Experimentally driven verification of synthetic biological circuits

Article · March 2012

DOI: 10.1109/DATE.2012.6176468

Experimentally Driven Verification of Synthetic Biological Circuits

Boyan Yordanov[5], Evan Appleton[1], Rishi Ganguly[1], Ebru Aydin Gol[4], Swati Banerjee Carr[2], Swapnil Bhatia[3], Traci Haddock[3], Calin Belta[1,4], Douglas Densmore[1,2,3]
Departments of Bioinformatics[1], Molecular and Cell Biology and Biochemistry[2], Electrical and Computer Engineering[3], and Systems Engineering[4], Boston University, Boston, MA, USA, and Microsoft Research[5], Cambridge, UK
Email: {cbelta, dougd}@bu.edu

***Abstract*—We present a framework that allows us to construct and formally analyze the behavior of synthetic gene circuits from specifications in a high level language used in describing electronic circuits. Our back-end synthesis tool automatically generates genetic-regulatory network (GRN) topology realizing the specifications with assigned biological "parts" from a database. We describe experimental procedures to acquire characterization data for the assigned parts and construct mathematical models capturing all possible behaviors of the generated GRN. We delineate algorithms to create finite abstractions of these models, and novel analysis techniques inspired from model-checking to verify behavioral specifications using Linear Temporal Logic (LTL) formulae.**

I. Introduction

Synthetic bioengineering is an emerging field in which the goal is to extend or modify the behavior of organisms and engineer them to perform new tasks [1]. The bioengineering approach holds tremendous promise in enabling sophisticated computation, control, and manufacturing in a biological cell leading to breakthrough applications [2], [3], [4], [5]. Following the discovery of logic in biology [6], a number of biological *devices*—engineered genetic circuits that function as switches [7], oscillators [8], [9], counters [10], and digital logic-gates [11] have been designed and implemented *in vivo*. Approaches, such as [12], [13] further the use of digital logic abstractions used in electrical engineering to construct genetic regulatory networks (GRNs).

A. Problem

The success in implementing genetic circuits has relied largely on manual trial and error experimentation. Synthetic biologists must account for cell death, crosstalk, mutations, intracellular, intercellular and extracellular conditions, noise and other biological phenomena [14]. A major goal of synthetic biology, however, is also to develop a deeper understanding of biological design principles from the bottom up, by building genetic circuits and studying their behavior *in vivo*. There are several key principles in computationally constructing robust circuits: defining appropriate levels of hierarchical abstraction; delineating principles for standardization and characterization of biological *parts*[1]; and integrating this data while creating a system level circuitry using the parts. Prior work [9] has successfully coupled computational modeling and characterization techniques to build a robust system. However, the computational model was specific to a single topology and cellular context. Building systems that are robust in a variety of application and cellular contexts will require realistic and analyzable computational models, combining a given GRN topology with experimental data. Given a GRN specification, we address the problem of deriving a computational model and obtaining experimental data, and using them to verify the behavior of the genetic circuit.

[1]A *part* is a sequence of DNA with specific function with flanking sequences necessary for concatenating it with other parts.

B. Solution

We demonstrate a novel approach for verifying genetic circuits. We explicitly incorporate experimentally obtained characterization data for the primitive components in the circuit. We use this data to construct a computational model for the composite system. To illustrate our approach, we provide an overview of genetic circuit design, genetic component characterization, a genetic inverter circuit, and its verification.

II. Design

To illustrate the design of a GRN we select two simple designs: a genetic inverter and a genetic NOR gate. These were chosen because they represent useful biological primitives as well as abstractions familiar to the design automation community.

A. Design Background

The "Central Dogma" in molecular biology is the process of protein production from deoxyribonucleic acid (DNA) through the process of DNA *transcription* and *translation*. Figure 1 depicts the stages of the central dogma. DNA serves as the storage molecule of biological information for the construction and function of cells. Transcription is the process during which *mRNA* (messenger ribonucleic acid) is transcribed from DNA by *RNA-polymerase*. The mRNA is then translated to protein by a ribosome molecule according to the genetic code [15].

The principal components in our simplified presentation of the central dogma are Promoters, Terminators, Genes, and Ribosome Binding Sites (RBS). DNA transcription is initiated when RNA polymerase (RNAP) recognizes and binds to the *promoter* sequence and is terminated when the RNAP

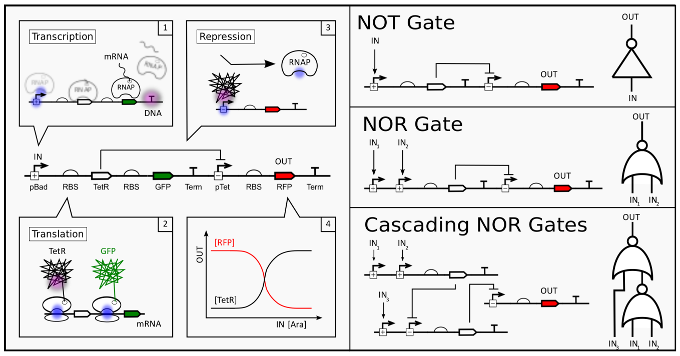

Fig. 1. Boolean logic implemented using genetic regulatory networks.

transcribes the Terminator sequence. The binding of RNAP to the Promoter sequence can either be enhanced or repressed by Transcription Factors (TFs) and small molecules. Thus, the expression of a gene, which is ultimately transcribed and translated to a protein, can be *regulated*—turned "on" or "off"—depending upon the type of the promoter sequence at the beginning of the gene and its regulators. The ribosome recognizes and binds to the Ribosome Binding Site which is an untranslated sequence at the beginning of the mRNA transcript and translates the mRNA into protein.

B. GRN examples

In Figure 1, we show a simple example of a GRN to illustrate how a logic circuit can be constructed from genetic parts. The left panel shows a genetic circuit that functions as an inverter. In this circuit, the promoter labeled pBad acts as the input and the promoter labeled pTet acts as the output. The gene labeled RFP is used as a reporter of the state of the output promoter and the gene labeled GFP is used as a reporter of the input promoter. The pBAD promoter is an inducible promoter which is induced when Arabinose binds to the protein AraC. Thus, a specific concentration of Arabinose in the cellular environment acts as the "high" logical input for this circuit. In its presence the promoter is turned "on" intiating the transcription of the *tetR* gene and its reporter, the GFP gene. The TetR mRNA is translated to the TetR protein, which in turn represses the pTet promoter, silencing the expression of the RFP gene. We use the absence of RFP as our logical low output signal. In the absence of Arabinose in the cellular environment, which we use as the logical "low" input, the input promoter is turned off, the *tetR* gene is not transcribed, and the pTet promoter is not repressed. Consequently, the RFP gene is highly expressed, which we interpret as our "high" output signal. The full relationship between the input Arabinose level and the output RFP level is captured by a *transfer function* as shown in figure 1-4. While the logical abstraction uses only high and low thresholds, the actual relationship follows a nonlinear curve known as a Hill function [16]. The right panel of figure 1 shows other examples of logic gates implemented as genetic circuits. In the middle figure, an additional promoter creates an OR gate which, when coupled with the inverter, creates a NOR gate [11]. The lower figure shows an optimized (with respect to DNA primitives) cascade of two NOR gates.

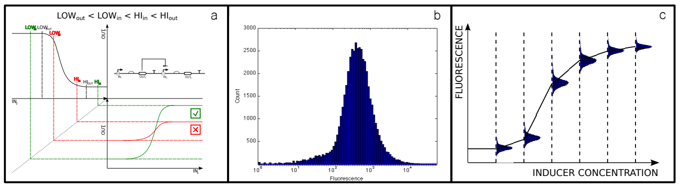

Fig. 2. (A) Compatible part matches: In the case of our inverter circuit, the output range between "on" (HI in) and "off" (LOW in) states of the pBad promoter encompass the full range of the "on" (LOW in) and "off" (HI in) states for the repressible promoter, pTet. In cases where this relationship is not true, the circuit will fail. (B) Characterization data is measured in arbitrary fluorescent units. (C) The full distribution fluorescent output at each level of induction is recorded.

C. User-specifications for Genetic Circuits

The specification of a GRN begins with a functional description capturing the logical behavior. This description is then synthesized into a collection of biological parts which capture this behavior. To perform this synthesis we can use techniques familiar to the design automation community. Specifically, we use a tool called "Cello" *(Cell-Logic)* [17] which transforms continuous assignment statements in Verilog to a Directed Acyclic Graph (DAG). This DAG undergoes a technology transformation to a single logic family (e.g. NOR2) using standard logic synthesis techniques. The back-end synthesis tool then uses the NOR-gate and inverter biological motifs described in Sec. II-b to "technology map" [18] the DAG, resulting in a GRN. Databases curated by our group then provide actual DNA parts for these motifs as a final "covering" step. LTL specifications [19] are then used to specify the behavior and relevant cellular dynamics the circuit is designed for. A formalism introduced in Sec. IV, coupled with Characterization data of the parts (Sec. III) can then be used to formally verify the behavioral specifications of the generated GRN.

III. Characterization

To satisfy a given functional Verilog specification, Cello first generates an abstract genetic circuit. To instantiate a physically constructible circuit, Cello primarily uses two types of parts: promoters—which can be turned on and off—and regulators—which switch the promoters on and off. The efficacy with which a regulator controls a promoter depends on the specific sequence of the promoter and the probability of binding events among one or more regulators and the promoter sequence. Because precise models for any given promoter-regulator interaction are not available, theoretically predicting whether the set of promoters and regulators chosen for a genetic circuit will work together and exhibit the functional behavior specified is not a trivial task.

For a given circuit and a given choice of parts, we construct a Linear Temporal Logic (LTL) specification describing the expected dynamic behavior, in terms of the concentrations of the regulators and reporters used in the circuit. This approach is advantageous because the transfer function of a promoter relating its regulator input to its output can be obtained experimentally. If the transfer function for each regulator-promoter used in the circuit is known, then the LTL specification can be quantitatively verified.

To obtain the transfer function of a promoter-regulator pair, we first construct a characterization circuit: a DNA sequence containing the promoter, with the sequence coding for the gene that produces fluorescence, downstream of it. Thus, as the promoter is regulated to different levels of transcription, different levels of fluorescence will be observed. By normalizing to the same fluorescence reporter gene, one can obtain comparable transfer curves for various promoter-regulator pairs. In our example inverter circuit, characterizing the pBad promoter involves partitioning a population of cells containing the characterization circuit into k parts, and subjecting part i to the regulator at concentration c_i. The cell culture is allowed to fluoresce, and then the fluorescence level is measured using a flow cytometer.

IV. Approach

Starting from a high level description, the candidate GRNs realizable from available parts are proposed automatically as described in Sec. II-C. In general, not all these GRNs behave "as required" but selecting the functionally correct ones experimentally is infeasible. Instead, we automatically construct a mathematical model of each GRN from part characterization data (Sec. III) and analyze it from high level behavioral specifications expressed in LTL—a formalism we briefly introduce in Sec. IV-B. In this section, we review our model construction (Sec. IV-A) and analysis (Sec. IV-C) procedures and direct the interested reader to [20] and [21], respectively, for additional details.

A. Model Construction

We model a simplified mechanism of gene expression, which captures transcriptional regulation but considers only genes and promoters as parts[2] (see Sec. II-A). We assume that a gene g is expressed from promoter p at rate β_p (capturing both transcription and translation) to make protein, which degrades at rate α_g. The protein concentration is denoted by x_g and is bounded in a physiologically relevant range $x_g^{min} \leq x_g \leq x_g^{max}$ (usually, $x_g^{min} = 0$). We express the dynamics of protein concentration as

$$x_g(k+1) = \alpha_g x_g(k) + \beta_p, \qquad (1)$$

where, for each gene (protein) g, α_g has a fixed value but β_p varies in a range $B_p(x_{g'}) \subset \mathbb{R}$, which is a function of the regulator[3] concentration $x_{g'}$. This range is unknown and must be computed from the available promoter characterization data.

The experimental data collected as described in Sec. III (Fig. 2) related the measured concentration of an inducer or a transcriptional regulator $\tilde{x}_{g'}$ for promoter p to the measured steady-state concentration $\tilde{x}_g$ of a protein expressed from p (*i.e.* the data set consisted of a set of measurements $(\tilde{x}^i_{g'}, \tilde{x}^i_g), i = 1, \ldots, n$)). Assuming that the system parameters within individual cells are fixed and that the degradation rates of all proteins are known, each steady-state concentration $\tilde{x}^i_g$ is converted to a measurement of the expression rate from promoter p using Eqn. (1) as $\tilde{\beta}^i_p = (1-\alpha_g)\tilde{x}^i_g$. The data set $D_p = \{(\tilde{\beta}^i_p, \tilde{x}^i_{g'}), i = 1, 2, \ldots\}$ characterizes the rate of expression from promoter p at different regulator concentrations (see Fig. 3-a,b).

In the following, we focus on the construction of a set

$$\bar{B}_p = \{(\beta_p, x_{g'}) \mid x_{g'} \in [x^{min}_{g'}, x^{max}_{g'}], \beta_p \in B_p(x_{g'})\}, \qquad (2)$$

which allows us to compute the range of expression rates $B_p(x_{g'})$ at an arbitrary $x_{g'}$ as the slice of $\bar{B}_p$ at $x_{g'}$ (*i.e.* $B_p(x_{g'}) = \{\beta_p \mid (\beta_p, x_{g'}) \in \bar{B}_p\}$). By constructing the tightest $\bar{B}_p$ that contains all experimental measurements (*i.e.* $D_p \subset \bar{B}_p$), we guarantee that the model we identify can reproduce all observed behavior but our subsequent analysis in Sec. IV-C is not overly conservative.

To capture the nonlinearity of gene regulation we introduce a set of thresholds[4] $\theta^i_{g'}$, such that $\theta^1_{g'} = x^{min}_{g'}, \theta^{n_{g'}}_{g'} = x^{max}_{g'}$ and $\theta^i_{g'} < \theta^{i+1}_{g'}$ for all $i = 1, \ldots, n_{g'} - 1$. For each regulator concentration region (*i.e.* when $\theta^i_{g'} < x_{g'} < \theta^{i+1}_{g'}$), we construct a trapezoid $\bar{B}^i_p$ that has the two thresholds as its bases and contains all expression rates observed in that region (see Fig. 3-b). We define $\bar{B}_p = \bigcup_{i=1}^{n_{g'}-1} \bar{B}^i_p$ and, given regulator concentration $x_{g'}$ such that $x_{g'} = \lambda\theta^i_{g'} + (1-\lambda)\theta^{i+1}_{g'}$ for some $i = 1, \ldots, n_{g'} - 1$ and $\lambda \in [0, 1]$,

$$B_p(x_{g'}) = \lambda B_p(\theta^i_{g'}) + (1-\lambda) B_p(\theta^{i+1}_{g'}), \qquad (3)$$

where $B_p(\theta^i_{g'})$ and $B_p(\theta^{i+1}_{g'})$ are induced by the construction of $\bar{B}_p$.

Remark 1: The rate of expression from a *constitutive* promoter p (a promoter that is not regulated) is uncertain and varies in a range that does not depend on the concentration of other proteins (*i.e.* $\beta_p \in B^c_p \subset \mathbb{R}$ in Eqn. (1)). If a data set $D^c_p = \{\tilde{\beta}^i_p, i = 1, \ldots, n\}$ of experimentally measured expression rates is available (Sec. III), this range is simply $B^c_p = [min(D^c_p), max(D^c_p)]$. Furthermore, a promoter p regulated by an *external inducer* can be treated as constitutive. The transfer function for p is reconstructed as before (*i.e.* treating all measurements as if p was regulated by a transcription factor) but the set of thresholds is induced from the data (Fig. 3-a). Let x_I denote the concentration of the inducer, in which case $\beta_p \in B_p(x_I)$ can be computed through Eqn. (3) as before. While the exact value of x_I might be hard to control experimentally, it can be restricted to a range $x_I \in [x^{min}_I, x^{max}_I]$. Therefore, $B^c_p = [min(\bar{B}_p), max(\bar{B}_p)]$, where $\bar{B}_p = \{\beta_p \in B_p(x_I) \mid x_I \in [x^{min}_I, x^{max}_I]\}$ (Fig. 3-a).

Given a genetic circuit, let G and P denote its set of genes and promoters, respectively, where $N = |G|$ is the circuit size.

[2]For modeling purposes, we assume that each gene and promoter contain a terminator and an RBS, respectively, which is consistent with the more detailed mechanism described in Sec. II-A.

[3]For simplicity of presentation, we assume that a promoter can have only a single regulator but our approach can be easily extended to handle promoters with multiple regulators.

[4]Computing these thresholds is not the focus of this paper but related methods are available [22]. Here, we implement a sampling procedure where, out of a number of randomly generated thresholds, we select the subset of a given size that minimizes the volume of $\bar{B}_p$.

For notational simplicity, we assume that for $i = 1, \ldots, N$, gene $g_i \in G$ is expressed from promoter $p_i \in P$, which is either constitutive or regulated by the protein produced by gene $g'_i \in G$. The hyper-rectangle

$$\mathcal{X} = [x_{g_1}^{min}, x_{g_1}^{max}] \times \ldots \times [x_{g_N}^{min}, x_{g_N}^{max}] \quad (4)$$

is the feasible state space of the circuit, where each $x \in \mathcal{X}$ is a vector of the concentrations of all proteins (*i.e.* $x = (x_{g_1}, \ldots, x_{g_N})$). $\mathcal{X}$ is partitioned by the thresholds $\theta_{g'}^i, i = 1, \ldots, n_{g'}$ of all regulators $g' \in G$ into a number of hyper-rectangular regions. Given a state $x \in \mathcal{X}$, the overall system dynamics are given by[5]

$$x(k+1) \in Ax(k) + B(x(k)), \quad (5)$$

where A is the diagonal matrix of degradation rates $A = diag(\alpha_{g_1}, \ldots, \alpha_{g_N})$ and $B(x) = B_1(x_{g'_1}) \times \ldots \times B_N(x_{g'_N})$. For a constitutive promoter p_i, we have $B_i(x_{g'_i}) = B_{p_i}^c$ (see Remark 1) and, if p_i is regulated, then $B_i(x_{g'_i}) = B_{p_i}(x_{g'_i})$ is computed according to Eqn. (3). Given an initial state $x(0) \in \mathcal{X}$, the concentrations of the proteins evolve over time according to Eqn. (5) to produce an infinite *trajectory* $x(0), x(1), \ldots$ where $x(k) \in \mathcal{X}$ is the state at step $k = 1, \ldots$. This model can reproduce all experimental data used for part characterization and its structure allows the construction and formal analysis of finite abstractions as discussed in Sec IV-C

B. Temporal Logic Specification

To verify that a given GRN behaves "as required", we first formalize the expected behavior of its model (5) as a linear temporal logic formula. Temporal logics were originally developed for specifying and verifying the correctness of digital circuits and computer programs [19]. However, due to their expressiveness, resemblance to natural language and the existence of automated, off-the-shelf model-checking algorithms, temporal logics are also gaining popularity in several other fields, including the specification of behavior of genetic networks [20], [23].

Given system (5), we define a set of atomic propositions Π as a set of linear inequalities

$$\Pi = \{\pi_i, i = 1, \ldots, K\}, \pi_i = \{x \in \mathcal{X} \mid c_i^T x + d_i \le 0\}. \quad (6)$$

Each atomic proposition π_i partitions the feasible space $\mathcal{X}$ into a satisfying and violating subset for π_i. Given a state $x \in \mathcal{X}$ we write $x \models \pi_i$ if and only if $c_i^T x + d_i \le 0$ (*i.e* x satisfies π_i). A trajectory $x(0), x(1), \ldots$ of (5) produces an infinite word $w(0), w(1), \ldots$ where $w(k) = \{\pi \in \Pi \mid x(k) \models \pi\}$ is the set of propositions satisfied at step k.

To specify temporal and logical properties of the trajectories of (5), we use Linear Temporal Logic [19]. Informally, LTL formulas over Π are inductively defined by using the standard Boolean operators (*e.g.,* $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction)) and temporal operators, which include $\bigcirc$ ("next"), U ("until"), $\Box$ ("always"), and $\Diamond$ ("eventually"). LTL formulas are interpreted over infinite words, as those generated by system (5). For example, the word $w(0), w(1), \ldots$ where $w(0) = \{\pi_1, \pi_2\}, w(1) = \{\pi_1, \pi_2, \pi_3\}$, and $w(2), w(3), \ldots = \{\pi_1, \pi_4\}$ satisfies formulas $\Box\pi_1, \Diamond\pi_3, \Diamond\Box(\pi_1 \wedge \pi_4)$, and $\pi_2 U \pi_4$ but violates $\Box\pi_2$ and $\Diamond\pi_5$. We say that a trajectory $x(0), x(1), \ldots$ satisfies an LTL formula ϕ if and only if the corresponding word $w(0), w(1), \ldots$ satisfies ϕ. System (5) satisfies ϕ from a given region $X \subseteq \mathcal{X}$ if and only if all trajectories originating in X satisfy the formula.

C. Formal Analysis

Model-checking algorithms capable of verifying the correctness of finite state systems against LTL specifications exist [19], but such models are often too simple to accurately capture the dynamics of GRNs. In Sec. IV-A we used part characterization data to construct infinite state models, which are more realistic but cannot be verified directly. Here, we review an approach for the construction of finite state abstractions of such infinite systems which allows their formal verification and analysis.

The state space $\mathcal{X}$ from Eqn. (4) was partitioned by the set of thresholds into a number of hyper-rectangular regions. We partition $\mathcal{X}$ further using all linear inequalities $\pi \in \Pi$ from Eqn. (6) and ignore the measure-zero set consisting of all boundaries[6]. This results in a set of open polytopes $\mathcal{X}_l, l \in L$ such that, for all $l_1, l_2 \in L$, $\mathcal{X}_{l_1} \cap \mathcal{X}_{l_2} = \emptyset$ and $\cup_{l \in L} cl(\mathcal{X}_l) = \mathcal{X}$, where $cl()$ denotes the closure of a set.

All states from a given region satisfy the same atomic propositions and are defined as equivalent. This leads to the construction of a *finite quotient* T - a transition system with a set of states L where each state $l \in L$ represents the set of all equivalent states $x \in \mathcal{X}_l$. A transition between states l and l' is included in T only if a transition between a state from region $\mathcal{X}_l$ to a state from $\mathcal{X}_{l'}$ is possible in (5). The finite T simulates (in the sense of [24]) the infinite (5) identified through our procedure from Sec. IV-A (in other words, T can reproduce any word of (5)). This allows us to guarantee that if an arbitrary LTL formula ϕ is satisfied by T at state $l \in L$, then all trajectories of (5) originating in region $\mathcal{X}_l$ satisfy the formula. Note that when T does not satisfy ϕ from state l we cannot say anything about the satisfaction of ϕ from region $\mathcal{X}_l$, which makes the overall method conservative. In [20] we showed the uncertain parameter piecewise affine structure of (5) can be exploited and developed an algorithm for the construction of T through polyhedral operations.

In [21] we developed an analysis procedure based on the construction, model checking and refinement of simulation quotients such as T. Our algorithm used model checking to partition the set of states L into set $L^\phi \subseteq L$ from which T satisfied an LTL formula ϕ and $L^{\neg\phi} \subseteq L$ from which T satisfied the negation $\neg\phi$. This allowed us to guarantee that all trajectories originating in the *satisfying region* $\mathcal{X}^\phi = \bigcup_{l \in L^\phi} \mathcal{X}_l$ and none of the trajectories originating in the *violating region* $\mathcal{X}^{\neg\phi} = \bigcup_{l \in L^{\neg\phi}} \mathcal{X}_l$ satisfied ϕ. Both satisfying and violating

[5]The dynamics of (5) are not well defined at states that fall on thresholds but we ignore these measure zeros sets as it will become clear in Sec. IV-C.

[6]It is unreasonable to assume that equality constraints can be detected in practice and, in general, trajectories of the system do not disappear in such measure zero sets.

trajectories originated in region $(\cup_{l\in L}\mathcal{X}_l) \setminus (\mathcal{X}^{\phi} \cup \mathcal{X}^{\neg\phi})$ and our algorithm implemented an iterative refinement procedure to separate them, in which case $\mathcal{X}^{\phi}$ and $\mathcal{X}^{\neg\phi}$ can be expanded.

The relative volumes of the satisfying and violating regions can be used to asses the correctness of a GRN with respect to a temporal logic specification. A design is considered "good" if analysis reveals a large satisfying and an empty or small violating region, while a design is "bad" whenever a substantial violating region is found. This information can be used select GRNs proposed by the procedure from Sec. II-C.

V. Results

As a proof-of-concept example, we applied the design procedure proposed in this paper to the construction of the inverter GRN given in Fig. 1. First, we considered the simulated characterization data shown in Figs. 3-a and 3-b in order to test our procedure on an uncertain system but in the absence of measurement noise[7]. The data was used for the construction of an uncertain parameter piecewise affine model using the procedure from Sec. IV-A. The overall model is two dimensional ($N = 2$), where $G = \{RFP, TetR\}$, $P = \{pBad, pTet\}$ and $\bar{B}_{pBad}$ and $\bar{B}_{pTet}$ are the unions of all polytopes shown in gray in Figs. 3-a and 3-b, respectively.

Two LTL specifications over the concentrations of the output RFP are defined to capture the behavior of the inverter (see IV-B). First, we specify thresholds $T^L = 0.25$ and $T^H = 0.4$ in the normalized concentrations of RFP. The set of propositions in Eqn. (6) is defined as $\Pi = \{\pi_1, \pi_2\}$, where atomic propositions $\pi_1 := x_{RFP} < T^L$ and $\pi_2 := x_{RFP} > T^H$ are satisfied when the RFP concentrations (output) are low and high, respectively. For high concentrations of the input (Ara), we require this output to eventually reach, and for all future times, remain at low concentrations (*i.e.* satisfy π_1), which we express as the LTL formula $\phi_1 = \Diamond\Box\pi_1$. Similarly, we express specification $\phi_2 = \Diamond\Box\pi_2$ which must be satisfied for low concentrations of Ara.

To test whether the inverter behaves according to these specifications, we analyze the identified model with ϕ_1 and ϕ_2 at different concentrations of Ara. We first partition the range of possible input concentrations into a number of regions. For each input concentration region, we treat promoter $pBad$ as constitutive as discussed in Remark 1 (see Fig. 3-a). The relative volumes of the satisfying and violating regions (as a fraction of the overall state space) for ϕ_1 and ϕ_2 are computed using the procedure from Sec. IV-C and are shown in Fig. 3-d.

Analysis reveals that specification ϕ_1 is satisfied by all trajectories of the system (*i.e.* from all initial conditions) for Ara concentrations above 0.85 (Fig. 3-d top-left) but none of the trajectories satisfy ϕ_1 for Ara concentrations below 0.2 (Fig. 3-d bottom-left). Similarly, ϕ_2 is satisfied or violated by all the system's trajectories for $Ara < 0.15$ (Fig. 3-d top-right) or $Ara > 0.75$ (Fig. 3-d bottom-right), respectively. For intermediate concentrations $Ara \in [0.7; 0.8]$ the trajectories originating in some but not all initial conditions can be guaranteed to satisfy ϕ_1 (Fig. 3-d top-left) or violate ϕ_2 (Fig. 3-d bottom-right), while nothing can be guaranteed for the system at concentrations $Ara \in [0.4; 0.6]$. To guarantee that the device functions robustly as an inverter, we select threshold $I^H = 0.85$ as the smallest Ara concentration guaranteeing the satisfaction of ϕ_1 from all initial conditions and, similarly, we select $I^L = 0.2$ as the largest Ara concentration guaranteeing the satisfaction of ϕ_2 (Ara concentrations above I^H and below I^L are considered "high" and "low", respectively). Several simulated trajectories of the system shown in Fig. 3-c demonstrate the correct behavior of the inverter when appropriate input signals are applied.

Next, we applied our procedure as described above but considered experimental data collected as discussed in Sec. III. Starting from this characterization data, we compute the rates of expression from the $pBad$ and $pTet$ promoters at different concentrations of the inducer Arabinose (Ara) and the transcription factor $TetR$, respectively, as described in Sec. IV-A. The computed expression rate observations and allowed ranges for promoters $pBad$ and $pTet$ are shown in Figs. 3-e and 3-f, respectively. Due to sources of uncertainty not accounted for in simulations and additional measurement errors, the allowed expression rates ranges are much wider. In particular, minimal expression from $pBad$ and $pTet$ is allowed regardless of the concentration of their respective inducer and repressor. Simulated trajectories reveal that the circuit behaves as an inverter for some parameters from the allowed ranges (solid lines in Fig. 3-g). However, the circuit could not be verified using our procedure (*i.e.* neither a satisfying nor a violating region were identified), signifying that both satisfying and violating behavior might be possible. This is not surprising since, regardless of the concentration of Ara, low expression rates are allowed from both $pBad$ and $pTet$, in which case the circuit no longer behaves as an inverter (dashed lines in Fig. 3-g). For experimental data resulting in such wide uncertainty ranges, considering all observed parameters might be too conservative but other characterization techniques can be applied to decrease measurement error and obtain tighter uncertainty ranges. Alternatively, a probabilistic framework can be developed where the probability of different expression rates is also considered, in order to decrease the conservatism of the method. Both directions are currently being explored.

VI. Conclusion

Synthetic biology as a discipline is committed to applying engineering principles. If we are to fulfill this commitment, forward engineering of designs from primitive components is required. We have presented a design flow which begins with the conversion of digital logic specifications into genetic circuits. The DNA components used in these circuits are then individually characterized. Given these circuits and their primitive characterizations we have introduced a verification mechanism using LTL which can reason about the potential performance of the composite circuit. While still in its infancy, this approach will be vital to prevent time consuming and costly creation of genetic circuits and allow synthetic biologists to focus their efforts on creating interesting designs.

[7]Although we keep the names of all species unchanged, the simulated data is for demonstration only and does not characterize the actual promoters.

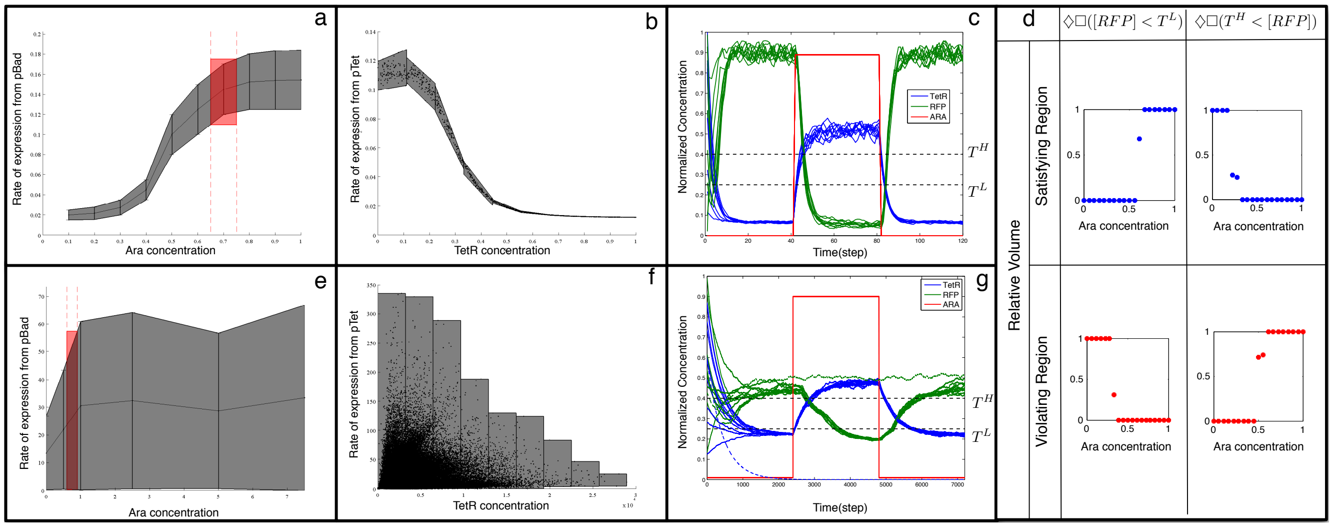

Fig. 3. (a) Simulated characterization data for the $pBad$ promoter. A whole family of possible regulation functions with the data is captured by the uncertain parameter piecewise affine model (Sec. IV-A), where possible expression rate ranges fall within the shaded polytopes. Once the concentration of the inducer Ara is fixed to the range between the dashed red lines, $pBad$ can be considered constitutive with range of possible expression rates between the minimal and maximal rate allowed in the polytope shaded in red (Remark 1). (b) Simulated characterization data for the $pTet$ promoter. In this case, the concentrations of the regulator $TetR$ are not controlled externally and in general do not fall on the thresholds. (c) Simulated trajectories of the GRN from Fig. 1 demonstrates its function as an inverter. (d) The relative volumes of the satisfying and violating regions for specifications $\phi_1 = \Diamond\Box(x_{RFP} < T^L)$ and $\phi_2 = \Diamond\Box(T^H < x_{RFP})$ are computed using the analysis procedure from Sec. IV-C. (e) Characterization of the $pBad$ promoter. The range of expression rates at different inducer concentrations is computed from experimental data (Sec. III). (f) Characterization of the $pTet$ promoter. (g) Simulated trajectories confirm that under some expression rates from the allowed ranges the circuit functions as an inducer (solid lines) but it fails for others (dashed lines).

Of equal importance we have identified an interdisciplinary research opportunity for electronic design automation to play a role in synthetic biology.

Acknowledgments

The authors thank the Voigt lab at MIT for discussions about the genetic NOR gate design and the Weiss lab at MIT and Jacob Beal at BBN Technologies for discussions about part characterization. This material is based upon work supported by the Office of Naval Research under Grant No. N00014-11-1-0725.

References

[1] E. Andrianantoandro, S. Basu, D. K. Karig, and R. Weiss, "Synthetic biology: new engineering rules for an emerging discipline," *Mol. Sys. Biol.*, vol. 2, 2006.
[2] D.-K. Ro, E. M. Paradise, M. Ouellet, K. J. Fisher, K. L. Newman, J. M. Ndungu, K. A. Ho, R. A. Eachus, T. S. Ham, J. Kirby, and et al., "Production of the antimalarial drug precursor artemisinic acid in engineered yeast." *Nature*, vol. 440, no. 7086, pp. 940–943, 2006.
[3] J. D. Keasling and H. Chou, "Metabolic engineering delivers next-generation biofuels," *Nature Biotech.*, vol. 26, no. 3, pp. 298–299, 2008.
[4] Z. Xie, L. Wroblewska, L. Prochazka, R. Weiss, and Y. Benenson, "Multi-input rnai-based logic circuit for identification of specific cancer cells," *Science*, vol. 333, no. 6047, pp. 1307–1311, 2011.
[5] M. J. Dunlop, J. D. Keasling, and A. Mukhopadhyay, "A model for improving microbial biofuel production using a synthetic feedback loop," *Systems and synthetic biology*, vol. 4, no. 2, pp. 95–104, 2010.
[6] J. Monod and F. Jacob, "Teleonomic mechanisms in cellular metabolism, growth, and differentiation," in *Cold Spring Harbor Symposia on Quantitative Biology*, vol. 26, 1961, pp. 389–401.
[7] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in escherichia coli." *Nature*, vol. 403, no. 6767, 2000.
[8] E. M. Judd, M. T. Laub, and H. H. McAdams, "Toggles and oscillators: new genetic circuit designs," *BioEssays*, vol. 22, no. 6, 2000.
[9] J. Hasty, M. Dolnik, V. Rottschäfer, and J. J. Collins, "Synthetic gene network for entraining and amplifying cellular oscillations," *Phys. Rev. Lett.*, vol. 88, p. 148101, Mar 2002.
[10] A. E. Friedland, T. K. Lu, X. Wang, D. Shi, G. Church, and J. J. Collins, "Synthetic gene networks that count," *Science*, vol. 324, no. 5931, 2009.
[11] A. Tamsir, J. J. Tabor, and C. A. Voigt, "Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'," *Nature*, vol. 469, pp. 212–215, Jan. 2011.
[12] C. J. Myers, *Engineering Genetic Circuits*. Chapman and Hall, 2009.
[13] J. C. Anderson, C. A. Voigt, and A. P. Arkin, "Environmental signal integration by a modular and gate," *Molecular Systems Biology*, vol. 3, no. 133, p. 133, 2007.
[14] P. E. M. Purnick and R. Weiss, "The second wave of synthetic biology: from modules to systems," *Nat Rev Mol Cell Biol*, 2009.
[15] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular biology of the cell*, 4th ed. Garland Science, 2002.
[16] R. M. Hill, "Evaluation of susceptibility functions," *physica status solidi (b)*, vol. 103, no. 1, pp. 319–328, 1981.
[17] R. Ghamari, B. Stanton, T. Haddock, S. Bhatia, K. Clancy, T. Peterson, C. Voigt, and D. Densmore, "Applying hardware description languages to genetic circuit design," in *Proc. of IWBDA*, 2011.
[18] K. Keutzer, "Dagon: technology binding and local optimization by dag matching," in *Proc. of the 24th ACM/IEEE Design Automation Conference*, ser. DAC '87. New York, NY, USA: ACM, 1987.
[19] E. M. Clarke, *Model Checking*. MIT Press, 1999.
[20] B. Yordanov and C. Belta, "A Formal Verification Approch to the Design of Synthetic Gene Networks," in *Proceedings of the 50th IEEE Conference on Decision and Control (CDC)*, Orlando,FL, Dec. 2011.
[21] ——, "Formal Analysis of Discrete-Time Piecewise Affine Systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 12, Dec. 2010.
[22] S. Drulhe, G. Ferrari-Trecate, and H. de Jong, "The Switching Threshold Reconstruction Problem for Piecewise-Affine Models of Genetic Regulatory Networks," *IEEE Transactions on Automatic Control*, vol. 53, no. Special Issue, pp. 153–165, 2008.
[23] G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider, "Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in escherichia coli," *Bioinformatics*, vol. 21, no. suppl 1, 2005.
[24] R. Milner, *Communication and concurrency*. Prentice-Hall, 1989.