

A general computational method for robustness analysis with applications to synthetic gene networks

Aurélien Rizk, Gregory Batt, François Fages and Sylvain Soliman

INRIA Paris-Rocquencourt, 78153 Le Chesnay Cedex, France

Received on January 7, 2009

Associate Editor: XXXXXXXX

ABSTRACT

Motivation: Robustness is the capacity of a system to maintain a function in the face of perturbations. It is essential for the correct functioning of natural and engineered biological systems. Robustness is generally defined in an ad-hoc, problem-dependent manner, thus hampering the fruitful development of a theory of biological robustness, advocated by Kitano [Mol Syst Biol, 3:137, 2007].

Results: In this paper, we propose a general definition of robustness that applies to any biological function expressible in temporal logic LTL, and to broad model classes and perturbation types. Moreover, we propose a computational approach and an implementation in BIOCHAM 2.8 for the automated estimation of the robustness of a given behavior with respect to a given set of perturbations. The applicability and biological relevance of our approach is demonstrated by testing and improving the robustness of the timed behavior of a synthetic transcriptional cascade that could be used as a biological timer for synthetic biology applications.

Availability: Version 2.8 of BIOCHAM and the transcriptional cascade model are available at <http://contraintes.inria.fr/BIOCHAM/>

Contact: gregory.batt@inria.fr

1 INTRODUCTION

Robustness can be defined as the capacity of a system to maintain a function in the face of perturbations. Over the years, many studies have demonstrated theoretically and experimentally that robustness is a key property of numerous biological processes, and have proposed mechanisms that promote robustness (e.g. [3, 12, 34, 5, 35, 22, 36, 15, 11, 39, 25, 9]). Robustness is now regarded as one of the fundamental characteristics of biological systems because it allows their correct functioning in presence of molecular noise and environmental fluctuations. Excellent reviews have surveyed the role of biological robustness, and discussed its interesting relations with evolvability of biological systems, modularity of biological networks and the trade-off between robustness and fragility (e.g. [26, 40, 37]). In particular, in the context of synthetic biology applications, these are key issues to take into account at the design level.

Intuitively, the notion of robustness seems easy to define. One considers (i) a particular system, (ii) a particular function, and (iii) a particular set of perturbations, and one assesses how perturbations affect (or not) the given function. However, with the notable exception of Kitano [27], no general formal definition of robustness has been proposed. The precise definition of robustness is generally highly problem-specific. This makes it difficult to discuss and compare the robustness found in different contexts, or even in

similar contexts but computed using different formal definitions of robustness. In [27], Kitano proposes the mathematical foundations of a theory of biological robustness with the aim of providing a unified perspective on robustness.

Although very interesting from a theoretical point of view, Kitano's definition might be too general when applying it to particular problems. Indeed the definition relies on a so-called evaluation function, defined using an unspecified, problem-dependent real-valued performance function. Here, we propose to define the evaluation function using the newly-introduced notion of violation degree of temporal logic formulae [17]. Intuitively, the violation degree reflects the distance between a particular behavior of the perturbed system, given as a numerical timed trace, and the expected reference behavior, expressed by a temporal logic formula. Because (i) temporal logics are expressive languages to formalize temporal behavior of dynamical systems and (ii) the violation degree can be automatically computed, our instantiation of Kitano's definition is both *general* and *computational*. The main contribution of our work is that we propose a computational approach for -and an implementation of- the automatic estimation of the robustness that applies to a broad class of dynamical properties and a large variety of possible perturbations. We simply require that the property describing the expected behavior can be expressed in temporal logic and that the behavior of the system can be represented by a numerical trace (possibly obtained by numerical simulation of deterministic or stochastic models).

A second contribution of this work is that we propose two closely-related but different notions of robustness that have been used indiscriminately in publications, namely the *absolute robustness* of a system, representing the average functionality of the system under perturbations, and the *relative robustness* w.r.t. a given nominal behavior of the system, quantifying the impact of perturbations on the nominal behavior. We believe that distinguishing these two notions will help to clarify the analysis of robustness of biological systems. Undoubtedly, formal definitions are useful for making this distinction.

The applicability and biological relevance of our approach is illustrated on the analysis of the robustness of the timed response of a synthetic transcriptional cascade built in *E. coli*. This system presents a high cell-to-cell variability that prevents using it as a biological timer. We look for parameter modifications that improve the robustness of a "well-timed" behavior.

The remainder of this paper is organized as follows. In the next section, we provide a brief description of the notion of

violation degree introduced in [17]. In Section 3, we present the proposed method for robustness estimation and its implementation in BIOCHAM 2.8. In Section 4, we detail the application of our method to the analysis of the robustness of the synthetic transcriptional cascade.

2 VIOLATION DEGREE OF TEMPORAL LOGIC PROPERTIES

We first define the Boolean semantics of LTL on timed numerical traces (Section 2.1). Then, we show how using the variable abstraction technique of Section 2.2, we can define a continuous satisfaction degree for temporal logic formulae (Section 2.3) better adapted to a quantitative notion of robustness.

2.1 Temporal logic semantics of numerical traces

In this article, we consider that the behavior of a biological system is described by numerical timed traces. These traces can be obtained either by experimentation on the actual system or by numerical simulation of stochastic or deterministic models. Formally, a *numerical trace* is a finite sequence of tuples describing system's evolution with time: $T = (s_0, s_1, \dots, s_n)$, with $s_i = (t_i, \mathbf{x}_i, \dot{\mathbf{x}}_i)$, $(t_i)_{i \in [0, n]}$ being a strictly increasing sequence of time points, and $\mathbf{x}_i, \dot{\mathbf{x}}_i \in \mathbb{R}^m$ being vectors of state variable values and of their derivatives at time t_i . In Fig 1, a hypothetical evolution of a protein concentration is represented. The associated trace is $T = ((0, 6, 1.3), (2, 8, 0.8), \dots, (24, 5, 0))$.

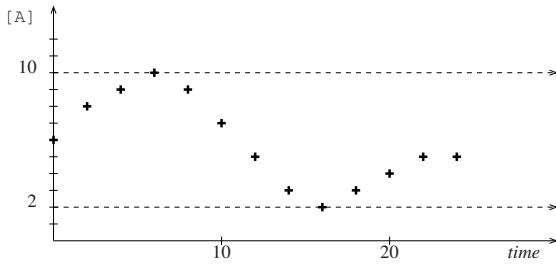


Fig. 1. Numerical trace depicting the time evolution of a protein concentration

We use Linear Temporal Logic (LTL) to express dynamical properties of biological systems. Temporal logics have been developed to specify the behavior of (usually discrete) dynamical systems [16]. 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 of events (such as oscillations). Illustrative examples of the expressiveness of temporal logics in systems biology can be found in [1, 8, 4, 6, 7]. LTL formulae are built using atomic propositions and LTL operators.

In our approach, atomic propositions π express real-valued linear constraints on time, protein concentrations and their derivatives. The infinite set of atomic propositions is denoted by Π .

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 *next* ($\mathbf{X}$), *future* ($\mathbf{F}$), *globally* ($\mathbf{G}$) and *until* ($\mathbf{U}$). $\mathbf{X}\phi$, $\mathbf{F}\phi$, $\mathbf{G}\phi$ and $\phi \mathbf{U} \psi$ respectively

mean that a property ϕ holds at the next time, at some future time, holds for all future times, or holds continuously until another property ψ holds. These operators can be combined to express complex dynamical properties. For example, the trace T represented in Fig. 1 satisfies the formula $\phi_1 = \mathbf{F}([A] > 7 \wedge \mathbf{F}[A] < 3)$, expressing that at some time point, protein A concentration exceeds 7 and later goes below 3. Because negations can be pushed to atomic propositions with the usual duality properties of operators, and the set of atomic propositions is closed by negation, we consider without loss of generality only negation-free LTL formulae.

The standard semantics of LTL formulae is generally defined with respect to infinite executions, *i.e.* infinite traces. Because in our case, the traces are finite, the usual semantics of LTL has to be adapted. Let $T = (s_0, s_1, \dots, s_n)$ be a finite numerical trace, $\pi \in \Pi$ be an atomic proposition and ϕ, ψ be LTL formulae. Then the semantics of LTL formulae with respect to finite traces is defined inductively as

- $T \models \phi$ iff $s_0 \models \phi$,
- $s_i \models \pi$ iff $s_i = (t_i, \mathbf{x}_i, \dot{\mathbf{x}}_i)$ satisfies π with the usual semantics,
- $s_i \models \phi \wedge \psi$ iff $s_i \models \phi$ and $s_i \models \psi$,
- $s_i \models \phi \vee \psi$ iff $s_i \models \phi$ or $s_i \models \psi$,
- $s_i \models \mathbf{X}\phi$ iff $i < n$ and $s_{i+1} \models \phi$, or $i = n$ and $s_n \models \phi$,
- $s_i \models \mathbf{F}\phi$ iff $\exists j \in [i, n]$ such that $s_j \models \phi$,
- $s_i \models \mathbf{G}\phi$ iff $\forall j \in [i, n]$, $s_j \models \phi$,
- $s_i \models \phi \mathbf{U} \psi$ iff $\exists j \in [i, n]$ s. t. $s_j \models \psi$ and $\forall k \in [i, j-1]$, $s_k \models \phi$.

The first equivalence simply states that the trace satisfies a formula if the formula holds at the initial time t_0 . This semantics of LTL coincides with the standard semantics used on finite traces completed by a self-loop on the last state [17]. This semantics differs from the neutral semantics of Eisner and colleagues [14] for finite traces only for the next operator, which in their definition is always false on the last state, whereas in our case it enjoys the duality property $\neg \mathbf{X}\phi = \mathbf{X}\neg\phi$ and either $\mathbf{X}\phi$ or $\mathbf{X}\neg\phi$ holds. In practice, the next operator being mainly used to detect local extrema, this difference of interpretation is not significant.

It is worth noticing that when the numerical trace corresponds to a discrete representation of a continuous process, the discrete time semantics that we use may cause that particular events are “missed” independently of the numerical errors that can be made by the numerical integration method. For example, the formula $\mathbf{F}([A] \geq 10)$ interpreted on trace T of Fig. 1 and expressing that eventually $[A]$ exceeds 10 might be found true or false depending on the integration step and precision. So care must be taken when checking temporal properties on finite discrete time traces (for a discussion, see [14, 19, 28], and references therein).

2.2 From model checking to constraint solving

The Boolean interpretation of temporal logic is not well adapted to defining a quantitative notion of robustness. Indeed, neither of the two formulae $\phi_2 = \mathbf{F}([A] > 12 \wedge \mathbf{F}[A] < 3)$ and $\phi_3 = \mathbf{F}([A] > 14 \wedge \mathbf{F}[A] < 3)$ hold for the trace T of Fig 1. However, intuitively ϕ_2 is closer to satisfaction than ϕ_3 , since it only requires that $[A]$ reaches 12 instead of 14.

To provide a formal definition, for a continuous degree of satisfaction of an LTL formula, we consider quantifier-free LTL (QFLTL) formulae with free (non state) real-valued variables $\mathbf{y}$ [17]. Then, the original model checking problem is transformed into the following *constraint solving problem*: for which values $\mathbf{y}$ does $\phi(\mathbf{y})$ hold on T ? Accordingly, we define for any trace T the *satisfaction domain* of the formula $\phi(\mathbf{y})$ as the set of $\mathbf{y}$ values for which $\phi(\mathbf{y})$ holds:

$$\mathcal{D}_{T,\phi(\mathbf{y})} = \{\mathbf{y} \in \mathbb{R}^q \mid T \models \phi(\mathbf{y})\} \quad (1)$$

In the sequel, $\phi(\mathbf{y})$ will denote the QFLTL formula obtained by variable abstraction from a (QF)LTL formula ϕ .

Interestingly, an LTL formula can be seen as an instance of a QFLTL formula obtained by abstracting the constants appearing in the formula by new variables $\mathbf{y} \in \mathbb{R}^q$. For example, to $\phi_1 = \mathbf{F}([A] > 7 \wedge \mathbf{F}[A] < 3)$, we associate the formula $\phi(\mathbf{y}) = \phi(y_1, y_2) = \mathbf{F}([A] > y_1 \wedge \mathbf{F}[A] < y_2)$. Then we have $\phi_1 = \phi(7, 3)$. Moreover, one can easily check that for our example trace T , $\mathcal{D}_{T,\phi(y_1,y_2)} = \{y_1 \leq 10 \wedge y_2 \geq 2\}$, 10 and 2 being respectively the maximal and minimal values of $[A]$ in T .

More generally, this variable abstraction/instantiation process allows us to view a LTL formula as a point in the QFLTL formula space $\mathbb{R}^q$, where q is the number of constants appearing in ϕ (or the number of constants that are replaced by variables, if not all constants are abstracted away). In Fig 2, ϕ_1 , ϕ_2 , ϕ_3 , and $\mathcal{D}_{T,\phi}$ are represented in this formula space.

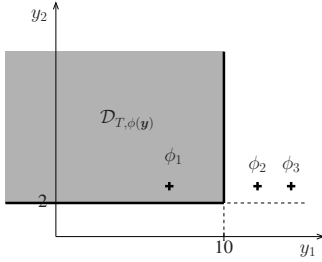


Fig. 2. Satisfaction domain $\mathcal{D}_{T,\phi(\mathbf{y})}$ of QFLTL formula $\phi(\mathbf{y}) = \mathbf{F}([A] > y_1 \wedge \mathbf{F}[A] < y_2)$ and trace T , and LTL formulae represented in formula space.

Given any trace $T = (s_0, s_1, \dots, s_n)$ and formula $\phi(\mathbf{y})$ we showed in [17] that the satisfaction domain $\mathcal{D}_{T,\phi(\mathbf{y})}$ can be computed by induction on T and the subformulae of $\phi(\mathbf{y})$ using the equalities of Proposition 1.

PROPOSITION 1 (Computation of satisfaction domains [17]).

- $\mathcal{D}_{T,\phi(\mathbf{y})} = \mathcal{D}_{s_0,\phi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\pi(\mathbf{y})} = \{\mathbf{y} \in \mathbb{R}^m \mid \pi(\mathbf{y}) \text{ holds with the usual semantics}\}$,
- $\mathcal{D}_{s_i,\phi(\mathbf{y}) \wedge \psi(\mathbf{y})} = \mathcal{D}_{s_i,\phi(\mathbf{y})} \cap \mathcal{D}_{s_i,\psi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\phi(\mathbf{y}) \vee \psi(\mathbf{y})} = \mathcal{D}_{s_i,\phi(\mathbf{y})} \cup \mathcal{D}_{s_i,\psi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\mathbf{X}\phi(\mathbf{y})} = \begin{cases} \mathcal{D}_{s_{i+1},\phi(\mathbf{y})}, & \text{if } i < n, \\ \mathcal{D}_{s_i,\phi(\mathbf{y})}, & \text{if } i = n, \end{cases}$
- $\mathcal{D}_{s_i,\mathbf{F}\phi(\mathbf{y})} = \bigcup_{j \in [i,n]} \mathcal{D}_{s_j,\phi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\mathbf{G}\phi(\mathbf{y})} = \bigcap_{j \in [i,n]} \mathcal{D}_{s_j,\phi(\mathbf{y})}$,
- $\mathcal{D}_{s_i,\phi(\mathbf{y}) \mathbf{U} \psi(\mathbf{y})} = \bigcup_{j \in [i,n]} (\mathcal{D}_{s_j,\psi(\mathbf{y})} \cap \bigcap_{k \in [i,j-1]} \mathcal{D}_{s_k,\phi(\mathbf{y})})$.

Here, the atomic propositions in formula $\phi(\mathbf{y})$ being linear constraints on free variables (y) and values of the state variables (*time*, x_j , or $\dot{x}_j$), the satisfaction domains are finite unions and intersections of polytopes which can be computed with standard polyhedral libraries.

2.3 Violation degree

To quantify how far from satisfaction a system's behavior is, we introduce the *violation degree* $vd(T, \phi)$ of a formula ϕ with respect to trace T as the *distance* between the actual specification and validity domain $\mathcal{D}_{T,\phi(\mathbf{y})}$ of the QFLTL formula $\phi(\mathbf{y})$ obtained by variable abstraction:

$$vd(T, \phi) = \text{dist}(\phi, \mathcal{D}_{T,\phi(\mathbf{y})}).$$

The violation degree has thus a simple interpretation, since it *quantifies by how much a given numerical trace must be changed to satisfy the given formula*.

Considering again our example in Fig 2 and using the Euclidean distance, we have that $vd(T, \phi_1) = 0$, meaning that the formula is satisfied by T , and $vd(T, \phi_2) = 2$ and $vd(T, \phi_3) = 4$, reflecting that T is further from satisfaction of ϕ_3 than of ϕ_2 .

We would like to emphasize that abstracting constants by variables in temporal logic formulae is a means to define a *metric* on the set of formulae. All set operations and distance computations are made in the corresponding metric space, known as the formula space. It might seem more intuitive to define distances directly between traces. For example, Fainekos and Pappas [18] use with a similar aim -defining a continuous interpretation of temporal logic formulae on traces- the distance between a given trace T and the set of traces satisfying a formula ϕ . One major advantage of our approach is that the dimensionality of the formula space (number of abstracted constants) is generally much lower than the dimensionality of the trace space (trace length). Performing set operations and distance computation in low dimensional spaces may strongly affect the practical applicability of these methods. In this article, we use the Euclidean distance. However, many other distances can be used (*e.g.* Manhattan or Chebyshev), depending on the desired interpretation of distance and, as we will see in the next paragraph, on the desired interpretation of robustness.

To define the robustness of a behavior, it is more convenient to reason with a positive notion that describes how good the (possibly perturbed) system performs, *i.e.* satisfies a dynamical property. To do so, we introduce the notion of continuous *satisfaction degree* of a formula with respect to a trace T :

$$sd(T, \phi) = \frac{1}{1 + vd(T, \phi)} \in [0, 1], \quad (2)$$

where vd the violation degree previously introduced. The satisfaction degree is normalized such that it ranges between 0 and 1, with a satisfaction degree equal to 1 when the property is true and tending towards 0 when the system is far from satisfying the expected property. For some applications, the satisfaction degree might be normalized differently, using a given constant K instead of the ones in equation (2).

3 ROBUSTNESS DEFINITIONS AND COMPUTATIONS

3.1 Absolute robustness

In this article, we mainly use Kitano's general definition of robustness. In [27], Kitano defines the robustness of a property a of a system s with respect to a set of perturbations P as the *average* of an evaluation function D_a^s of the system over all perturbations $p \in P$, weighted by the perturbation probabilities $prob(p)$:

$$R_{a,P}^s = \int_{p \in P} prob(p) D_a^s dp \quad (3)$$

One should emphasize that this definition is very *general* and can be used in many cases. Unfortunately, Kitano does not provide much information on how to define the so-called *evaluation function* D_a^s of the system. This function should determine if the system still maintains its function under a perturbation and to what degree. The evaluation function needs to be defined for each specific problem in an ad-hoc manner and re-implemented for the computation of the robustness. A central contribution of this paper is to demonstrate that using the notion of satisfaction degree presented previously, one can provide a *general computational framework* based on temporal logic and Kitano's definition that can be used to evaluate the robustness of broad types of dynamical properties and perturbations.

Formally, the robustness of the system is defined as:

$$R_{\phi,P}^s = \int_{p \in P} prob(p) sd(T_p, \phi) dp, \quad (4)$$

where ϕ is the specification of the functionality in temporal logic and T_p is the trace representing the behavior of the system under perturbation p . This notion of *robustness* corresponds to a mean functionality, that is, describes on average how the system behaves under perturbations. To illustrate this, consider the plots 1-2 of Fig. 3 that describe the performance D_a^s -or equivalently in our case, the satisfaction degree- of two hypothetical systems in the face of perturbations p . Because these two plots have the same average, the robustness of these three systems would be equal for evenly-distributed perturbations. For example in a bioengineering context, if the "property" reflects the quantity of some product exported by cells, these two systems will indeed produce on average the same quantity of the desired product.

3.2 Relative robustness

When comparing plots 1 and 2 of Fig 3, it appears that the consequences of perturbations of the nominal behavior are not the same, with T_0 the nominal behavior. In system 1, perturbations degrade the system's performance more severely than in system 2. So, with a different meaning of robustness, one could say that system 2 is more robust than system 1. These two robustness interpretations (as average behavior or as impact of perturbations on nominal behavior) have been indiscriminately used in the literature (see e.g. [39, 13]). To reflect this second interpretation, let us define the *relative robustness* of a system w.r.t. a nominal behavior as the system's robustness divided by its nominal performance, that is, by the satisfaction degree of the reference behavior.

$$R_{\phi,P}^{s,p*} = R_{\phi,P}^s / sd(T_{p*}, \phi), \quad (5)$$

where T_{p*} denotes the unperturbed, nominal behavior of the system. In Fig. 3, one can distinguish the relative robustness of

systems 1 and 2 w.r.t. their nominal performance, reflecting that the performance is more impacted by perturbations in system 1 than in system 2. The performance function of system 3 equals half of the performance function of system 1. Consequently, these systems have the same relative robustness w.r.t. their nominal performance, although they have different absolute robustness.

3.3 Robust satisfaction degree

Using the notion of satisfaction domain, we can also define the *distance from robust satisfaction* of a property ϕ with respect to a set of perturbations P as $dist(\cap_p \mathcal{D}_{T_p}, \phi)$. This distance reflects the minimal change in the formula such that it holds for *all* perturbations. Then, we define the *robust satisfaction degree* as:

$$Rsd_{\phi,P}^s = \frac{1}{1 + dist(\cap_p \mathcal{D}_{T_p}, \phi)} \quad (6)$$

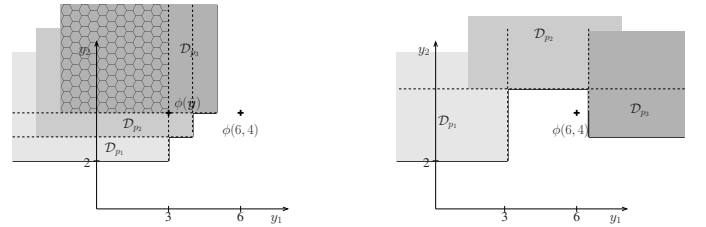


Fig. 4. Representation of satisfaction domains for three perturbations p_1 , p_2 , and $p_3 \in P$. $\mathcal{D}_{p_i}$ denotes $\mathcal{D}_{T_{p_i}, \phi(y)}$. Left: the intersection of satisfaction domains (shaded) is not empty and $Rsd_{\phi,P}^s = 3$. The property $\phi(\tilde{y}) = \phi(3, 4) = \mathbf{F}([A] > 3 \wedge \mathbf{F}[A] < 4)$ is satisfied for all perturbations. Right: the intersection of satisfaction domains is empty and $Rsd_{\phi,P}^s = \infty$.

This notion allows us to distinguish whether it is possible to relax the specification to have it satisfied for all perturbations or not. In the first case of Figure 4, one can guarantee that the system always presents a (possibly suboptimal) behavior. Moreover, the closest property $\phi(\tilde{y})$ robustly satisfied (i.e. such that $\tilde{y} = \text{argmin}_{y \in \cap_p \mathcal{D}_{T_p}} dist(\cap_p \mathcal{D}_{T_p}, \phi)$) can provide interesting hints for the system's design: because $\tilde{y} = (0, 3)$, it suggests that only the first value in ϕ (i.e. the maximum of $[A]$ in T) needs to be modified.

3.4 Implementation

For the computation of $R_{\phi,P}^s$, $R_{\phi,P}^{s,p*}$, and $Rsd_{\phi,P}^s$, one needs to distinguish whether the set of perturbations is finite (e.g. gene knockouts) or infinite (e.g. normally-distributed parameter variations). In the first case, the values can be computed exactly, whereas in the second case, they can be estimated by sampling the perturbation set for sufficiently many perturbations.

The following algorithm is implemented in version 2.8 of the freely-available tool BIOCHAM, a modeling environment for the analysis of biological systems [7]. Given an ODE model f , a set P of perturbations of initial conditions or parameters, and (QF)LTL properties ϕ and $\phi(y)$, the tool computes the robustness, the relative robustness and the robust satisfaction degree of the property with respect to the given perturbations. The computation of the trace T_p is done by numerical integration. The computation of the satisfaction domain $\mathcal{D}_{T_p, \phi(y)}$ is made by induction on the formula structure, using for each subformula a direct implementation of

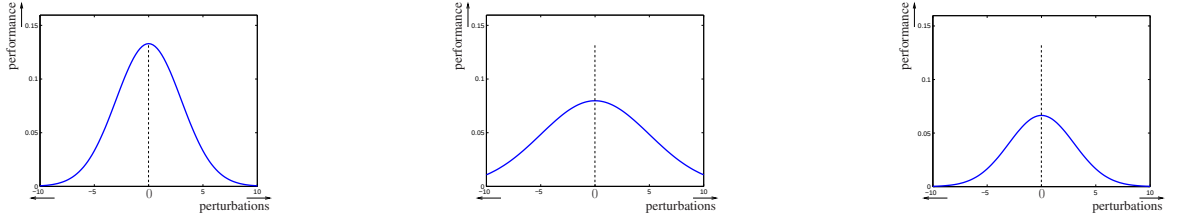


Fig. 3. Systems having same absolute robustness (1 and 2) or same relative robustness (1 and 3), assuming evenly-distributed perturbations. Performance functions of system 1 and 2 have same the average, whereas the performance function of system 3 is half of the one of system 1.

Algorithm 1 Robustness computation

```

// input: a (model of the) system  $f$ , (QF)LTL formulae  $\phi$  and  $\phi(y)$ , a set of perturbations  $P$  and their probabilities, and nominal behavior  $p^*$ 
// output: robustness estimates  $R_{\phi,P}$ ,  $R_{\phi,P}^{p^*}$ , and  $Rsd_{\phi,P}$ 
1: for every perturbation  $p \in P \cup \{p^*\}$  do
2:    $T_p := \text{COMPUTE\_TRACE}(f, p)$ 
3:    $\mathcal{D}_{T_p, \phi(y)} := \text{COMPUTE\_SAT\_DOMAIN}(T_p, \phi(y))$ 
4: end for
5:  $R_{\phi,P} := \sum_{p \in P} \text{prob}(p) (1 + \text{dist}(\mathcal{D}_{T_p, \phi(y)}, \phi))^{-1}$ 
6:  $R_{\phi,P}^{p^*} := R_{\phi,P} * (1 + \text{dist}(\mathcal{D}_{T_{p^*}, \phi(y)}, \phi))$ 
7:  $Rsd_{\phi,P} := (1 + \text{dist}(\cap_{p \in P} \mathcal{D}_{T_p, \phi(y)}, \phi))^{-1}$ 

```

the definition. Polytopes operations are implemented in BIOCHAM using a standard polyhedral library [2].

4 APPLICATION TO ROBUSTNESS ANALYSIS OF A TRANSCRIPTIONAL CASCADE

We consider the design of a synthetic transcriptional cascade that could be used for the temporal sequencing of events in synthetic biology applications. This cascade has already been built by Hooshanghi and colleagues [24] and here we investigate the robustness of a desired behavior, and the possibilities to make it more robust. To do so, after having introduced the system, we formalize the expected behavior, develop a model of the system taking into account the observed variability, and apply the method presented previously to investigate the robustness of the desired property.

4.1 System description

We consider a cascade of transcriptional inhibitions built in *E.coli* [24]. The network is represented in Figure 5. 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, it has been shown that the time response of the system to an inducer addition is characterized by a rapid increase of the fluorescence, preceded by a significant lag-phase. Unfortunately, a high cell-to-cell variability has also been observed. The heterogeneity of the cell responses

makes it difficult to use this system as a biological timer, for example for developmental programs as suggested in [24]. In this context, as for many synthetic biology applications, having even a low proportion of cells sending a signal too early or too long might compromise the correct functioning of the whole system. Our goal here is precisely to investigate the possibilities to obtain a robustly “well-timed” system, that is to ensure that all cells will indeed change state in a given time window.

4.2 Specifying the expected behavior

Here we consider that the system is well-timed if the fluorescence remains below 10^3 for at least 150 minutes, then exceeds 10^5 after at most 450 minutes, and switches rapidly from low to high levels, that is, in less than 150 minutes. These specifications are consistent with the experimentally observed behavior of the cell population. These specifications are graphically-represented in Fig. 6 and can be formalized in temporal logic as follows:

$$\begin{aligned} \phi(t_1, t_2) = & \mathbf{G}(time < t_1 \rightarrow [\text{EYFP}] < 10^3) \\ & \wedge \mathbf{G}(time > t_2 \rightarrow [\text{EYFP}] > 10^5) \\ & \wedge t_1 > 150 \wedge t_2 < 450 \wedge t_2 - t_1 < 150 \end{aligned}$$

which is abstracted into

$$\begin{aligned} \phi(t_1, t_2, b_1, b_2, b_3) = & \mathbf{G}(time < t_1 \rightarrow [\text{EYFP}] < 10^3) \\ & \wedge \mathbf{G}(time > t_2 \rightarrow [\text{EYFP}] > 10^5) \\ & \wedge t_1 > b_1 \wedge t_2 < b_2 \wedge t_2 - t_1 < b_3 \end{aligned}$$

for the computation of validity domains and satisfaction degree in a given trace.

4.3 Modeling the system’s variability

There are many ways to model cell variability (see for example [29]). Our goal here is to construct a simple model such that the predicted behavior and standard deviation are in

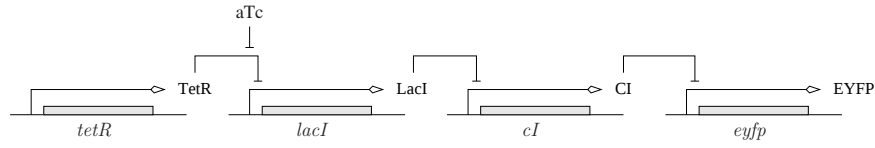


Fig. 5. Synthetic transcriptional cascade. TetR represses *lacI*, LacI represses *cI*, and CI represses *eyfp*. aTc controls the repression of *lacI* by TetR. The fluorescence of the protein EYFP is the output.

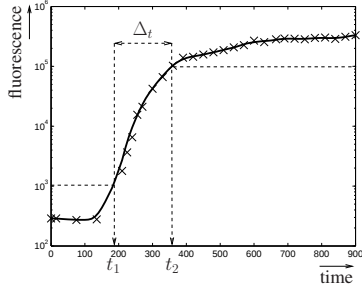


Fig. 6. Graphical representation of a “well-timed” behavior: fluorescence remains below 10^3 until time t_1 , exceeds 10^5 after time t_2 and switches between low and high levels in Δ_t time. One expects that $t_1 > 150$, $t_2 < 450$ and $\Delta_t < 150$. Crosses represent experimental data from [24].

agreement with the available experimental data. We first develop a simple ODE model similar to [5] but using Hill functions, with parameters fitted to experimental data (Fig 7(a) and (b)). These parameter reference values are denoted $\mathbf{p}^*$ in the sequel. Second, we consider various ways to model cell variability, including stochastic differential equations with either additive or multiplicative noise and random parameter variations with (log)normal distributions. We have obtained a good qualitative and quantitative agreement between the predicted and observed mean and standard deviation for log-normally distributed parameters, as shown in Fig 8(a) and (b). So we selected these log-normal parameter distributions as our “perturbation model”. Using either stochastic differential equations or normally-distributed parameters, we have not been able to find an agreement between model predictions and experimental observations (not shown). This could be partially explained by the very high cell-to-cell variability. In particular, the observed coefficient of variation reaches 1.4 at some time point, meaning that the standard deviation is higher than the mean.

4.4 Improving robustness of the desired behavior

Having specified the “well-timed” behavior and found an ODE model and a perturbation model, we wondered whether the system is robustly well-timed, and to what degree. When considering 5000 log-normally distributed parameter values, we estimated the robustness of the system as $\hat{R}_{\phi, P} = 0.9$: the specification is not robustly satisfied. As expected, the property holds for the reference parameter values $\mathbf{p}^*$ (i.e. $sd(T_{\mathbf{p}^*}, \phi) = 1$), and consequently the robustness and absolute robustness are equal ($\hat{R}_{\phi, P}^* = 0.9$). The distribution of the satisfaction degree is represented in Fig 8(d), showing that although the majority of timed traces satisfies the specification, this is not always the case.

As said earlier, for most synthetic biology applications, a more robust timer would be needed. Can we find other parameters so as to improve the robustness of the system with respect to

similar parameter perturbations? To do so, we use the state-of-the-art non-linear optimization tool CMAES that uses a covariance matrix adaptation evolution strategy [23], with the *robustness as optimization criteria* (i.e. as fitness function).

We found the following parameter values:

$$\tilde{\mathbf{p}} = (\kappa^0, \kappa, \gamma, \theta, \eta) = ((2.30, 4.20, 3.78), (1234.5, 514.5, 5174.3), (0.024, 0.015, 0.012), (1647.2, 662.8, 936.4), (4.8, 3.7, 8.4))$$

The comparison between original parameters $\mathbf{p}^*$ and so-called optimized parameters $\tilde{\mathbf{p}}$ reveals that the EYFP production rate and the Hill coefficients η have been significantly increased. Given that one wants to ensure a fast transition between the low and high states, these parameters were obvious targets for optimizations. Because tuning Hill coefficients is experimentally difficult, we looked for and found parameters with unchanged Hill coefficients that ensure a robust well timed behavior.

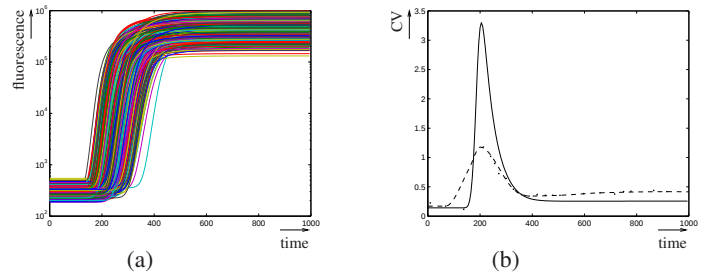


Fig. 9. (a) Numerical simulations of the ODE model with log-normally distributed parameters with mean $\tilde{\mathbf{p}}$. (b) Temporal evolution of the coefficient of variation of the fluorescence following addition of aTc. Crosses and solid line represent coefficient of variations obtained from experimental data in [24] and from 5000 numerical simulations of the ODE model with log-normally distributed parameters with mean $\tilde{\mathbf{p}}$, respectively.

Numerical integrations illustrate that the expected behavior is indeed more robustly obtained (compare Fig 8(c) and Fig 9(a)). Interestingly, the coefficient of variation suggests that cell-to-cell variability will be significantly decreased when the time constraints hold (for $time < 150$) and is significantly increased otherwise (for $150 < time < 450$, see Fig 9(b)). It would be interesting to study whether this feature appears systematically for parameter variations improving the robustness of the desired behavior. This could reveal trade-offs between robustness and fragility [26].

4.5 Parameter influence on robust behavior

To obtain a more comprehensive picture of the variations of the robustness of the expected behavior, we sample the parameter space for large parameter variations, and for each parameter, we compute the robustness.

More precisely, we consider grids on the parameter space centered on the reference parameter values $\mathbf{p}^*$ and corresponding

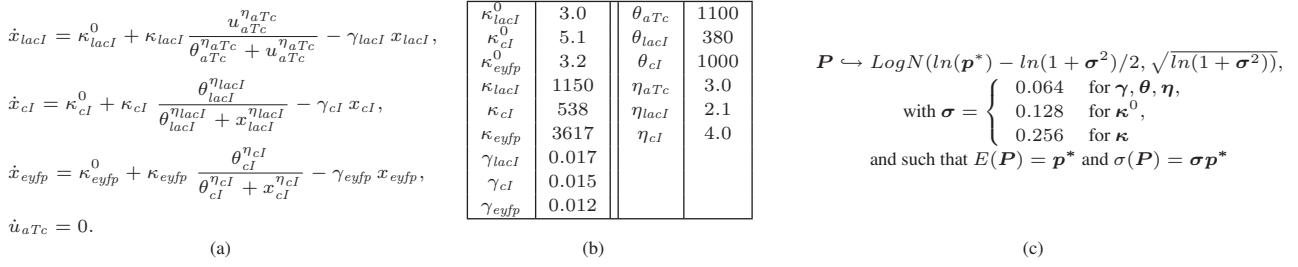


Fig. 7. (a) ODE model of the transcriptional cascade. The concentrations of protein LacI, CI, EYFP and of aTc are denoted by x_{lacI} , x_{cI} , x_{eyfp} , and u_{aTc} , respectively. The concentration of the constitutively expressed protein TetR is assumed constant. (b) Reference parameter values $\mathbf{p}^*$ and (c) parameter distributions modeling system's variability. σ is a noise intensity parameter vector.

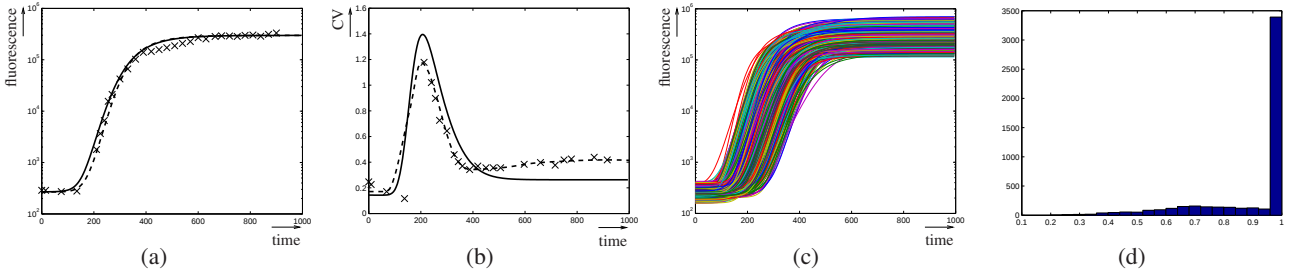


Fig. 8. (a) Temporal evolution of the fluorescence following addition of aTc. Crosses, dotted line and solid line represent experimental data from [24], predictions obtained using the ODE model with reference parameters $\mathbf{p}^*$, and average of 5000 numerical simulations of the ODE model with log-normally distributed parameters, respectively. (b) Temporal evolution of the coefficient of variation of the fluorescence following addition of aTc. Crosses and solid line represent coefficient of variations obtained from experimental data in [24] and from 5000 numerical simulations of the ODE model with log-normally distributed parameters with mean $\mathbf{p}^*$, respectively. (c) Numerical simulations of the ODE model with log-normally distributed parameters with mean $\mathbf{p}^*$. (d) Distribution of satisfaction degrees for 5000 numerical traces of the perturbed transcriptional cascade model. The corresponding robustness is $\hat{R}_{\phi, P} = 0.9$.

to ± 10 fold parameter variations of either 2 parameters (κ_{eyfp} and γ ; 2D grids) or 8 parameters (κ^0, κ, γ , and u_{aTc} ; 8D grids). Then, for each grid point -taken as reference value for relative robustness computations- we estimate the robustness of the network behavior when all 16 parameters vary. Note that we consider the initial aTc concentration as a parameter. The γ parameter correspond to a scaling factor of all degradation parameters γ_{lacI} , γ_{cI} , and γ_{eyfp} , with $\gamma^* = 1$. It is used to assess the impact of growth rate variations, affecting similarly all protein dilution rates, and consequently, all degradation rates.

For the 2D grid, results can be visually displayed. In Figure 10, the satisfaction degree, the robustness and the relative robustness are represented in the (κ_{eyfp}, γ) parameter space. It appears that the constraints on γ are much tighter than the constraints on κ_{eyfp} . Both for the satisfaction degree and for the robustness, γ has to remain in a narrow interval, whereas κ_{eyfp} simply has to exceed some value. This result can be explained by the fact that high production rates of the fluorescent protein helps the system to have a fast and marked response, whereas variations in protein degradation rates γ has subtle effects on the behavior, since it lowers the concentration of the fluorescent protein and of its repressor. It seems that the nominal behavior, and even more the average behavior, are rather fragile to growth rate variations.

The robustness landscape appears like a blurred version of the satisfaction degree landscape. This corresponds to the fact that parameter variations corresponding to cell to cell variability used for computing the robustness are generally smaller than parameter perturbations considered when exploring the parameter space. However, one should also stress that the robustness takes into

account parameter variations in all dimensions and with particular distributions (here log-normal, with various noise intensities σ). Thus Fig 10(b) is not merely a blurred version of Fig 10(a).

In Fig 10(c), it appears that the relative robustness that quantifies how different the average behavior is from the nominal one, efficiently identifies regions where the satisfaction degree changes significantly. In the context of system design, this information is of great interest. This could be compared with the sensitivity of satisfaction degree with respect to parameter perturbations. However, contrary to the sensitivities, the relative robustness takes into account a given perturbation model.

The preceding analysis is naturally not possible when considering parameter variations in higher dimensions. To carry the analysis on 8D grids, we use a variance-based global sensitivity method [33]. When a measure (in our case the robustness) is affected by variations of several parameters, one can statistically assess the importance of the variations of each parameter by computing its *sensitivity index*:

$$S_i = \frac{\text{Var}(E(R | P_i))}{\text{Var}(R)} \in [0, 1],$$

These sensitivity indices and higher order sensitivity indices quantify how the variance of a parameter P_i or a group of parameters contributes to the variance of R .

We consider 8D grids defined as follows. Each grid is defined by 3 parameter values (p_i^1, p_i^2, p_i^3) in each dimension. These values -or more precisely their log- are obtained by dividing evenly the parameter domain $[\ln(p_i/10), \ln(10p_i)]$ in 3 subintervals and by choosing randomly a value in each subinterval.

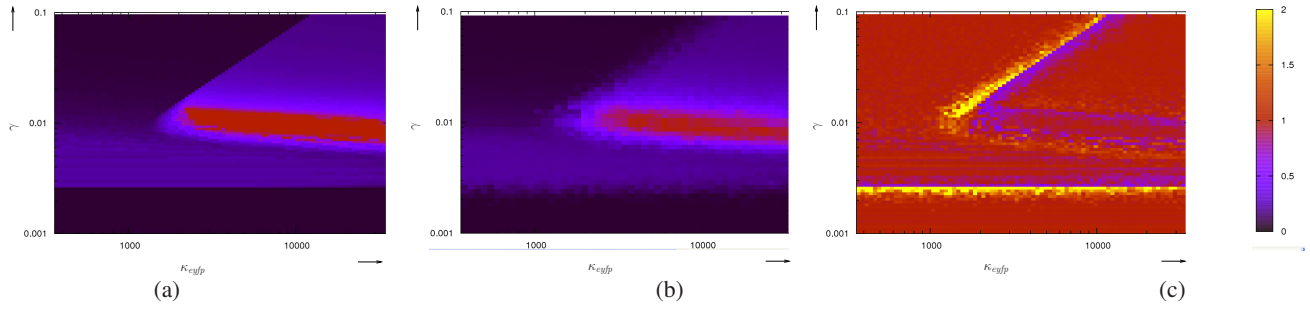


Fig. 10. (a) Satisfaction degree, (b) robustness and (c) relative robustness represented in the (κ_{eyfp}, γ) parameter space.

The first and most significant second order sensitivity indices are given in table 1. They correspond to average values obtained on 10 similarly defined grids.

S_γ	20.2 %	$S_{\kappa_{eyfp}, \gamma}$	8.7 %
$S_{\kappa_{eyfp}}$	7.4 %	$S_{\kappa_{cl}, \gamma}$	6.2 %
$S_{\kappa_{cl}}$	6.1 %	$S_{\kappa_{cl}^0, \gamma}$	5.0 %
$S_{\kappa_{lacI}^0}$	3.3 %	$S_{\kappa_{cl}^0, \kappa_{eyfp}}$	2.8 %
$S_{\kappa_{cl}^0}$	2.0 %	$S_{\kappa_{cl}, \kappa_{eyfp}}$	1.8 %
$S_{\kappa_{lacI}}$	1.5 %	$S_{\kappa_{eyfp}, \gamma}$	1.5 %
$S_{\kappa_{eyfp}^0}$	0.9 %	$S_{\kappa_{cl}^0, \kappa_{cl}}$	1.1 %
S_{uaTc}	0.4 %	$S_{\kappa_{cl}^0, \kappa_{lacI}}$	0.5 %
total first order	40.7 %	total second order	31.2 %

Table 1. First and most significant second order sensitivity indices defined for the robustness with respect to large parameter variations and computed on 8D grids.

The analysis of the first order sensitivity indices corroborates our previous finding that γ variations have a very strong impact on the robust behavior of the cascade. The variations of this parameter alone are responsible for 20 % of the robustness variations. In contrast, aTc variations seem to have a very low impact on the cascade behavior. Although it might seem in contradiction with the ultrasensitivity of the input/output behavior [24], it simply indicates that the aTc concentrations used for inducing the cascade are high enough to make the network insensitive to even large aTc variations.

A surprising outcome of this analysis is the very different importance of variation in the basal and regulated EYFP production rates, κ_{eyfp}^0 and κ_{eyfp} (Table 1). Given that the specification imposes similar constraints on the “low” and “high” EYFP levels, and that these levels are under mild approximations proportional to the ratio $\kappa_{eyfp}^0/\gamma_{eyfp}$ and $\kappa_{eyfp}/\gamma_{eyfp}$, respectively, one could have expected similar sensitivity indices for κ_{eyfp}^0 and κ_{eyfp} . In fact the low EYFP levels also depends -and in a non-linear way- on the steady state value of CI, itself proportional to κ_{cl}/γ_{cl} . Because κ_{cl} variations have strong effects on robust behavior of the cascade, our results suggest that when uninduced, the basal production of EYFP is due to an incomplete repression of the promoter by CI, explaining the high effect of κ_{cl} variations, rather than a constitutive leakage of the promoter, explaining the low effect of κ_{eyfp}^0 variations. This hypothesis is also consistent with the second order sensitivity indices we found: $S_{\kappa_{cl}, \gamma} > S_{\kappa_{eyfp}^0, \gamma}$.

The analysis of second order sensitivity indices indicates that joint variations of production and degradation rates play a significant role in robustness variations. This comes with no surprise, since as said

earlier, the ratios κ_i/γ_i largely determine the steady state levels of the proteins.

5 DISCUSSION

We have presented a general and computational framework for the definition of the robustness of biological functions with respect to a set of perturbations. This framework is general because it applies (i) to any biological function expressible in the temporal logic LTL, an expressive language for specifying dynamical behaviors widely-used in computer science and engineering, and (ii) to any perturbation set, provided that the behaviors of the perturbed system can be obtained as numerical timed traces, for example by numerical integration of ODEs. In this setting, the computation of robustness can be fully automated. It is implemented in the free software BIOCHAM [7].

When formalizing the robustness notion, we found that several notions of robustness can be proposed. In particular, one can distinguish the absolute robustness, quantifying the average performance of a perturbed system, from the relative robustness, quantifying the performance degradation (or improvement) due to perturbations.

To illustrate the applicability of our approach and demonstrate its biological relevance, we considered the possibility to improve the robustness of the timed response of a transcriptional cascade to an addition of inducer. The significant cell-to-cell variability makes it difficult to use this system as a reliable biological timer for synthetic biology applications. We found parameter modifications for which a desired timed behavior is robustly obtained. Moreover, we explored the impact of possibly-large parameter variations on the robustness of the desired behavior. Using global sensitivity analysis, we obtained several interesting results that could potentially help for the optimization of the system.

Central to our approach is the notion of *satisfaction degree* of temporal logic formulae. In systems and synthetic biology, many computational approaches use a rather simple measure of the performance of the system, either for parameter searching, robustness computation or local and global sensitivity analysis. Finding a relevant measure of the system performance limits the applicability of the above-mentioned approaches. Examples of such measures are the gain of a response, and the perturbation of a steady state or of the period of oscillations [21, 20, 22]. In contrast, using the satisfaction degree as a performance measure allows us to take advantage of the expressivity of temporal logics and consequently to significantly broaden the applicability of these techniques. In [32], we showed that using the satisfaction

degree, one can efficiently find parameter values for which complex dynamical behaviors are observed. In this article, we show how using the same notion, one can define and estimate the robustness of any dynamical behavior expressible in LTL with respect to a set of perturbations, and how one can apply global sensibility analysis to find the effect of parameter variations on the robustness of any LTL specification of an expected behavior. Other approaches have been proposed that use temporal logic to define robustness of biological systems [34, 5]. However, these approaches use a Boolean interpretation of temporal logic that is not well-adapted to defining a quantitative notion of robustness.

The relations between robustness and evolvability, and between robustness and modularity have been extensively studied in systems biology (*i.e.* [11, 26]). In synthetic biology however not much work has focused on robustness analysis. For obvious reasons, achieving a robust behavior despite cell variability and environmental fluctuations is a central issue in synthetic biology. Because large synthetic networks are very likely to be modular [10, 30], one could envision an approach in which each module is designed to robustly present a given behavior such that one has some guarantee that when included in a more complex system the module still functions as expected. In this context, input/output robustness [35] and insulation [38] are notions of particular interest.

ACKNOWLEDGEMENT

We would like to thank Ron Weiss, Priscilla Purnick, and other members of the Weiss lab for fruitful discussions and for providing experimental data. We gracefully acknowledge partial support from the European FP7 Strep project Tempo, the INRIA AE Colage and the INRA AgroBi project Insight for this research.

REFERENCES

- [1] M. Antonioti, A. Policriti, N. Ugel, and B. Mishra. Model building and model checking for biochemical processes. *Cell Biochem. Biophys.*, 38:271–286, 2003.
- [2] R. Bagnara, P.M. Hill, and E. Zaffanella. The Parma Polyhedra Library: toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Sci. Comput. Program.*, 72(1–2):3–21, 2008.
- [3] N. Barkal and S. Leibler. Robustness in simple biochemical networks. *Nature*, 387(6636):913–916, 1997.
- [4] G. Batt, D. Ropers, H. de Jong, J. Geiselman, 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*, 21(Suppl.1):i19–i28, 2005.
- [5] G. Batt, B. Yordanov, R. Weiss, and C. Belta. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics*, 23(18):2415–2422, 2007.
- [6] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: Extending Thomas’ asynchronous logical approach with temporal logic. *J. Theor. Biol.*, 229(3):339–347, 2004.
- [7] L. Calzone, F. Fages, and S. Soliman. BIOCHAM: An environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.
- [8] N. Chabrier and F. Fages. Symbolic model checking of biochemical networks. In C. Priami, editor, *Proc. of Computational Methods in Systems Biology (CMSB’03)*, vol. 2602 of *LNCS*, pp. 149–162. Springer, 2003.
- [9] M. Chaves, A. Sengupta, and E.D. Sontag. Geometry and topology of parameter space: investigating measures of robustness in regulatory networks. *J. Math. Biol.*, 104(34):13591–13596, 2007.
- [10] J.W. Chin. Modular approaches to expanding the functions of living matter. *Nature Chemical Biology*, 2(6):304–311, 2006.
- [11] S. Ciliberti, O. C. Martin, and A. Wagner. Innovation and robustness in complex regulatory gene networks. *PNAS*, 104(34):13591–13596, 2007.
- [12] E.H. Davidson and M.S. Levine. Properties of developmental gene regulatory networks. *PNAS*, 105(51):20063–20066, 2008.
- [13] E.H. Davidson, D.R. McClay, and L.E. Hood. Regulatory gene networks and the properties of the developmental process. *PNAS*, 100(4):1475–1480, 2003.
- [14] C. Eisner, D. Fisman, J. Havlicek, Y. Lustig, A. McIsaac, and D. van Campenhout. Reasoning with temporal logic on truncated paths. In *Proc. of 15th Computer-Aided Verification conference (CAV’03)*, vol. 2725 of *LNCS*, pp. 27–39, 2003.
- [15] H. El-Samad, H. Kurata, J.C. Doyle, C.A. Gross, and M. Khammash. Surviving heat shock: Control strategies for robustness and performance. *PNAS*, 102(8):2736–2741, 2005.
- [16] E.A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, vol. B, pp. 995–1072. MIT Press, 1990.
- [17] F. Fages and A. Rizk. On temporal logic constraint solving for the analysis of numerical data time series. *Theoretical Computer Science*, 408(1):55–65, 2008.
- [18] G.E. Fainekos and G.J. Pappas. Robustness of temporal logic specifications. In K. Havelund, M. Núñez, G. Rosu, and B. Wolff, editors, *Proc. of the International Workshop on Formal Approaches to Software Testing and Runtime Verification, FATES/RV’06*, vol. 4262 of *LNCS*, pp. 178–192. Springer, 2006.
- [19] G.E. Fainekos and G.J. Pappas. Robust sampling for MITL specifications. In J.-F. Raskin and P.S. Thiagarajan, editors, *Proc. of the Fifth International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS’07*, vol. 4763 of *LNCS*, pp. 147–162. Springer, 2007.
- [20] M.-A. Felix and A. Wagner. Robustness and evolution: concepts, insights and challenges from a developmental model system. *Heredity*, 100(2):132–140, 2008.
- [21] X.-J. Feng, S. Hooshangi, D. Chen, G. Li, R. Weiss, and H. Rabitz. Optimizing genetic circuits by global sensitivity analysis. *Biophys. J.*, 87(2):2195–2202, 2004.
- [22] D. Gonze, J. Halloy, and A. Goldbeter. Robustness of circadian rhythms with respect to molecular noise. *PNAS*, 99(2):673–678, 2002.
- [23] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
- [24] S. Hooshangi, S. Thiberge, and R. Weiss. Ultrasensitivity and noise propagation in a synthetic transcriptional cascade. *PNAS*, 102(10):3581–3586, 2005.
- [25] N. T. Ingolia. Topology and robustness in the drosophila segment polarity network. *PLoS Biology*, 2(6):e123, 2004.
- [26] H. Kitano. Biological robustness. *Nature Review Genetics*, 11(5):826–837, 2004.
- [27] H. Kitano. Towards a theory of biological robustness. *Mol. Syst. Biol.*, 3:137, 2007.
- [28] O. Maler, D. Nickovic, and A. Pnueli. Checking temporal properties of discrete, timed and continuous behaviors. In A. Avron, N. Dershowitz, and A. Rabinovich, editors, *Pillars of Computer Science*, vol. 4800 of *LNCS*, pp. 475–505, 2008.
- [29] T. Manninen, M.-L. Linne, and K. Ruohonen. Developing Itô stochastic differential equation models for neuronal signal transduction pathways. *Comp. Biol. Chem.*, 30(4):280–291, 2006.
- [30] R. McDaniel and R. Weiss. Advances in synthetic biology: On the path from prototypes to applications. *Current Opinion in Biotechnology*, 16:476–483, 2005.
- [31] D. Nickovic and O. Maler. AMT: a property-based monitoring tool for analog systems. In J.-F. Raskin and P.S. Thiagarajan, editors, *Proc. of the Fifth International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS’07*, vol. 4763 of *LNCS*, pp. 304–319. Springer, 2007.
- [32] A. Rizk, G. Batt, F. Fages, and S. Soliman. On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology. In M. Heiner and A. Uhrmacher, editors, *Proc. of the Fourth International Conference on Computational Methods in Systems Biology (CMSB’08)*, vol. 5307 of *LNCS*, pp. 251–268. Springer, 2008.
- [33] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto. *Sensitivity analysis in practice: A guide to assessing scientific models*. Wiley Press, 2004.
- [34] X. Shen, J. Collier, D. Dill, L. Shapiro, M. Horowitz, and H. H. McAdams. Architecture and inherent robustness of a bacterial cell-cycle control system. *PNAS*, 105(32):11340–11345, 2008.
- [35] G. Shinar, R. Milo, M.R. Martinez, and U. Alon. Input-output robustness in simple bacterial signaling systems. *PNAS*, 104(50):19931–19935, 2007.
- [36] J. Stelling, E.D. Gilles, and F.J. Doyle III. Robustness properties of circadian clock architectures. *PNAS*, 101(36):13210–13215, 2004.
- [37] J. Stelling, U. Sauer, Z. Szallasi, F.J. Doyle III, and J. Doyle. Robustness of cellular functions. *Cell*, 118(6):675–685, 2004.
- [38] D. Del Vecchio, A. J. Ninfa, and E. D. Sontag. Modular cell biology: retroactivity and insulation. *Mol. Syst. Biol.*, 4:161, 2008.
- [39] G. von Dassow, E. Meir, E.M. Munro, and G.M. Odell. The segment polarity network is a robust developmental module. *Nature*, 406(6792):188–192, 2000.
- [40] T. Yi, Y. Huang, M. I. Simon, and J. Doyle. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *PNAS*, 97(9):4649–4653, 2000.