circuits must take explicit steps to ensure that the signal ranges for coupled gates are appropriately matched.

A given transfer band can be defined by a pair of functions. As shown in Figure 10, let $\mathcal{I}^{min}$ be the function that maps an input to the minimum corresponding operational output, and let $\mathcal{I}^{max}$ be the function that maps an input to the maximum corresponding operational output.

If I_{il} and I_{ih} are the input thresholds, then the suitability condition given above can be written as:

$$\begin{array}{llll} [\text{in LOW}] & \langle 0, \mathrm{I}_{il} \rangle & \xrightarrow{\text{into}} \langle \mathcal{I}^{min}(\mathrm{I}_{il}), \mathcal{I}^{max}(0) \rangle & [\text{out HIGH}] \\ [\text{in HIGH}] & \langle \mathrm{I}_{ih}, \infty \rangle & \xrightarrow{\text{into}} \langle 0, \mathcal{I}^{max}(\mathrm{I}_{ih}) \rangle & [\text{out LOW}] \end{array}$$

Consider the case of two inverters, I and J, with J's output coupled to I's input. Then, the coupling is correct *iff*:

$$\begin{array}{rcl} \langle \mathcal{J}^{min}(\mathrm{J}_{il}), \mathcal{J}^{max}(0) \rangle & \subset & \langle \mathrm{I}_{ih}, \infty \rangle \\ \langle 0, \mathcal{J}^{max}(\mathrm{J}_{ih}) \rangle & \subset & \langle 0, \mathrm{I}_{il} \rangle \end{array}$$

Then the following conditions are necessary and sufficient for correct coupling:

$$\begin{array}{rcl} \mathcal{J}^{min}(\mathrm{J}_{il}) & > & \mathrm{I}_{ih} \\ \mathcal{J}^{max}(\mathrm{J}_{ih}) & < & \mathrm{I}_{il} \end{array}$$

6.3. Modifying the Inverter Characteristics. The first step in developing the microbial circuit design process is to design, build, and characterize several inverters. It is likely that these inverters will not match correctly according to the definitions above. Fortunately, there are techniques to adjust them so they are matched for use in complex circuits. These include:

- Modifying the strength of the promoter or the ribosome binding site (RBS) changes the output scaling of an inverter. DNA sequence determinants of promoter and RBS strengths have been studied extensively [**3, 24, 7**]. Figure 11 shows the effect of hypothetical reductions in promoter strength on the transfer functions of an inverter and two inverters in series.
- Modifying the repressor/operator binding affinity changes the input scaling of an inverter and the shape of its transfer function. This is also accomplished via base-pair substitutions, although the effects of these substitutions are different for each repressor/operator pair. Figure 12 shows the effect of hypothetical reductions in this affinity on the transfer function of an inverter.
- Altering the degradation rate of a protein changes the steady state relation between its synthesis rate and its concentration. This can be done on a per-protein basis by changing the few amino acid residues on the C terminus [**2, 15, 16**].
- The simulated transfer functions shown above are not ideal due to the lack of noise margin at the LOW signal level. Autorepression could improve this by limiting the steady state concentration of the output protein to a much lower maximum value. An operator that binds the output protein may be added to the promoter/operator region to accomplish autorepression.

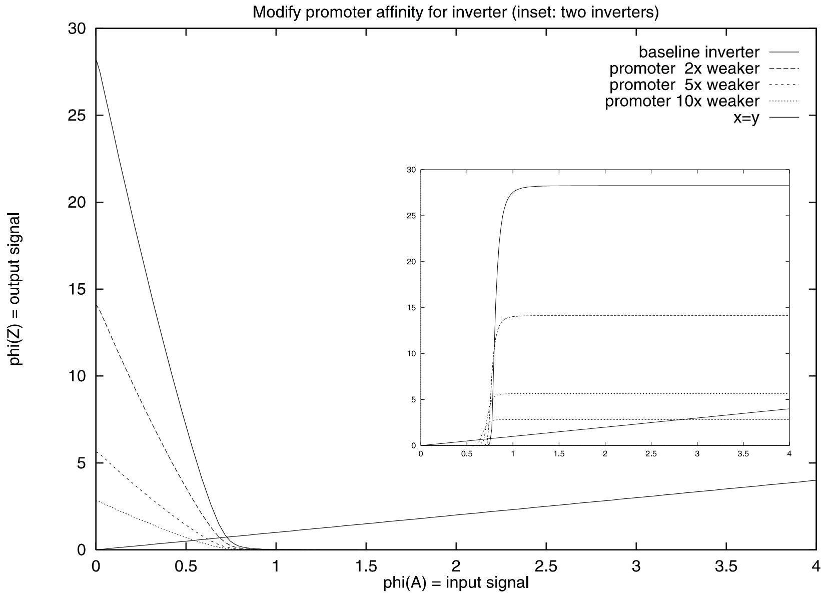


FIGURE 11. The effects of reducing the binding affinity of the RNA polymerase on the transfer functions of an inverter. Inset shows the effects on the transfer functions of two inverters in series. The diagonal lines correspond to input equals output.

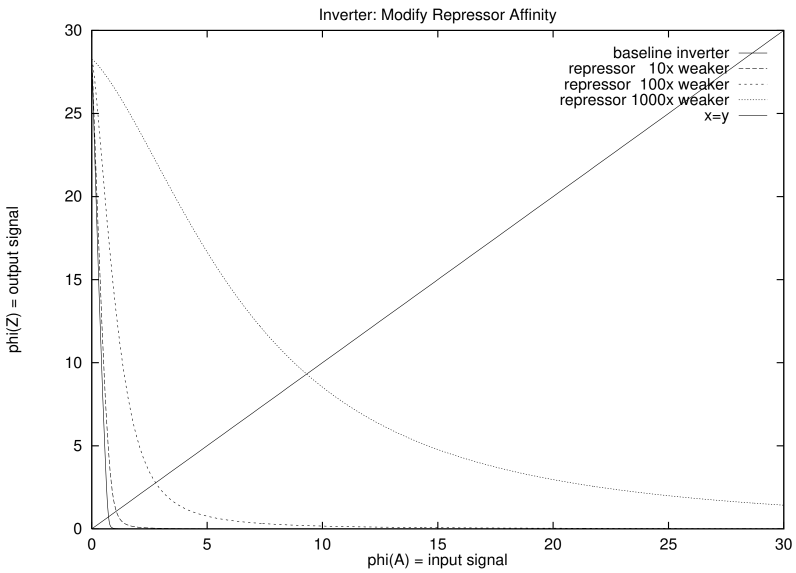


FIGURE 12. The effects of reducing the binding affinity of the repressor on the transfer function of an inverter.

7. BioSpice

BioSpice is a prototype system for simulating and verifying genetic digital circuits. It takes as inputs a network of gene expression systems (including the relevant protein products) and a small layout of cells on some medium. BioSpice then consults a database of reaction kinetics and diffusion rates in order to simulate the dynamic characteristics of the target system. The simulation computes the time-domain behavior of concentration of intracellular proteins and intercellular message passing chemicals.

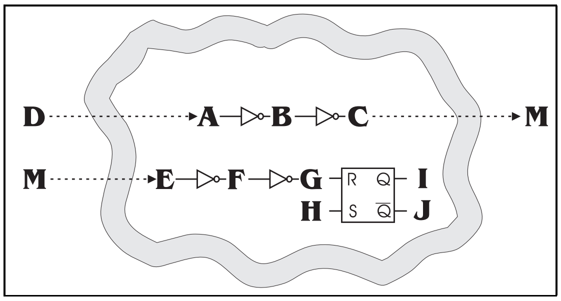


FIGURE 13. Gate level representation of a genetic circuit to accomplish a simple bacterial task.

Consider a simple bacterial task, where upon receipt of a message (represented by inward diffusion of a message passing chemical), a cell communicates to its neighbors and instructs them to set a state bit. Figure 13 represents a genetic digital circuit designed to perform this task. The initiating signal D is a chemical that traverses the cell membrane and results in the presence of protein A in the cytoplasm. This can be achieved with certain signal transduction pathways. The presence of A results in controlled synthesis of C. Notice that the gate with input A can be chosen or adjusted to be sensitive to even small quantities of A. Once a sufficient concentration of A accumulates, C is synthesized and secreted into the surrounding environment as protein M. M diffuses through the medium and serves as a message to neighboring cells. In response to M, the neighbors each set their RS latch, whose output is I.

Figure 14 shows a BioSpice simulation of the above system on a 4×4 grid (representing the medium) with two bacterial cells (heavily shaded squares). The initial condition, depicted in the top-left snapshot, shows that the output of the RS latch (represented by I) is LOW. Then, a drive D is introduced into the environment next to one of the cells, as illustrated in the top-right snapshot. This causes the cell to transmit a message M. Once the other cell receives M (recognizable by the presence of E) it uses G to set the RS latch. Finally, when the drive is removed and the message M decays, the value of I remains latched at HIGH.

8. Conclusions

This paper presents a design paradigm for gene-expression based digital logic implemented *in vivo*. The proposed modular abstraction enables the construction of complex circuits using a library of interchangeable components. Simulation results indicate the feasibility of this paradigm. This paper also presents a measurement

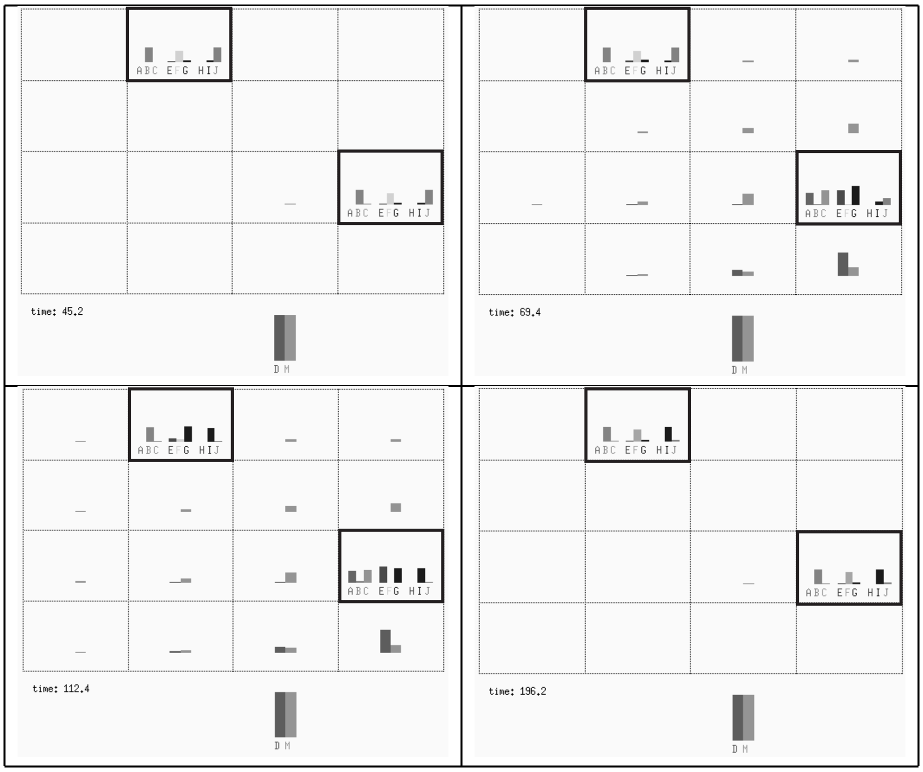


FIGURE 14. BioSpice simulation snapshots of intracellular protein and intercellular message chemical concentrations.

technique for characterizing the steady state behavior of the system components. This technique accommodates systematic fluctuations and noise. Microbial circuit design uses these measurements in matching gates for correct function in complex logic circuits. BioSpice is a prototype tool for simulation and verification of distributed genetic digital systems.

If the initial experiments are successful, future work will concentrate on developing the technology from a simple laboratory model with one or two flip flops, to genetic circuits of several hundred or thousand gates. An important component of this effort is designing new repressors and matching operator sequences, either *de novo* or by altering existing systems. Other related problems include harnessing signal transduction pathways to accomplish environmental sensing and intercellular communication.

References

1. A. Arkin and J. Ross, *Computational functions in biochemical reaction networks*, Biophysical Journal **67** (1994), 560–578.
2. James U. Bowie and Robert T. Sauer, *Identification of c-terminal extensions that protect proteins from intracellular proteolysis*, Journal of Biological Chemistry **264** (1989), no. 13, 7596–7602.
3. David E. Draper, *Escherichia coli and salmonella*, (Frederick C. Neidhardt, ed.), ASM Press, Washington, D.C., 2 ed., 1992, pp. 902–908.

4. Alice Longobardi Givan, *Flow cytometry: First principles*, Wiley-Liss, New York, 1992.
5. Roger W. Hendrix, *Lambda ii*, Cold Spring Harbor Press, Cold Spring Harbor, New York, 1983.
6. A. Hjelmfelt, E. D. Weinberger, and J. Ross, *Chemical implementation of neural networks and turing machines*, Proc. Natl. Acad. Sci. **88** (1991), 10983–10987.
7. M. Thomas Record Jr., William S. Reznikoff, Maria L. Craig, Kristi L. McQuade, and Paula J. Schlax, *Escherichia coli and salmonella*, (Frederick C. Neidhardt, ed.), ASM Press, Washington, D.C., 2 ed., 1992, pp. 792–821.
8. Thomas F. Knight Jr. and Gerald Jay Sussman, *Cellular gate technology*, Unconventional Models of Computation, 1997, pp. 257–272.
9. S. A. Kauffman, Current Topics in Developmental Biology (A. Moscona and A. Monroy, eds.), Academic Press, New York, 1971, pp. 145–181.
10. Harley H. McAdams and Adam Arkin, *Stochastic mechanisms in gene expression*, Proc. Natl. Acad. Sci. **94** (1997), 814–819.
11. ———, *Simulation of prokaryotic genetic circuits*, Annu. Rev. Biophys. Biomol. Struc. **27** (1998), 199–224.
12. Harley H. McAdams and Lucy Shapiro, *Circuit simulation of genetic networks*, Science **269** (1995), 650–656.
13. J. Monod and F. Jacob, Cellular Regulatory Mechanisms, 389–401, Cold Spring Harbor, New York, 1961, pp. 389–401.
14. Frederick C. Neidhardt and Michael A. Savageau, *Escherichia coli and salmonella*, (Frederick C. Neidhardt, ed.), ASM Press, Washington, D.C., 2 ed., 1992, pp. 1310–1324.
15. Andrew A. Pakula and Robert T. Sauer, *Genetic analysis of protein stability and function*, Annual Review of Genetics **23** (1989), 289–310.
16. Dawn A. Parsell, Karen R. Silber, and Robert T. Sauer, *Carboxy-terminal determinants of intracellular protein degradation*, Genes and Development **4** (1990), 277–286.
17. Mark Ptashne, *A genetic switch: Phage lambda and higher organisms*, 2 ed., Cell Press and Blackwell Scientific Publications, Cambridge, MA, 1986.
18. O. Roessler, Lecture Notes in Biomathematics 4 (M. Conrad, W. Guettinger, and M. Dal Cin, eds.), Springer, Berlin, 1974, pp. 399–418.
19. ———, Lecture Notes in Biomathematics 4 (M. Conrad, W. Guettinger, and M. Dal Cin, eds.), Springer, Berlin, 1974, pp. 546–582.
20. P. Roessler, J. Theor. Biol. **36** (1972), 413–417.
21. F. Seelig and O. Roessler, Z. Naturforsch **27** (1972), 1441–1444.
22. M. Sugita, J. Theor. Biol. **4** (1963), 179–192.
23. R. Thomas, *Boolean formalization of genetic control circuits*, J. Theor. Biol. **42** (1973), 563–585.
24. Peter H. von Hippel, Thomas D. Yager, and Stanley C. Gill, Transcriptional Regulation, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, New York, 1992, pp. 179–201.

Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Boston, Massachusetts 02139

E-mail address: {rweiss,ghomsy,tk}@ai.mit.edu