BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 18 2007, pages 2415–2422
doi:10.1093/bioinformatics/btm362

Systems Biology

Robustness analysis and tuning of synthetic gene networks

Grégory Batt[1,*], Boyan Yordanov[2], Ron Weiss[3] and Calin Belta[1]

[1]Centers for Information and Systems Engineering and for BioDynamics, [2]Department of Biomedical Engineering, Boston University, Boston, MA and [3]Departments of Electrical Engineering and Molecular Biology, Princeton University, Princeton, NJ, USA

Received on March 17, 2007; revised on June 8, 2007; accepted on July 8, 2007

Advance Access publication July 27, 2007

Associate Editor: Chris Stoeckert

ABSTRACT

Motivation: The goal of synthetic biology is to design and construct biological systems that present a desired behavior. The construction of synthetic gene networks implementing simple functions has demonstrated the feasibility of this approach. However, the design of these networks is difficult, notably because existing techniques and tools are not adapted to deal with uncertainties on molecular concentrations and parameter values.

Results: We propose an approach for the analysis of a class of uncertain piecewise-multiaffine differential equation models. This modeling framework is well adapted to the experimental data currently available. Moreover, these models present interesting mathematical properties that allow the development of efficient algorithms for solving robustness analyses and tuning problems. These algorithms are implemented in the tool RoVerGeNe, and their practical applicability and biological relevance are demonstrated on the analysis of the tuning of a synthetic transcriptional cascade built in *Escherichia coli*.

Availability: RoVerGeNe and the transcriptional cascade model are available at http://iasi.bu.edu/%7Ebatt/rovergene/rovergene.htm

Contact: gregory.batt@imag.fr

1 INTRODUCTION

The main goal of the nascent field of synthetic biology is to design and construct biological systems that present a desired behavior (Andrianantoandro *et al*., 2006; Endy, 2005). Synthetic biology is foreseen to have important applications in biotechnology and medicine, and to contribute significantly to a better understanding of the functioning of complex biological systems (McDaniel and Weiss, 2005). The construction of networks of interregulating genes, so-called genetic regulatory networks, has demonstrated the feasibility of this approach (e.g. Gardner *et al*., 2000). Still, the development of gene networks is difficult: most newly created networks are non-functioning and need tuning. One important reason is that the lack of precise knowledge on molecular concentrations and on parameter values hampers the design of synthetic networks. These uncertainties are the consequence of current technological limitations and also of the fluctuations of intra- and extracellular environments.

Existing solutions for the analysis of dynamical properties of gene networks consist essentially either in qualitative simulation of coarse-grained models or in extensive numerical simulations of nonlinear differential equation models or stochastic versions thereof (de Jong, 2002; Szallasi *et al*., 2006). For applications in synthetic biology, these approaches are not satisfying. For qualitative models, the predictions obtained are generally too coarse for answering the—often quantitative—questions of interest. For uncertain quantitative models, a common approach is to perform many numerical simulations so as to 'sample' the state and parameter spaces, often in conjunction with local sensitivity analyses. This approach provides only a partial description of all the possible behaviors of a network. In particular, it cannot provide the guaranty that a network behaves as expected for all initial conditions and parameters in given ranges. Moreover, obtaining a 'reasonably dense' coverage of the state and parameter spaces quickly becomes computationally intractable when the size of the networks grows.

In this work, we demonstrate the biological relevance of a method specifically developed to support the *design of synthetic gene networks*. This method allows to analyze dynamical properties of uncertain, yet quantitative models of gene networks. More precisely, we consider piecewise-multiaffine differential equation models in which uncertain initial conditions and parameters are given by intervals. These models capture essential aspects of genetic regulations and still allow for efficient analyses by tailored formal verification techniques. Dynamical properties of the network are given by temporal logic formulas that specify temporal constraints on the state of the system, that is, on protein concentrations. Temporal logics are specification languages that allow to express a variety of properties on the behavior of dynamical systems (Emerson, 1990). Then, the proposed approach allows to *check* automatically that a network satisfies a given dynamical property for all initial conditions and all parameter values in the given intervals. This provides us a means to *assess the robustness* of the expected behavior of a network with respect to parameter variations. In particular, our technique does *not* rely on numerical simulations. Additionally, the proposed approach has the capability to generate constraints on parameters, and can consequently be extended to *search* for parameter sets for which a given property is satisfied. This feature allows to solve *network tuning problems* by suggesting modifications of biological parameters. These techniques are implemented in a publicly available tool called RoVerGeNe (for Robust Verification of Gene Networks) and their applicability and biological relevance is demonstrated on the analysis of the tuning of a synthetic transcriptional cascade.

*To whom correspondence should be addressed.

© The Author 2007. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oxfordjournals.org

Downloaded from https://academic.oup.com/bioinformatics/article/23/18/2415/239215 by guest on 29 December 2023

The remainder of this article is organized as follows. In the next section, we provide a brief description of the proposed method and of its implementation in the computer tool RoVerGeNe. In Section 3, we detail the application of our method to the tuning of a synthetic transcriptional cascade built in *Escherichia coli*. The results are summarized in the last section. We refer the reader to (Batt *et al*., 2007a, b) for a detailed presentation of the method and for computational results using a preliminary version of the transcriptional cascade model.

2 ANALYSIS OF PIECEWISE-MULTIAFFINE MODELS WITH PARAMETER UNCERTAINTY

In this section, we provide an intuitive overview of the proposed approach by means of a simple example.

2.1 Piecewise-multiaffine models and LTL properties

Consider the cross-inhibition network represented in Figure 1. The network is made of two genes, a and b, that code for two repressor proteins, A and B. More specifically, protein B represses the expression of gene a, whereas protein A represses the expression of gene b, and at a higher concentration, the expression of its own gene. Protein degradations are not regulated.

This system can be modeled by differential equations as follows.

$$\dot{x}_a = \kappa_a\, r_{a1}(x_b)\, r_{a2}(x_a) - \gamma_a\, x_a, \tag{1}$$

$$\dot{x}_b = \kappa_b\, r_b(x_a) - \gamma_b\, x_b, \tag{2}$$

with $x = (x_a, x_b) \in \mathcal{X} = [0, \max_a] \times [0, \max_b]$. The state variables x_a and x_b denote the concentrations of protein A and B. x is the vector of state variables and $\mathcal{X}$ is the state space. $\max_a$ and $\max_b$ denote a maximal concentration for proteins A and B. κ's and γ's are respectively *production* and *degradation rate parameters*, and r's are *regulation functions*. The latter capture the regulatory effect of an effector protein on gene expression. In contrast to most nonlinear models in which the regulation functions are smooth sigmoidal functions (e.g. Hill functions) (de Jong, 2002), we assume that regulation functions are *piecewise-affine* (Fig. 2). These functions are uniquely defined by their values at *breakpoints*, denoted by λ's. For our example model, we used the simplest piecewise-affine functions approximating sigmoidal curves: ramp functions. These functions have only four break points (including 0 and $\max_i$). The ordered set of all breakpoints associated with the variable x_i is denoted by Λ_i. For example, we have $\Lambda_a = \{0, \lambda_{a1}, \lambda_{a2}, \lambda_{a3}, \lambda_{a4}, \max_a\}$ and $\Lambda_b = \{0, \lambda_{b1}, \lambda_{b2}, \max_b\}$.

Products of regulation functions (involving different state variables) can be used to capture complex genetic regulations. In Equation (1), for example, the product of regulation functions captures the hypothesis that in order to have a maximal expression of gene a both proteins must be present in low concentration (i.e. below λ_{a3} and λ_{b1}). Because products of piecewise-affine functions are allowed, the resulting models are in general *piecewise-multiaffine*. We recall that a multiaffine function is a polynomial with the property that the degree in any of its variable is at most 1 (Belta and Habets, 2006). In particular, products of different variables are allowed. A motivation for considering piecewise-affine regulation functions is that

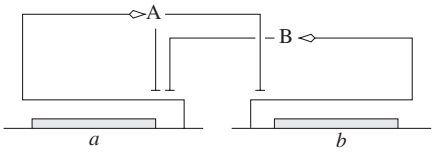


Fig. 1. Cross-inhibition network.

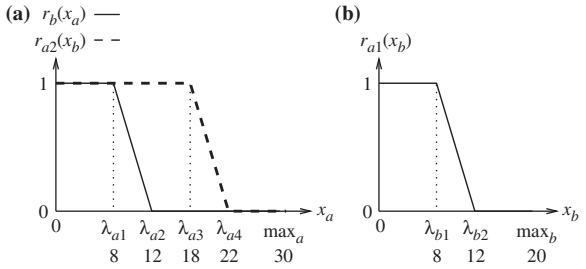


Fig. 2. Regulation functions in Equations (1) and (2) for the cross-inhibition network.

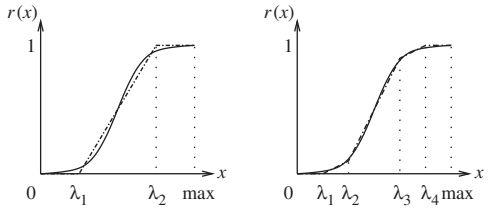


Fig. 3. Approximation of sigmoidal functions by piecewise-affine functions. Better approximations can be obtained by using more breakpoints.

piecewise-affine functions have universal approximation properties (Lin and Unbehauen, 1992), which means that any nonlinear function can be approximated by a piecewise-affine function with arbitrary accuracy (Fig. 3). Moreover, while the analysis of general nonlinear systems (e.g. Hill-type models) is notoriously difficult, efficient approaches have been recently developed for multiaffine systems (Belta and Habets, 2006).

Some parameters might be uncertain. Their values are then given by intervals. We assume that production and/or degradation rate parameters can be uncertain (i.e. κ's and γ's), but that regulation functions are precisely known. We denote by p the vector of uncertain parameters and by $\mathcal{P}$ the parameter space. For the cross-inhibition network, we assume that parameters satisfy $\kappa_a \in [0, 30]$, $\kappa_b \in [0, 40]$, $\gamma_a = 1$, and $\gamma_b = 2$. So, we have $p = (\kappa_a, \kappa_b) \in \mathcal{P} = [0, 30] \times [0, 40]$. More generally, the models that we consider are *piecewise-multiaffine (PMA) systems* Σ of the general form

$$\dot{x} = f(x, p), \quad x \in \mathcal{X}, p \in \mathcal{P} \tag{3}$$

where f is a piecewise-multiaffine function of the state variables x and an affine function of the uncertain parameters p.

A number of different formalisms has been proposed to describe gene networks (de Jong, 2002). The use of piecewise-multiaffine models for gene networks was first proposed by Belta *et al.* (2002) (see Mestl *et al.*, 1995 for a related, piecewise-continuous approach). The class of piecewise-multiaffine models that we consider is also related to the class

Downloaded from https://academic.oup.com/bioinformatics/article/23/18/2415/239215 by guest on 29 December 2023

of piecewise-affine (PA) differential equation models proposed by Glass and Kauffman (1973). Even if multiaffine models do not present the monotonicity properties that make the qualitative, symbolic analysis of PA models attractive (de Jong *et al*., 2004; Ghosh and Tomlin, 2004; see also Kauffman, 1969; Thomas *et al*., 1995, for alternative, discrete formalisms), the use of piecewise-affine functions to represent genetic regulations (instead of step functions for PA models) allows to develop finer-grained models, better adapted to quantitative analyses. In particular, PMA models capture the graded response of gene expression to continuous changes in effector (activator or repressor) concentrations.

We use Linear Temporal Logic (LTL) to express dynamical properties of gene networks. Temporal logics have been developed to specify the behavior of (usually discrete) dynamical systems (Emerson, 1990). Typical properties include reachability (the system can reach a given state), inevitability (the system will necessarily reach a given state), invariance (a property is always true), response (an event necessarily triggers a specific behavior) and infinite occurrences (such as oscillations). Illustrative examples of the expressiveness of temporal logics in systems biology can be found in Antoniotti *et al*. (2003), Batt *et al*. (2005), Bernot *et al*. (2004) and Fages *et al*. (2004). LTL formulas are built using atomic propositions *and* LTL operators. In our approach, atomic propositions express simple constraints on protein concentrations and are of type '$x_i < \lambda$' or '$x_i > \lambda$' .[1] LTL operators include the usual logical operators, such as *negation* ($\neg$), logical *and* ($\wedge$), logical *or* ($\vee$), and *implication* ($\rightarrow$), and specific temporal operators, such as *future* ($\mathbf{F}$), *globally* ($\mathbf{G}$) and *until* ($\mathbf{U}$). $\mathbf{F}\,p$, $\mathbf{G}\,p$ and $p\,\mathbf{U}\,q$ respectively mean that a property p holds at some future time, holds for all future times, or holds continuously until an other property q holds. These operators can be combined to express complex dynamical properties.

The cross-inhibition network is known to be *bistable*. If the system is in a state in which the concentration of protein A is low and the concentration of protein B is high, then it will remain in such a state for all time. A symmetrical property holds with the concentrations of A and B being high and low, respectively. This property can be expressed in LTL by the formula ϕ_1, where, for example, the first part of the property expresses that if the concentrations of protein A and B are respectively low ($x_a < \lambda_{a1}$) and high ($x_b > \lambda_{b2}$), then the system will always ($\mathbf{G}$) remain in such a state.

$$\begin{aligned}\phi_1 = (x_a < \lambda_{a1} \wedge x_b > \lambda_{b2} \rightarrow \mathbf{G}\,(x_a < \lambda_{a1} \wedge x_b > \lambda_{b2}))\\ \wedge (x_b < \lambda_{b1} \wedge x_a > \lambda_{a3} \rightarrow \mathbf{G}\,(x_b < \lambda_{b1} \wedge x_a > \lambda_{a3}))\end{aligned} \quad (4)$$

The semantics of LTL formulas is defined over executions of transition systems (Emerson, 1990). Transition systems consist of a (finite or infinite) set of states and of a set of transitions between states. Transition systems define a set of executions, which are sequences of states for which there exists a transition from each state to its successor. So, in order to define what it means that a PMA system Σ satisfies an LTL property ϕ for a given parameter $p \in \mathcal{P}$, we introduce an embedding transition system, denoted by $T_{\mathcal{X}}(p)$, in which the states are the points x in $\mathcal{X}$, and the transitions between two points correspond to the existence of a solution of the differential equation (3) going from one point to the other. Consequently, executions of $T_{\mathcal{X}}(p)$ correspond to solution trajectories of (3). Then, a PMA system Σ satisfies an LTL property ϕ for a given parameter p if every execution of the associated embedding transition system $T_{\mathcal{X}}(p)$ satisfies the property ϕ, denoted by $T_{\mathcal{X}}(p) \models \phi$. We say that the parameter p is *valid* for ϕ. Finally, a parameter *set* P is valid for ϕ if every parameter in P is valid for ϕ. In this work, we consider the following two problems.

Problem *Let Σ be a PMA system, $\mathcal{P}$ an hyperrectangular parameter space, and ϕ an LTL formula.*
Problem 1. Robustness analysis: *Check whether $\mathcal{P}$ is valid for ϕ.*
Problem 2. Tuning: *Find a set $P \subseteq \mathcal{P}$ such that P is valid for ϕ.*

The state space associated with our two-gene example is shown in Figure 4a. The flow and a solution trajectory passing through three points, x^1, x^2 and x^3, are also represented for a given parameter $\hat{p} = (26, 34)$. In $T_{\mathcal{X}}(\hat{p})$, there is for example a transition from x^1 to x^2, and from x^2 to x^3. The solution trajectory represented in Figure 4a can be associated with the execution $(x^1, x^2, x^3, \ldots)$.

2.2 Analysis of uncertain PMA systems

Problems 1 and 2 amount to prove that a given property is satisfied for sets of initial conditions and for sets of parameters. Consequently, these problems cannot be solved by numerical integration, since it would require to check whether the property holds for an infinite number of solution trajectories. Instead, we use a combination of techniques developed for the verification of continuous, and more generally hybrid (i.e. continuous and discrete) dynamical systems. The principle of the analysis is simple. *Discrete abstractions* (Alur *et al*., 2000) are used to transpose problems defined on (infinite) continuous state and parameter spaces into problems defined on (finite) discrete spaces. Algorithmic analysis by *model checking* (Clarke *et al*., 1999) is then possible.

The first step of our analysis is to define a partition of the state space. Given the piecewise nature of the differential equation system (3), it is natural to partition the state space into regions in which the differential equations have a same expression. So we consider the hyperrectangular partition defined by the breakpoints in Λ_i for every variable x_i (Fig. 4a). Full-dimensional regions of the partition are called *rectangles* $R \in \mathcal{R}$. For our example network, $\mathcal{R}$ contains 15 rectangles: $R^1, \ldots, R^{15}$ (Fig. 4a).

For every parameter set P, we define the discrete abstraction of a PMA system Σ as the discrete transition system $T_{\mathcal{R}}(P)$ in which the states are the rectangles, and the transitions between (adjacent) rectangles correspond to the existence of solution trajectories of (3) for some parameter in P, going from one rectangle to the other. In Batt *et al*. (2007a), we have shown that the discrete abstraction $T_{\mathcal{R}}(P)$ captures every possible behavior of the original system Σ for every parameter $p \in P$. More precisely, $T_{\mathcal{R}}(P)$ is a *conservative approximation* of Σ, in the sense that to every solution trajectory of (3), there exists a corresponding execution in $T_{\mathcal{R}}(P)$.[2] Note however, that $T_{\mathcal{R}}(P)$

[1]Note that the assumption $\lambda \in \Lambda_i$ is made without loss of generality.

[2]In fact, this property holds only for *almost all* solution trajectories of (3). For our biological applications, this technical restriction is of no practical importance and is disregarded in the sequel (see Batt *et al*., 2007a).

Downloaded from https://academic.oup.com/bioinformatics/article/23/18/2415/239215 by guest on 29 December 2023

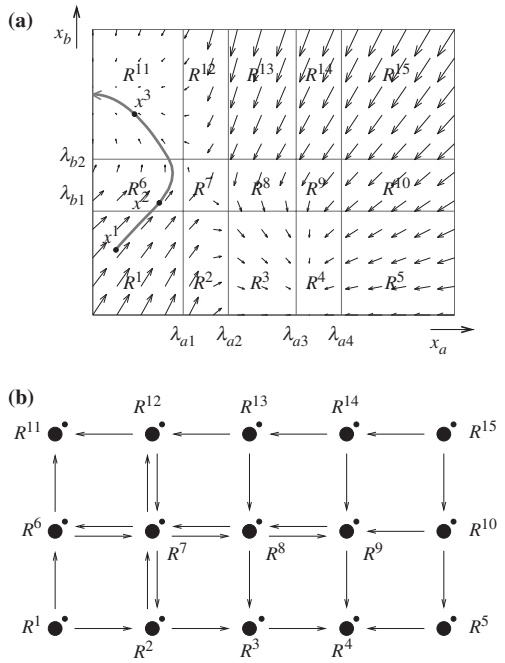


Fig. 4. (**a**) Continuous dynamics in the state space of the cross-inhibition network for a given parameter, $\hat{p} = (\kappa_a, \kappa_b) = (26, 34)$. (**b**) Discrete abstraction $T_{\mathcal{R}}(\hat{P})$ for the parameter set $\hat{P} = [20, 30] \times [30, 40]$. Dots denote self transitions.

may contain spurious executions, that is, executions corresponding to no solution trajectory of the original system. For our example network, the discrete abstraction of the system associated with the parameter set $\hat{P}=[20,30]\times[30,40]\subseteq\mathcal{P}$ is represented in Figure 4b. For parameter $\hat{p}\in\hat{P}$, there exists a solution reaching R^6 from R^1, and R^{11} from R^6 (Fig. 4a), so there exists transitions from R^1 to R^6 and from R^6 to R^{11} in $T_{\mathcal{R}}(\hat{P})$ (Fig. 4b). The solution trajectory represented in Figure 4a corresponds to the execution $(R^1, R^6, R^{11}, R^{11}, \ldots)$ in $T_{\mathcal{R}}(\hat{P})$.

Contrary to the original, infinite system, the abstract system being finite can be analyzed by model checking techniques. Model checking techniques are highly efficient automatic techniques developed for the analysis of finite transition systems. In particular, off-the-shelf tools exist to check whether discrete transition systems satisfy given temporal logic properties. Using these tools, we can test whether $T_{\mathcal{R}}(P) \models \phi$, and if this holds, we can conclude that the original system Σ satisfies the property ϕ for every parameter in P using the fact that conservative approximations weakly preserve LTL (Browne *et al.*, 1988): if a property is true for the abstract system, then it holds for the original system. Note, however, that due to the possible existence of spurious executions in the abstract system, the converse is not necessarily true. Stated differently, we might fail to prove some properties.

It is easy to check on the discrete transition system $T_{\mathcal{R}}(\hat{P})$ represented in Figure 4b that the network satisfies the bistability property ϕ_1 for every parameter value in $\hat{P}$. If the state $x = (x_a, x_b)$ of the system satisfies $x_a < \lambda_{a1}$ and $x_b > \lambda_{b2}$, then $x \in R^{11}$ (Fig. 4a), and because there is no transition leaving R^{11} in $T_{\mathcal{R}}(\hat{P})$ (Fig. 4b), the system can not leave R^{11}. By a similar reasoning, one can check that the second half of property ϕ_1 also holds (the system always remains in R^4 or R^5, where protein A and B concentrations are respectively high and low).

We have still not provided a means to actually *compute* the discrete abstraction $T_{\mathcal{R}}(P)$. In fact, we need to be able to decide whether solutions starting from a rectangle can enter an adjacent rectangle. For general, uncertain nonlinear dynamical system, there is no known method to solve this problem. Fortunately, we can exploit two specific properties of the class of models that we consider. First, because the models are piecewise-multiaffine functions of the state variables, the existence of transitions between two adjacent rectangles only depends on the direction of the vector field at the vertices of the facet that separates the two rectangles. This comes from convexity properties of multiaffine functions in hyperrectangular regions (Belta and Habets, 2006). Second, because the models are affine functions of the uncertain parameters, the vector field at a vertex v depends affinely on the unknown parameters. So we can show that the set of parameters for which there exists a transition between two rectangles corresponds to a union of polyhedral sets in the parameter space (Batt *et al.*, 2007a). As a consequence, the discrete transition system $T_{\mathcal{R}}(P)$ can be computed by means of polyhedral operations for a hyperrectangular, or more generally, for a polyhedral parameter set P.

For the cross-inhibition network, there exists a transition from R^1 to R^2 if and only if the vector field at one of the vertices of the separating facet (of coordinates $(\lambda_{a1}, 0)'$ or $(\lambda_{a1}, \lambda_{b1})'$) points 'to the right' (i.e. is such that $\dot{x}_a > 0$). It holds for both vertices that $\dot{x}_a = \kappa_a - \gamma_a \lambda_{a1}$, which is positive for every $\kappa_a \in [20, 30]$ ($\gamma_a = 1$ and $\lambda_{a1} = 8$). So there is a transition from R^1 to R^2 in $T_{\mathcal{R}}(\hat{P})$. Conversely, one can show that there is no transition from R^2 to R^1 in $T_{\mathcal{R}}(\hat{P})$.

We can now solve robustness problems (Problem 1) by the following two-step procedure: first, compute the discrete abstraction $T_{\mathcal{R}}(P)$ by means of polyhedral operations, and second, test on the discrete abstraction whether the property ϕ is true by model checking. If it is true, then we can conclude that the property is true for all parameters in the parameter set, or stated differently, that the parameter set P is valid for ϕ. Note, however, that if the discrete abstraction does not satisfy the property, no conclusion can be drawn on the original system.

In order to deal with tuning problems (Problem 2), we use the observation made previously that the existence of transitions in the discrete abstractions depends on a set of affine constraints on parameters. All these constraints define a polyhedral partition $\mathcal{P}$ of the parameter space, represented in Figure 5 for our example network. All parameters in a same region $P \in \mathcal{P}$ are equivalent, in the sense that they are associated with a same discrete abstraction.

Then, a naive approach to find solutions to Problem 2 is to test the validity of every parameter equivalence class $P \in \mathcal{P}$ of the parameter space using the previous approach (i.e. for every $P \in \mathcal{P}$, compute $T_{\mathcal{R}}(P)$ and test whether $T_{\mathcal{R}}(P) \models \phi$). Every parameter set identified this way provides solutions to the tuning problem, since it suggests a way to modify network parameters such that the tuned system is guaranteed to satisfy the expected property. Conversely, not all valid parameters are guaranteed to be found by our approach. For our example network, if we test the validity of the regions $P^1, P^2, \ldots, P^{12}$

Downloaded from https://academic.oup.com/bioinformatics/article/23/18/2415/239215 by guest on 29 December 2023

represented in Figure 5, we find that P^{12} is a valid parameter set. So we can conclude that the gene network is bistable if $\kappa_a > 18$ and $\kappa_b > 24$. In fact, we have developed a more efficient approach that allows to reason with fewer, larger regions in parameter space, corresponding to unions of parameter equivalence classes $P \in \mathcal{P}$ (Batt *et al.*, 2007a). Still, computational times generally increase exponentially with the number of genes and uncertain parameters. Consequently, the applicability of our method is currently limited to the analysis of networks of moderate size (i.e. having less than a half-dozen genes) as currently encountered in most synthetic gene networks. The analysis of future, larger synthetic networks will require to extend our approach to exploit their modularity (Chin, 2006).

This method has been implemented in a freely available tool for Robust Verification of Gene Networks (RoVerGeNe), written in Matlab on top of several other tools (MPT, MatlabBGL, NuSMV). Additionally, RoVerGeNe supports an extension of the method presented here, dealing with problems specifically encountered when verifying liveness properties (Batt *et al.*, 2007b).

3 TUNING OF A SYNTHETIC TRANSCRIPTIONAL CASCADE

In this section, we illustrate the practical applicability and biological relevance of the approach presented in the previous section for the analysis of synthetic gene networks.

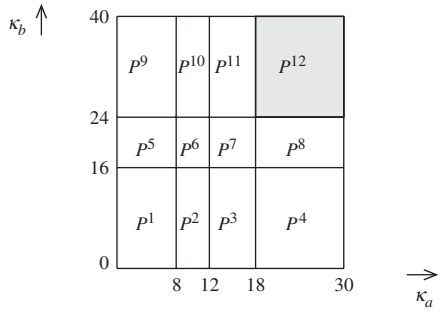


Fig. 5. Partition of the parameter space $\mathcal{P}$ of the cross-inhibition network. Valid regions are shaded.

3.1 Problem

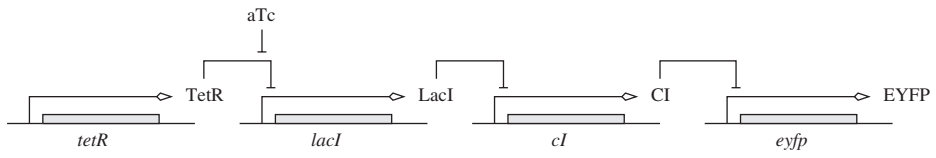
We consider a cascade of transcriptional inhibitions built in *E.coli* by Hooshangi *et al.* (2005). The network is represented in Figure 6a. It is made of four genes: *tetR*, *lacI*, *cI*, and *eyfp* that code respectively for three repressor proteins, TetR, LacI and CI, and the fluorescent protein EYFP. The fluorescence of the system, due to the protein EYFP, is the measured output. The system can be controlled by the addition or removal of a small diffusible molecule, aTc, in the growth media. More precisely, aTc binds to TetR and relieves the repression of *lacI*. The aTc concentration thus serves as a controllable input to the system. It is intuitively clear that the output (i.e. the fluorescence) of the system at steady state will be low for low inputs (i.e. aTc concentration), and high for high inputs. Moreover, because of the topology of the network (cascade of inhibitions), an *ultrasensitive* response may be achieved: the output at steady state undergoes a dramatic change for a moderate change of the input in a transition region. More precisely, we would like that the system at steady state satisfies the input/output specifications represented in Figure 7a, in which the output of the system is expected to remain between the two dotted lines. In particular, this specifies that a 1000-fold increase of the output is obtained for a 4-fold increase of the input.

Unfortunately, the actual network does not meet these specifications (Fig. 7a). So we used our method and tool to investigate how to tune it. In a preliminary step, we have developed a PMA model of the network. Then, using RoVerGeNe, we have investigated the possibility to tune the network by modifying some of its parameters (Problem 2), proposed parameter modifications, and evaluated computationally the robustness of the modified system (Problem 1). Note that it is important to perform this last step before experimentally tuning the network in order to gain confidence that the tuned system will behave as expected despite errors in parameter identification, incorrect parameter modifications or environmental fluctuations.

3.2 Modeling and specification

We have developed a piecewise-multiaffine model of the cascade, represented in Figure 6b. The notations are similar

(a)



(b)

$$\dot{x}_{tetR} = \kappa_{tetR} - \gamma_{tetR}\, x_{tetR}, \tag{5}$$

$$\dot{x}_{lacI} = \kappa_{lacI,0} + \kappa_{lacI}\,(r_{lacI,1}(x_{tetR}) + r_{lacI,2}(u_{aTc}) - r_{lacI,1}(x_{tetR})\, r_{lacI,2}(u_{aTc})) - \gamma_{lacI}\, x_{lacI}, \tag{6}$$

$$\dot{x}_{cI} = \kappa_{cI,0} + \kappa_{cI}\, r_{cI}\,(x_{lacI}) - \gamma_{cI}\, x_{cI}, \tag{7}$$

$$\dot{x}_{eyfp} = \kappa_{eyfp,0} + \kappa_{eyfp}\, r_{eyfp}(x_{cI}) - \gamma_{eyfp}\, x_{eyfp}, \tag{8}$$

$$\dot{u}_{aTc} = 0. \tag{9}$$

Fig. 6. (**a**) Synthetic transcriptional cascade. TetR represses *lacI*, LacI represses *cI*, and CI represses *eyfp*. aTc controls the repression of *lacI* by TetR. The fluorescence of the protein EYFP is the output. (**b**) Piecewise-multiaffine model of the cascade in (a). The concentrations of protein TetR, LacI, CI, EYFP and of aTc are denoted by x_{tetR}, x_{lacI}, x_{cI}, x_{eyfp} and u_{aTc}, respectively. Other notations follow those introduced in Section 2.1.

Downloaded from https://academic.oup.com/bioinformatics/article/23/18/2415/239215 by guest on 29 December 2023

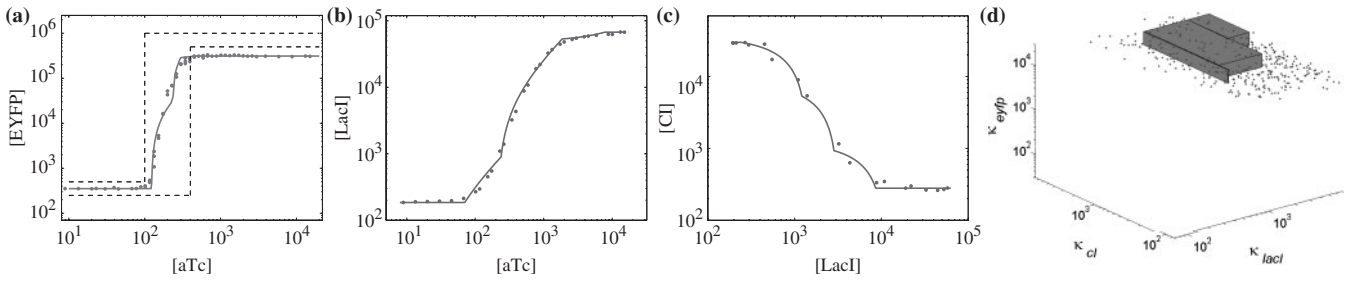


Fig. 7. (**a**) Steady-state input/output behavior of the cascade: desired (region delimited by black dashed lines) measured (red dots) and predicted (red line). (**b** and **c**) Relations between the concentrations of (b) LacI and aTc and (c) CI and LacI of the cascade at steady-state: experimental data (dots) and piecewise affine fits (solid lines). Note that the curved shape of the line segments is due to the log-log representation used. (**d**) Valid parameters in the parameter space as identified by RoVerGeNe (rectangular regions) or by brute-force sampling (dots). κ_{lacI}, κ_{cI} and κ_{eyfp} are production rate parameters for protein LacI, CI and EYFP, respectively.

to those introduced in Section 2. For regulated genes, two production terms are distinguished: a leakage term (with subscript 0) and a regulated term. The expression in Equation (6) for the regulation of *lacI* by TetR and aTc states that the expression of *lacI* increases if the concentration of aTc increases ($r_{lacI,2}$ is an increasing function since aTc is an activator) or if the concentration of TetR decreases ($r_{lacI,1}$ is a decreasing function since TetR is a repressor). We recall that the function $d(a,b) = a + b - ab$ increases when a or b increase and that $d(a,b) \in [0,1]$ if $a \in [0,1]$ and $b \in [0,1]$. It can thus be considered as an arithmetic equivalent of the logical *or*. Finally, Equation (9) states that the concentration of aTc is constant.

Because the proteins in the cascade are relatively stable, we neglected protein degradation and assumed that degradation rate parameters were simply equal to the dilution rate corresponding to the observed division time of the cells (about 45 minutes). Under the assumption that the concentration at steady state of the constitutively expressed protein TetR is sufficient to fully repress the expression of *lacI* (i.e. $r_{lacI,1}(x^*_{tetR}) = 0$), we deduce from our model that the concentrations at steady state of the proteins LacI, CI and EYFP satisfy the following relations.

$$x^*_{lacI} = \kappa_{lacI,0}/\gamma_{lacI} + \kappa_{lacI}/\gamma_{lacI}\ r_{lacI,2}(u_{aTc}) \quad (10)$$

$$x^*_{cI} = \kappa_{cI,0}/\gamma_{cI} + \kappa_{cI}/\gamma_{cI}\, r_{cI}(x^*_{lacI}) \quad (11)$$

$$x^*_{eyfp} = \kappa_{eyfp,0}/\gamma_{eyfp} + \kappa_{eyfp}/\gamma_{eyfp}\, r_{eyfp}(x^*_{cI}) \quad (12)$$

Under the assumption that the concentration of the protein EYFP corresponds to the fluorescence intensities measured for this cascade, and that the concentrations of the intermediate proteins of the cascade LacI and CI correspond to the fluorescence intensities of other, shorter cascades (not shown here, (Hooshangi *et al.*, 2005)), experimental data is available to describe these relations. More precisely, the relation between x^*_{lacI} and u_{aTc} is directly known from measurements, and the relations between x^*_{cI} and x^*_{lacI}, and x^*_{eyfp} and x^*_{cI} can be deduced from the experimentally-measured relations between x^*_{lacI} and u_{aTc}, x^*_{cI} and u_{aTc}, and x^*_{eyfp} and u_{aTc}. Then, existing techniques for fitting piecewise-affine functions to data can be used to identify the values of production rate parameters and the piecewise-affine regulation functions appearing in Equations (10)–(12) (Fig. 7b and c). We used an in-house implementation of the algorithm proposed in (Ferrari-Trecate *et al.*, 2001) that imposes the identification of horizontal plateaus. To obtain better fits, we interpolated the experimental data with splines, and fitted the piecewise-affine functions to the interpolated data.[3] For TetR, no experimental data was available. So, we have simply chosen a ramp function for $r_{lacI,1}(x_{tetR})$ and parameter values that guarantee that at steady state the concentration of TetR ($x^*_{tetR} = \kappa_{tetR}/\gamma_{tetR}$) is sufficient to fully repress the expression of $lacI$.

In order to assess the validity of the model, we compared model predictions with experimental data. For various concentrations of aTc and for randomly chosen initial concentrations, we computed the steady state of the network by numerical simulation (Fig. 7a). Given the simplicity of the model, we obtained a reasonably good fit between data and predictions. Additionally, we simulated the network response to the addition or removal of aTc and obtained a good agreement with experimental data (data not shown).

The last step was to formalize in temporal logic the desired behavior of the network depicted in Figure 7a. This specification can be expressed as a conjunction of three constraints of the type: if the aTc concentration is in a given range, then the concentration of EYFP at steady state must be in another given range:

$$\begin{aligned}\phi_2 = \quad & u_{aTc} < 100 \rightarrow \mathbf{FG}(2.5\,10^2 < x_{eyfp} < 5\,10^2) \\ \wedge\ & 100 < u_{aTc} < 400 \rightarrow \mathbf{FG}(2.5\,10^2 < x_{eyfp} < 10^6) \\ \wedge\ & u_{aTc} > 400 \rightarrow \mathbf{FG}(5\,10^5 < x_{eyfp} < 10^6),\end{aligned}$$

where to express that a property p holds at steady state, we used $\mathbf{FG}p$, meaning 'eventually (**F**), property p will always (**G**) hold'.

3.3 Parameter tuning

Using RoVerGene, we looked for parameter modifications that would improve the network behavior. Stated differently, we searched for valid parameters. We have chosen to tune production rate parameters, since recently developed techniques allow to tune promoter or ribosome binding site

[3]In Batt *et al.* (2007a,b), we used the simplest piecewise-affine function (i.e. a ramp function) to describe gene regulations. The use of more general piecewise-affine functions allows us to obtain a more faithful model.

Downloaded from https://academic.oup.com/bioinformatics/article/23/18/2415/239215 by guest on 29 December 2023

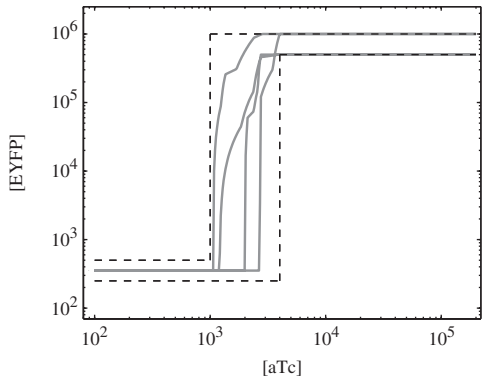


Fig. 8. Steady-state input/output behavior of the cascade for extreme parameter values in the valid parameter sets represented in Figure 7d.

efficiencies with relative ease and precision (Hammer *et al.*, 2006). So, we assumed that the production rate parameters of LacI, CI and EYFP were unconstrained, or more precisely, were ranging in intervals spanning at least two orders of magnitude (their original values are $\kappa_{lacI} = 875$, $\kappa_{cI} = 386$ and $\kappa_{eyfp} = 4048$):

$$\kappa_{lacI} \in [70, 7000], \kappa_{cI} \in [75, 8000] \text{ and } \kappa_{eyfp} \in [30, 30000]$$

Using RoVerGeNe, we identified 15 valid parameter sets (< 5 h, PC, 3.4 GHz processor, 1 GB RAM). The analysis of these sets, represented in the parameter space in Figure 7d, suggests to increase by at least 50% the production rates of LacI and CI, and to approximatively double the production rate of EYFP. In particular, this might be achieved by tuning ribosome-binding sites as done by Basu *et al.* (2004).

In order to illustrate the relevance of the constraints found, we considered extreme parameter values in these sets (i.e. vertices of the rectangular regions in Fig. 7d), and computed the input/output behavior of the network at steady state for these parameters (Fig. 8). This clearly illustrate that relevant constraints on the parameters were found.

We recall that not all valid parameters are guaranteed to be found by our method. So in order to evaluate the capacity of our approach to successfully identify valid parameters, we compared our results with those obtained by a brute-force sampling of the parameter space using numerical simulations. 20 000 different samples were considered. For each sample, we randomly chose parameter values and initial protein concentrations. Given that possible parameter values and initial protein concentrations span several orders of magnitude, we considered uniform distributions in the log-transformed spaces. Then, for four different aTc concentrations (10, 100, 400 and 20 10^4 nM), we simulated the network behavior and considered that a parameter value is valid if the concentration of EYFP at steady state satisfies the constraints depicted in Figure 7a. Note that 'valid' here has not the same meaning than previously, since the validity of a parameter is tested solely for one initial condition and four different aTc concentrations. Out of the 20 000 different parameter values considered, we found that 2.27% were valid (Fig. 7d). This figure should be compared with the fact that the volume of the valid parameter sets found using RoVerGeNe represents 1.8% of the volume of the (log-transformed) parameter space. These figures indicate that using RoVerGeNe we were able to identify a significant subset of the set of all valid parameters.

3.4 Robustness of the tuned network

Before experimentally modifying network parameters as suggested by the previous analysis, it is important to verify that the modified network will robustly behave as expected. So, we let all production and degradation rate parameters range in ±10% (or ±20%) intervals centered at their reference values and tested whether the property is robustly satisfied for these parameter variations. Eleven parameters were thus considered uncertain. For tuned parameters, new references values were chosen in the valid parameter sets found in the previous approach. More precisely, we chose $\kappa_{lacI} = 1600, \kappa_{cI} = 1400$ and $\kappa_{eyfp} = 8100$. Using RoVerGeNe, we proved that the property ϕ_2 holds for every parameter in the ±10% parameter set, and we were not able to prove that the same hold for the ±20% set. This proves that the tuned network presents the desired behavior for modest (±10%) parameter variations, and suggests that it does not do so for large (±20%) parameter variations. We recall that there are two reasons for obtaining negative results with RoVerGeNe: either because the property is false, or because our approach fails to prove the property, due to excessive approximations. In this case, a manual analysis of the output given by RoVerGeNe (or more precisely of the counter-example given by the model checker) revealed that for some parameters (minimal production rates and maximal degradation rate for EYFP) the concentration of EYFP at steady state is below the minimal value allowed by the specifications (5 10^5). So, as suggested by RoVerGeNe, the property is indeed not robustly satisfied for ±20% parameter variations. This analysis again illustrates that relevant constraints on parameters have been identified by our approach.

These results were obtained in less than 1h. Consequently, our approach can be considered as rather efficient, especially given the difficulty of the problem: verifying that a non-trivial dynamical property holds for all initial conditions in a 5-dimensional state space and for all parameters in an 11-dimensional parameter space.

4 CONCLUSION

We have presented a method for the analysis of dynamical properties of genetic regulatory networks with parameter uncertainty. Given a PMA model, an LTL specification of a dynamical property and intervals defining a set of uncertain parameters, the proposed approach deals with two problems. The first one is to *test* whether the property is satisfied for every parameter in the parameter set. The second one is to *find* subsets of the given parameter set such that the property is satisfied for every parameter in these subsets. Both problems are of practical importance in quantitative biology. The first one amounts to assess the robustness of the behavior of a network, in the sense that we show that the system presents a given behavior despite environmental fluctuations or

Downloaded from https://academic.oup.com/bioinformatics/article/23/18/2415/239215 by guest on 29 December 2023

inaccurate parameter estimation. The second one suggests parameter modifications to tune network behaviors and is particularly important for network design, since most initial attempts at constructing gene networks do not result in a system exhibiting the desired behavior.

The motivation for considering uncertain piecewise-multiaffine differential equation models is twofold. First, they are well adapted to model genetic regulatory networks in the face of incomplete quantitative information. This is of utmost importance for applications in systems and synthetic biology, since precise quantitative information are generally not available. Second, they present interesting mathematical properties that allow the development of efficient, tailored algorithms implementing a combination of techniques for the formal verification of continuous dynamical systems, based on discrete abstraction and model checking. These algorithms are implemented in the publicly-available tool RoVerGeNe, and their practical applicability and biological relevance are demonstrated on the tuning of a synthetic network built in *E.coli*.

To the best of our knowledge, the approach presented here is the first computational approach developed specifically for tuning synthetic gene networks. In a different context, Kuepfer *et al*. (2007) have recently developed an approach based on semidefinite programming for partitioning the parameter space of polynomial differential equation models into so-called feasible and infeasible regions. It should be noted that in this approach, 'feasible' simply refers to the existence of a steady state of the system. In contrast, our approach allows to find parameter sets for which the system presents a particular *behavior*, expressed in the rich language of temporal logics.

Finally, the most promising direction for future applications seems to be the use of the proposed methods to support the modular design of large gene networks (Chin, 2006). In this perspective, the behavior of each module (i.e. sub-network) could be described by a temporal logic property that holds for sets of parameters, initial conditions and inputs. These properties could then be viewed as *certificates* of the robust behavior of the modules and could be used to support module assemblage on a sound basis. In particular, this would pave the way for the approach advocated by Collins and colleagues (Kobayashi *et al*., 2004) in which biologists use network modules as 'plug-and-play' devices to build complex synthetic systems.

ACKNOWLEDGEMENTS

We would like to thank Ilaria Mogno for sharing her implementation of the algorithm for piecewise-affine regression and the reviewers for their interesting comments. We acknowledge financial support by NSF 0432070.

Conflict of Interest: none declared.

REFERENCES

Alur,R. *et al*. (2000) Discrete abstractions of hybrid systems. *Proc. IEEE*, **88**, 971–984.

Andrianantoandro,E. *et al*. (2006) Synthetic biology: new engineering rules for an emerging discipline. *Mol. Syst. Biol*., **2**, 0028.

Antoniotti,M. *et al*. (2003) Model building and model checking for biochemical processes. *Cell Biochem. Biophys*., **38**, 271–286.

Basu,S. *et al*. (2004) Spatiotemporal control of gene expression with pulse-generating networks. *Proc. Nalt Acad. Sci. USA*, **101**, 6355–6360.

Batt,G. *et al*. (2005) Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in *E. coli*. *Bioinformatics*, **21** (Suppl.1), i19–i28.

Batt,G. *et al*. (2007a) Model checking genetic regulatory networks with parameter uncertainty. In Bemporad,A. *et al*. (eds). *Hybrid Systems: Computation and Control, HSCC'07, Lecture Notes in Computer Science 4416*. Springer, Berlin, pp. 61–75.

Batt,G. *et al*. (2007b) Model checking liveness properties of genetic regulatory networks. In Grumberg,O. and Huth,M. (eds). *Tools and Algorithms for the Construction and Analysis of Systems, TACAS'07, Lecture Notes in Computer Science 4424*. Springer, Berlin, pp. 323–338.

Belta,C. and Habets,L.C.G.J.M. (2006) Controlling a class of nonlinear systems on rectangles. *Trans. Aut. Control*, **51**, 1749–1759.

Belta,C. *et al*. (2002) Control of multi-affine systems on rectangles with applications to hybrid biomolecular networks. In *IEEE Conference on Decision and Control, CDC'02*.

Bernot,G. *et al*. (2004) Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *J. Theor. Biol*., **229**, 339–347.

Browne,M.C. *et al*. (1988) Characterizing finite Kripke structures in propositional temporal logic. *Theor. Comput. Sci*., **59**, 115–131.

Chin,J.W. (2006) Modular approaches to expanding the functions of living matter. *Nature Chem. Biol*., **2**, 304–311.

Clarke,E.M. *et al*. (1999) *Model Checking*, MIT Press, Cambridge, USA.

de Jong,H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Bio*., **9**, 69–105.

de Jong,H. *et al*. (2004) Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol*., **66**, 301–340.

Emerson,E.A. (1990) Temporal and modal logic. In van Leeuwen,J. (eds). *Handbook of Theoretical Computer Science*. vol. B: Formal Models and Sematics, MIT Press. Cambridge, USA, pp. 995–1072.

Endy,D. (2005) Foundations for engineering biology. *Nature*, **438**, 449–453.

Fages,F. *et al*. (2004) Modelling and querying interaction networks in the biochemical abstract machine biocham. *J. Biol. Phys. Chem*., **4**, 64–73.

Ferrari-Trecate,G. *et al*. (2001) A learning algorithm for piecewise linear regression. In Marinaro,M. and Tagliaferri,R. (eds.) *Neural Nets: WIRN Vietri-01*. Springer, Berlin, pp. 114–119.

Gardner,T.S. *et al*. (2000) Construction of a genetic toggle switch in *E. coli*. *Nature*, **403**, 339–342.

Ghosh,R. and Tomlin,C.J. (2004) Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-Notch protein signalling. *IEE Proc. Syst. Biol*., **1**, 170–183.

Glass,L. and Kauffman,S.A. (1973) The logical analysis of continuous non-linear biochemical control networks. *J. Theor. Biol*., **39**, 103–129.

Hammer,K. *et al*. (2006) Synthetic promoter libraries — tuning of gene expression. *Trends Biotechnol*, **24**, 53–55.

Hooshangi,S. *et al*. (2005) Ultrasensitivity and noise propagation in a synthetic transcriptional cascade. *Proc. Nalt Acad. Sci. USA*, **102**, 3581–3586.

Kauffman,S.A. (1969) Homeostasis and differentiation in random genetic control networks. *Nature*, **224**, 177–178.

Kobayashi,H. *et al*. (2004) Programmable cells: interfacing natural and engineered gene networks. *Proc. Nalt Acad. Sci. USA*, **101**, 8414–8419.

Kuepfer,L. *et al*. (2007) Efficient classification of complete parameter regions based on semidefinite programming. *BMC Bioinformatics*, **8**, 12.

Lin,J.N. and Unbehauen,R. (1992) Canonical piecewise-linear approximations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **39**, 697–699.

McDaniel,R. and Weiss,R. (2005) Advances in synthetic biology: on the path from prototypes to applications. *Curr. Opin. Biotechnol*., **16**, 476–483.

Mestl,T. *et al*. (1995) A mathematical framework for describing and analysing gene regulatory networks. *J. Theor. Biol*., **176**, 291–300.

Szallasi,Z. *et al*. (2006) (eds) *System Modeling in Cellular Biology: From Concepts to Nuts and Bolts*. MIT Press, Cambridge, USA.

Thomas,R. *et al*. (1995) Dynamical behaviour of biological regulatory networks: I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bull. Math. Biol*., **57**, 247–276.

Downloaded from https://academic.oup.com/bioinformatics/article/23/18/2415/239215 by guest on 29 December 2023