EPTCS 125

Proceedings of the

Second International Workshop on

Hybrid Systems and Biology

Taormina, Italy, 2nd September 2013

Edited by: Thao Dang and Carla Piazza

Published: 27th August 2013
DOI: 10.4204/EPTCS.125
ISSN: 2075-2180
Open Publishing Association

Table of Contents

Preface

Thao Dang and Carla Piazza

This volume contains the proceedings of the Second International Workshop Hybrid Systems and Biology (HSB 2013) held in Taormina (Italy), on September 2th, 2013. The workshop is affiliated to the 12th European Conference on Artificial Life (ECAL 2013).

Systems biology aims at providing a system-level understanding of biological systems by unveiling their structure, dynamics and control methods. Due to the intrinsic multi-scale nature of these systems in space, in organization levels and in time, it is extremely difficult to model them in a uniform way, e.g., by means of differential equations or discrete stochastic processes. Furthermore, such models are often not easily amenable to formal analysis, and their simulations at the organ or even at the cell levels are frequently impractical. Indeed, an important open problem is finding appropriate computational models that scale well for both simulation and formal analysis of biological processes. Hybrid modeling techniques, combining discrete and continuous processes, are gaining more and more attention in such a context, and they have been successfully applied to capture the behavior of many biological complex systems, ranging from genetic networks, biochemical reactions, signaling pathways, cardiac tissues electro-physiology, and tumor genesis. This workshop aims at bringing together researchers in computer science, mathematics, and life sciences, interested in the opportunities and the challenges of hybrid modeling applied to systems biology.

The workshop programme included the keynote presentation of Alessandro Astolfi (Imperial College of London, UK) on Immune response enhancement via hybrid control. Furthermore, 8 papers were selected out of 13 submissions by the Program Committee of HSB 2013. The papers in this volume address the hybrid modeling of a number important biological processes (iron homeostasis network, mammalian cell cycle, vascular endothelial growth factor (VEGF), genetic regulatory network in mammalian sclera) and, the formalisms and techniques for specifying and validating properties of biological systems (such as, robustness, oscillations).

The Program Committee of HSB 2013 involved:

- Ezio Bartocci, TU Wien, Austria
- Calin Belta, Boston University, USA
- Luca Bortolussi, Univerity of Trieste, Italy
- Thao Dang, VERIMAG/CNRS Grenoble, France (co-chair)
- Alexandre Donzé, University of California Berkley, USA
- James R. Faeder, University of Pittsburgh, USA
- François Fages, INRIA Rocquencourt, France
- Etienne Farcot, University of Nottingham, UK
- Gregor Goessler, INRIA Grenoble, France
- Adam Halasz, West Virginia University, USA
- Jane Hillston, University of Edinburgh, UK
- Agung Julius, Rensselaer Polytechnic Institute, USA

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 1–2, doi:10.4204/EPTCS.125.0

- Heinz Koeppl, Swiss Federal Institute of Technology Zurich, Switzerland
- Pietro Liò, University of Cambridge, UK
- Oded Maler, VERIMAG/CNRS Grenoble, France
- Giancarlo Mauri, University of Milano-Bicocca, Italy
- Bud Mishra, New York University, USA
- Chris Myers, University of Utah, USA
- Carla Piazza, University of Udine, Italy (co-chair)
- Ovidiu Radulescu, University of Montpellier 2, France
- Scott A. Smolka, University of Stony Brook, USA
- Gouhei Tanaka, University of Tokyo, Japan
- Ashish Tiwari, SRI International Menlo Park, USA
- Dongming Wang, UPMC-CNRS Paris, France
- Verena Wolf, Saarland University, Germany
- Paolo Zuliani, Newcastle University, UK

We would like to thank the ECAL organisers, all the authors, the invited speaker, the Programme Committee and the external referees for their valuable contributions. Special thanks go to the organizers of HSB 2012, Ezio Bartocci and Luca Bortolussi, who helped us during all the organizaion steps, and to GNCS (Gruppo Nazionale per il Calcolo Scientifico) at INdAM for financially supporting our workshop.
August 2013

Thao Dang and Carla Piazza

Workshop co-chairs

On the Robustness of Temporal Properties for Stochastic Models

Ezio Bartocci
TU Wien, Austria
ezio.bartocci@tuwien.ac.at

Luca Bortolussi
DMG, University of Trieste, Italy
CNR/ISTI, Pisa, Italy
luca@dmi.units.it

Laura Nenzi
IMT, Lucca, Italy
laura.nenzi@imtlucca.it

Guido Sanguinetti
School of Informatics, University of Edinburgh
SynthSys, Centre for Synthetic and Systems Biology, University of Edinburgh
g.sanguinetti@ed.ac.uk

Stochastic models such as Continuous-Time Markov Chains (CTMC) and Stochastic Hybrid Automata (SHA) are powerful formalisms to model and to reason about the dynamics of biological systems, due to their ability to capture the stochasticity inherent in biological processes. A classical question in formal modelling with clear relevance to biological modelling is the model checking problem, i.e. calculate the probability that a behaviour, expressed for instance in terms of a certain temporal logic formula, may occur in a given stochastic process. However, one may not only be interested in the notion of satisfiability, but also in the capacity of a system to mantain a particular emergent behaviour unaffected by the perturbations, caused e.g. from extrinsic noise, or by possible small changes in the model parameters. To address this issue, researchers from the verification community have recently proposed several notions of *robustness* for temporal logic providing suitable definitions of distance between a trajectory of a (deterministic) dynamical system and the boundaries of the set of trajectories satisfying the property of interest. The contributions of this paper are twofold. First, we extend the notion of robustness to stochastic systems, showing that this naturally leads to a distribution of robustness scores. By discussing two examples, we show how to approximate the distribution of the robustness score and its key indicators: the *average robustness* and the *conditional average robustness*. Secondly, we show how to combine these indicators with the satisfaction probability to address the *system design problem*, where the goal is to optimize some control parameters of a stochastic model in order to best maximize robustness of the desired specifications.

1 Introduction

Biological systems at the single cell level are inherently stochastic. Molecules inside the cells perform random movements (*random walk*) and the reactions among them may occur when the probability of collision is high enough. The number of molecules of each species at each time point is therefore a random process: assuming instantaneous reactions, this process can be modelled as a Markovian (i.e. memoriless) discrete state, continuous time process. When the number of molecules of each species involved is large, so that many reactions happen in any small interval of time, stochastic effects can be neglected. However, if the concentration of the molecules (of at least some of the species) is low the stochasticity plays an important role and must be taken into account. For this reason, stochastic models such as Continuous-Time Markov Chains (CTMC) [24] and Stochastic Hybrid Automata [14] are particularly powerful and suitable formalisms to model and to reason about biological systems defined as stochastic systems over time.

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 3–19, doi:10.4204/EPTCS.125.1

A classical question in formal modelling is to calculate the probability that a behaviour, expressed in terms of a certain temporal logic formula, may occur in a given stochastic process, with specified parameters. *Probabilistic Model Checking* [3, 4] (PMC) is a well-established verification technique that provides a quantitative answer to such a question. The algorithm used to calculate this probability [34] produces the exact solution, as it operates directly on the structure of the Markov chain. Despite the success and the importance of PMC, this technique suffers some computational limitations, either due to state space explosion or to the difficulty (impossibility) in checking analytically formulae in specific logics, like Metric Temporal Logic (MTL) [4, 16]. Furthermore, PMC provides only a quantitative measure of the *satisfability* (yes/no answer) of a temporal logic specification (i.e., the probability of the property being true).

However, especially when we deal with stochastic models, the notion of satisfability may be not enough to determine the capacity of a system to mantain a particular emergent behaviour unaffected by the uncertainty of the perturbations due to its stochastic nature or by possible small changes in the model parameters. A similar issue also arises when considering the satisfability of a property by deterministic dynamical systems which may be subject to extrinsic noise or uncertainty in the parameter. To address this question in the deterministic case, researchers from the verification community have proposed several notions of *temporal logic based robustness* [23, 27, 41], providing suitable definitions of distance between a trajectory of a system and the behavioural property of interest, expressed in terms of a temporal logic formula. These effectively endow the logic of interest with quantitative semantics, allowing us to capture not only whether a property is satisfied but also *how much* it is satisfied. A similar notion of robustness for stochastic models would clearly be desirable but, to our knowledge, has not been formalised yet.

The contributions of this paper are twofold. First,we provide a simulation-based method to define a notion of robust satisfability in stochastic models. Simulation-based approaches, such as statistical model checking [43], can be be used to estimate for a stochastic model the *robust satisfability distribution* for a given temporal logic formula, with a guarantee of asymptotic correctness. This distribution is the key to understand how the behaviour specified by the logic temporal formula is unaffected by the stochasticity of the system. In particular, in this paper we consider two important indicators of this distribution: the *robustness average* and the *conditional robustness average* on a formula being true or false. We discuss how to compute the robust satisfability distribution and its indicators on two biological examples. Second, we show how to combine these indicators with the satisfaction probability to address the *system design problem*, where the goal is to optimize (few) control parameters of a stochastic model in order to best maximize these three indicators. The proposed approach takes advantage of Gaussian Process Upper Confidence Bound (GP-UCB) algorithm introduced in [42].

The paper is structured as follows: in Section 2 we introduce the background material. In Section 3 we discuss the robustness of stochastic models using the quantitative semantics of the Signal Temporal Logic (STL). In Section 4 we present some experimental results for the robustness of STL formulae for two stochastic models that we have chosen as our case studies: the Schlögl system and the Reprissilator. In Section 5 we show an application of the robust semantics to the system design problem. The related works and the final discussion are in Section 6.

2 Background

Markov Population Models. The simplest class of stochastic processes we will consider are Continuous Time Markov Chains (CTMC) [24] that describe population processes (PCTMC). A population

process intuitively is a system in which agents or objects of different kinds, and with different internal states, interact together. The classical example are biochemical and genetic networks, but other population processes include ecological systems, computer networks, and social systems.

We will describe PCTMC by the simple formalism of biochemical reaction networks. The state of the system is described by a vector $\mathbf{X} = (X_1, \ldots, X_n)$ of n integer-valued variables X_i, each counting the number of entities of a given class or species. The dynamics of this system is specified by a set of m reactions $\mathscr{R} = \{\eta_1, \ldots, \eta_m\}$, which can be seen as description of events changing the state of the system. Each reaction η_l is of the form

$$r_1 X_{i_1} + \ldots + r_k X_{i_k} \to s_1 X_{j_1} + \ldots + s_h X_{j_h},$$

where X_{i_a} is a reactant and X_{j_b} is a product (they are both variables of $\mathbf{X}$), and r_i, s_j are the stoichiometric coefficients, i.e. the amount of agents/ entities consumed or produced by the reaction. Stoichiometric information of a reaction η_l can be condensed into an update vector $\mathbf{v}_l$, giving the net change in population variables due to η_l: $\mathbf{v}_l = \sum_{b \le h} s_b \mathbf{1}_{j_b} - \sum_{a \le k} r_a \mathbf{1}_{i_a}$, where $\mathbf{1}_j$ equals one in position j and zero elsewhere. Additionally, each reaction η_j has an associated rate function $f_j(\mathbf{X})$ giving the rate of the transition as a function of the global state of the system.

From a set of reactions $\mathscr{R}$ and species $\mathbf{X}$, we can easily derive the formal representation of a CTMC in terms of its infinitesimal generator matrix, see for instance [9]. Here we just recall that the state space of the CTMC is $\mathbb{N}^n$ (or a proper subset, if any conservation law is in force). Such CTMC can be simulated with standard algorithms, like SSA [29].

Fluid Approximation From a Markov population model, we can easily construct an alternative semantics in terms of Ordinary Differential Equations (ODE), assuming variables $\mathbf{X}$ to be continuous and interpreting each rate as a flow, thus obtaining the vector field

$$F(\mathbf{X}) = \sum_{\eta_l \in \mathscr{R}} \mathbf{v}_l f_l(\mathbf{X}),$$

defining the ODE $d/dt\mathbf{X} = F(\mathbf{X})$. This equation, known as fluid approximation, can be shown to be a first order approximation of the average of the CTMC, and, under a suitable rescaling of the variables (dividing by the system size, which for biochemical reactions is just the volume), one can prove convergence of the CTMC to the solution of the ODE (see [9]) as populations and system size go to infinity. Intuitively, this ODE is a good description of the system behaviour when populations are large.

Stochastic Hybrid Automata In many situations, it is not the case that all entities/ species in the model are present in large quantities. In such scenarios, fluid approximation can give poor results, yet dealing with CTMCs can be computationally unfeasible. An example are genetic regulatory networks, in which genes are modelled explicitly as a finite state machine [12]. In these cases, a better strategy is that of approximating continuously only some variables, keeping discrete the others. This reflects in the dynamics: some reactions will be converted into flows (generally those modifying only continuous variables), while the others will remain stochastic discrete events. This gives rise to a model that can be expressed in terms of a class of Stochastic Hybrid Automata (SHA, [12]) known as Piecewise-Deterministic Markov Processes [17]. Alternatives assuming a stochastic continuous dynamics have also been considered [37, 38]. More specifically, the SHA so obtained have discrete modes identified by the value of discrete variables. In between discrete transitions, the system evolves following the solution of the differential equation, whose vector field is mode-dependent (via the value of discrete variables). Discrete jumps happen at

exponentially random distributed times, at a non-constant rate that can depend on the continuous variables. After each jump, the value of discrete variables can change. Also continuous variables can be updated, even if we do not consider this possibility in this paper, see [12] for further details. Similarly to the fluid approximation case, we can see SHA models as the limit of CTMC, taking to the limit only the populations corresponding to continuous variables (under a suitable scaling of rates, see [8] for further details).

Signal Temporal Logic. Temporal logic [39] provides a very elegant framework to specify in a compact and formal way an emergent behaviour in terms of *time-dependent* events. Among the myriads of temporal logic extensions available, Signal Temporal Logic [36] (STL) is very suitable to characterize behavioural patterns in time series of real values generated during the simulation of a dynamical system. STL extends the dense-time semantics of Metric Interval Temporal Logic [1] (MITL), with a set of parametrized numerical predicates playing the role of atomic propositions. STL provides two different semantics: a boolean semantics that returns yes/no depending if the observed trace satisfies or not the STL specification and a quantitative semantics that also returns measure of robustness of the specification. Recently, Donzé et. al [22] proposed a very efficient monitoring algorithm for STL robustness, now implemented in the Breach [19] tool. The combination of robustness and sensitivity-based analysis of STL formulae have been successfully applied in several domains ranging from analog circuits [31] to systems biology [20, 21], to study the parameter space and also to refine the uncertainty of the parameter sets. In the following we recall [23] the syntax and the quantitative semantics of STL that will be used in the rest of the paper. The boolean semantics can be inferred using the sign of the quantitative result (positive for true and negative for false).

Definition 1 (STL syntax) *The syntax of the STL is given by*

$$\varphi := \top \mid \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \,\mathscr{U}_{[a,b]}\, \varphi_2,$$

where $\top$ is a true formula, conjunction and negation are the standard boolean connectives, $[a,b]$ is a dense-time interval with $a < b$ and $\mathscr{U}_{[a,b]}$ is the until operator.

The atomic predicate $\mu : \mathbb{R}^n \to \mathbb{B}$ is defined as $\mu(\mathbf{x}) := (y(\mathbf{x}) \geqslant 0)$, where $\mathbf{x}[t] = (x_1[t], \ldots, x_n[t])$, $t \in \mathbb{R}_{\geqslant 0}$, $x_i \in \mathbb{R}$, is the primary signal, and $y : \mathbb{R}^n \to \mathbb{R}$ is a real-valued function known as the secondary signal.

The (bounded) until operator $\varphi_1 \,\mathscr{U}_{[a,b]}\, \varphi_2$ requires φ_1 to hold from now until, in a time between a and b time units, φ_2 becomes true. The *eventually* operator $F_{[a,b]}$ and the *always* operator $G_{[a,b]}$ can be defined as usual: $F_{[a,b]}\varphi := \top \mathscr{U}_{[a,b)} \varphi$, $G_{[a,b]}\varphi := \neg F_{[a,b]} \neg\varphi$.

Definition 2 (STL Quantitative Semantics for space robustness)

$$\begin{aligned}
\rho(\mu, \mathbf{x}, t) &= y(\mathbf{x}[t]) \qquad \textit{where } \mu \equiv y(\mathbf{x}[t]) \geqslant 0 \\
\rho(\neg\varphi, \mathbf{x}, t) &= -\rho(\varphi, \mathbf{x}, t) \\
\rho(\varphi_1 \wedge \varphi_2, \mathbf{x}, t) &= \min(\rho(\varphi_1, \mathbf{x}, t), \rho(\varphi_2, \mathbf{x}, t)) \\
\rho(\varphi_1 \,\mathscr{U}_{[a,b)} \varphi_2, \mathbf{x}, t) &= \max_{t' \in t + [a,b]} (\min(\rho(\varphi_2, \mathbf{x}, t'), \min_{t'' \in [t,t']} (\rho(\varphi_1, \mathbf{x}, t''))))
\end{aligned}$$

where ρ is the quantitative satisfaction function, returning a real number $\rho(\varphi, \mathbf{x}, t)$ quantifying the degree of satisfaction of the property φ by the signal $\mathbf{x}$ at time t. Moreover, $\rho(\varphi, \mathbf{x}) := \rho(\varphi, \mathbf{x}, 0)$.

We stress here that the choice of the secondary signals $y : \mathbb{R}^n \to \mathbb{R}$ is an integral part of the definition of the STL formula expressing the behaviour of interest. Different choices of secondary signals result in different formulae, hence in different robustness measures. We also remark that the robustness score of Definition 2 has to be interpreted as a weight of how much a given model (with fixed initial conditions and parameters) satisfies an STL behaviour. More precisely, its absolute value can be seen as a distance of the signal $\mathbf{x}$ under consideration from the set of trajectories satisfying/ dissatisfying the formula [27], in STL trajectory are projected with respect to secondary signals y [23]. In this sense, this measure is different from the more common sensitivity-based notions of robustness, like those discussed in [33], measuring the size of a region in the parameter space in which the system behaviour is roughly constant. However, sensitivity analysis and its related techniques can be applied to the robustness score of Definition 2.

3 Robustness of Stochastic Models

Consider a STL formula φ, with predicates interpreted over state variables of a PCTMC model $\mathbf{X}(t)$. The boolean semantics of φ is readily extended to stochastic models as customary, by measuring the probability of the set of trajectories of the CTMC that satisfy the formula:

$$P(\varphi) = \mathbb{P}\{\mathbf{x} \mid \mathbf{x} \models \varphi\}.$$

The rationale behind such definition is that a PCTMC model defines a probability distribution on the space of trajectories, which is usually obtained by applying the cylindric construction [4]. Furthermore, the set of trajectories that satisfy/ falsify a formula is a measurable set, so that we can safely talk about its probability. In the following, we will refer to the space of trajectories as $\mathscr{D}$, and interpret the PCTMC model $\mathbf{X}(t)$ as a random variable $\mathbf{X}$ over $\mathscr{D}$. In order to extend this definition to the robustness score, it is convenient to think of the set of trajectories that satisfy φ as a measurable function $I_\varphi : \mathscr{D} \to \{0,1\}$, such that $I_\varphi(\mathbf{x}) = 1$ if and only if $\mathbf{x} \models \varphi$. Then, we can define the random variable $I_\varphi(\mathbf{X})$ on $\{0,1\}$ induced by the PCTMC $\mathbf{X}$ via I_φ as the Bernoulli random variable which is equal to 1 with probability $P(\varphi)$. We can equivalently write:

$$\mathbb{P}(I_\varphi(\mathbf{X}) = 1) = \mathbb{P}(\{\mathbf{x} \in \mathscr{D} \mid I_\varphi(\mathbf{x}) = 1\}) = \mathbb{P}(I_\varphi^{-1}(1))$$

We can extend the robustness score to PCTMC models in the same way: given a trajectory $\mathbf{x}(t)$, we can compute its robustness score according to Def. 2 and interpret $\rho(\varphi, \mathbf{x}, 0)$ as a function from the trajectories in $\mathscr{D}$ to $\mathbb{R}$. This function is easily seen to be measurable (with respect to the σ-algebra induced from the Skorokhod topology in $\mathscr{D}$), and so it induces a real-valued random variable $R_\varphi(\mathbf{X})$ with probability distribution given by

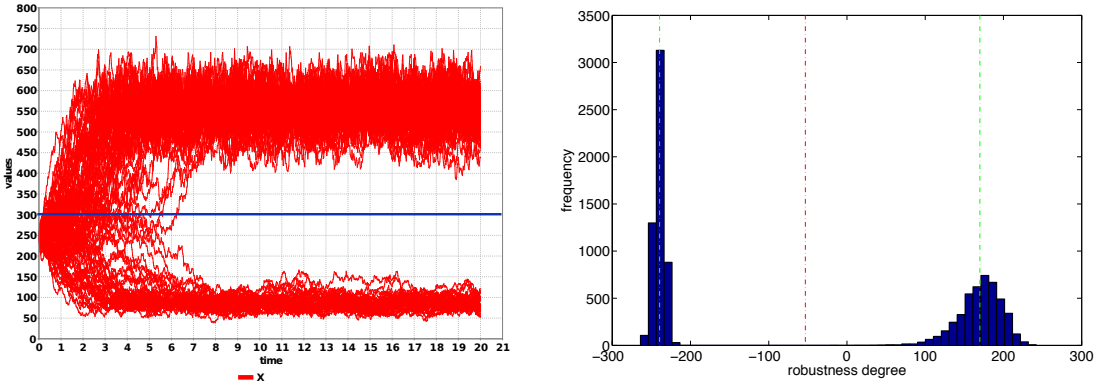
$$\mathbb{P}\left(R_\varphi(\mathbf{X}) \in [a,b]\right) = \mathbb{P}(\mathbf{X} \in \{\mathbf{x} \in \mathscr{D} \mid \rho(\varphi, \mathbf{x}, 0) \in [a,b]\})$$

Staten otherwise, if we apply the definition of robustness to a stochastic model, we obtain a distribution of robustness degrees. This distribution tells us much more than the standard probabilistic semantics, because it tells us "how much" a formula is true.

In particular, in this paper we will be interested in some statistics of this distribution, specifically the average robustness degree, and the average robustness conditional on a formula being true or false. The first quantity gives a measure of how strongly a formula is satisfied on average. The larger this number, the more robust is satisfaction. Most of the times, this number will be correlated with the satisfaction probability, yet we can have a large average satisfaction score even for a small probability

Reaction	rate constant	init pop
$A+2X \to 3X$	$k_1 = 3 \cdot 10^{-7}$	$X(0) = 247$
$3X \to A+2X$	$k_2 = 1 \cdot 10^{-4}$	$A(0) = 10^5$
$B \to X$	$k_3 = 1 \cdot 10^{-3}$	$B(0) = 2 \cdot 10^5$
$X \to B$	$k_4 = 3.5$	

Table 1: Biochemical reactions of the Schlögl model. Parameters are taken from [18].



(a) CTMC simulation

(b) robustness distribution

Figure 1: Simulation of the Schlogl model (100 runs), for parameters as in Table 1. The blue straight line is the value $X = 300$ (left). The robustness distribution of the STL formula 4.1 with $T_1 = 10$ and $T_2 = 15$ time units. It has average robustness -53.15 (vertical red line), conditional robustness 169.89 and -239.52 (vertical green lines), and satisfaction probability 0.4552 (right).

of satisfaction. Better indicators of the intensity of satisfaction and dissatisfaction are the conditional averages, $\mathbb{E}(R_\varphi \mid R_\varphi > 0)$ and $\mathbb{E}(R_\varphi \mid R_\varphi < 0)$. These are related to the average by the equation

$$\mathbb{E}(R_\varphi) = P(\varphi)\mathbb{E}(R_\varphi \mid R_\varphi > 0) + (1 - P(\varphi))\mathbb{E}(R_\varphi \mid R_\varphi < 0)$$

which holds provided $\mathbb{P}(R_\varphi = 0)$ is zero.

One goal of this paper is to investigate to what extent these three synthetic indices are good descriptors of the robustness distribution, and how they can be exploited to do parameter synthesis for PCTMC models.

4 Case Studies

In this section, we investigate experimentally the notion of robust semantics of STL formulae for stochastic models. We will consider two systems: the Schlögl system [30], a simple network of biochemical reactions exhibiting a bistable behaviour, and the Repressilator [25], a synthetic biological clock implemented as a network of gene regulations. More specifically, we consider a CTMC model of the Schlög system and a hybrid model of the Repressilator [10, 11], in order to illustrate the general applicability of the stochastic robust semantics introduced in Section 3.

4.1 Schlögl system

The Schlögl model is a simple biochemical network with four reactions, listed in Table 1. The rates of the reactions are computed according to the mass action principle for stochastic models [29]. Species A and B are considered to be present in large quantities, hence assumed constant. The characteristic of this system is to have, for certain parameter values, like the one shown in Table 1, a bistable behaviour. More specifically, the reaction rate ODE system has two stable steady states, and for this model the trajectories of the stochastic system starting from a fixed initial state x_0 can end up in one attractor or the other. The probability of choosing one stable state or the other depends on the position of x_0 relatively to the basin of attraction of the two equilibria. If we start close to its boundary, the bistable behaviour becomes evident, see Figure 1(a).

We now consider the property of eventually ending up in one basin of attraction, and express it with the STL formula

$$\varphi : \mathbf{F}^{[0,T_1]}\mathbf{G}^{[0,T_2]}(X \geq k_t) \quad k_t = 300 \tag{4.1}$$

stating that the system, after at most T_1 units of time, stabilises to a value which remains above $k_t = 300$ for as long as T_2 units of time. In this formula, the predicate $\mu(X) = X \geq k_t$ corresponds to the linear secondary signal $y(X) = X - k_t$. As can be seen from Figure 1(a), if the model is in the large equilibrium, then this property will be true, and false in the other case.

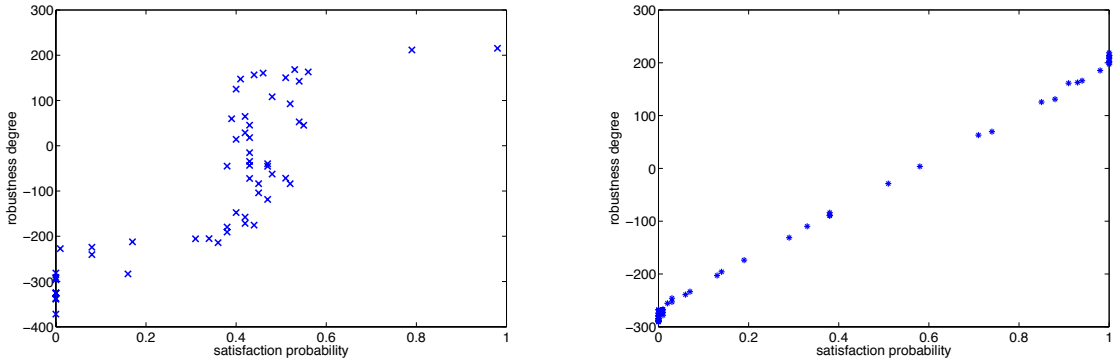
If we estimate the probability of the formula statistically, then we obtain the value $p = 0.4583$ (10000 runs, error ± 0.02 at 95% confidence level). However, this raw number does not tell us anything specific about bistability. A system stabilising just above the threshold 300, such that roughly 55% of its trajectories cross it "frequently", may satisfy the same formula with the same probability. However, the bimodal behaviour becomes evident if we look at the distribution of the robustness degree of the formula, see Figure 1(b) . Hence, the robustness score carries an additional amount of information with respect to the satisfaction probability of a STL formula. We stress that we are not comparing the robustness degree with the probability distribution of the CTMC $X[t]$: both the satisfaction probability of φ and its robustness are (unidimensional) quantities derived from $X[t]$, which are easier to compute and visualise.

In Figure 2, we investigate the behaviour of the average robustness degree, and its relationship with the satisfaction probability. In order to do this, we varied the threshold level k_t in the formula (Figure 2(a)), and the rate constant k_3 (Figure 2(b)), and estimated statistically the average robustness degree and the satisfaction probability from 10000 runs for each parameter combination. As we can see these two quantities are correlated. When we vary the threshold, the correlation between satisfaction probability and robustness score is around 0.8386, while the dependency seems to follow a sigmoid shaped curve. In the second case, instead, the correlation between satisfaction probability and average robustness degree is 0.9718, with an evident linear trend.

Finally, we consider the conditional robustness degrees. For model parameters as in Table 1, the average robustness conditional on Formula (4.1) being true is 169.89, while the robustness conditional on the formula being false is -239.52 (see also Figure 1(b)). These two indicators estimate how robustly the system remains in the basin of attraction of each steady state.

4.2 Repressilator

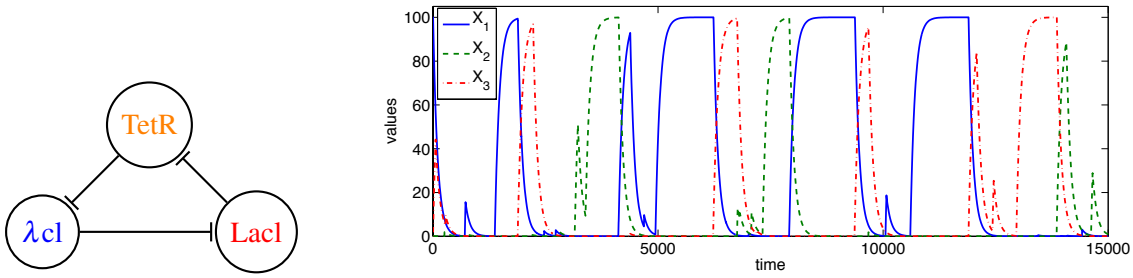
The second case study is a genuine stochastic hybrid model of the Repressilator [25], a synthetic genetic clock composed of three genes expressing three transcription factors repressing each other in a cyclical fashion (see Figure 3). The stochastic hybrid model we consider is taken from [10, 11]. In the model, we lump the transcription and translation in a single event, and model production and degradation of the



(a) varying threshold (b) varying k_3

Figure 2: Satisfaction probability versus average robustness degree for varying (left) threshold k_t in the STL formula (4.1) and (right) parameter k_3. k_3 was varied between 100 and 300 in steps of 10 units, while the threshold was varied between 50 and 600 in steps of 10.

protein as continuous flows. The binding and unbinding of transcription factors from gene promoters, instead, are modelled as discrete and stochastic events. As we can see in Figure 3(b), the model exhibit sustained oscillations, albeit with an irregular period. This happens for parameters giving a strong repression via a low unbinding rate.



(a) Repressilator, gene network (b) Hybrid stochastic simulation

Figure 3: The repressilator (left) is a cyclic negative-feedback loop composed of three repressor genes: TetR, λcl, Lacl. Oscillatory behaviour of the model (right), for model parameters: protein production rate $k_p = 1$, protein degradation rate $k_d = 0.01$, repressor binding rate $k_b = 0.1$, repressor unbinding rate $k_u = 0.001$.

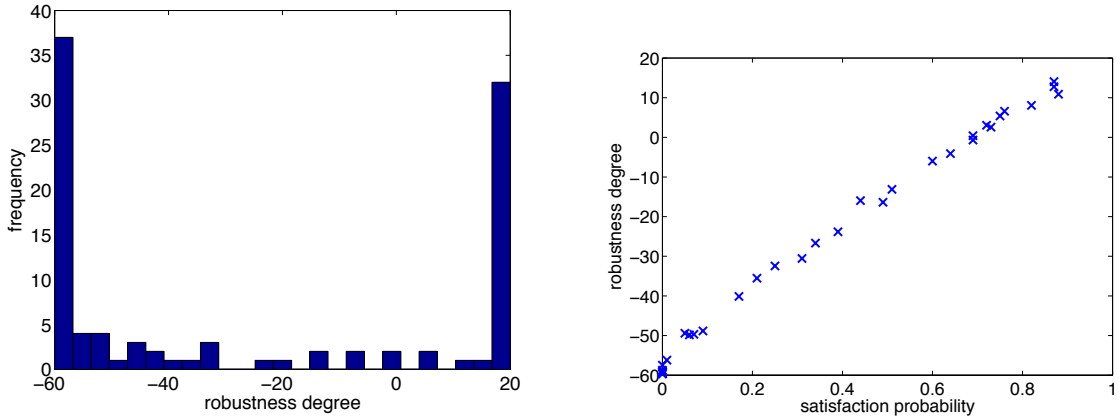
In order to check for the presence of oscillations, we use the STL formula

$$\mathbf{G}^{[0,T]}(((X_i < k_{low}) \rightarrow \mathbf{F}^{[T_1,T_2]}(X_i > k_{high})) \wedge ((X_i > \rho_{high}) \rightarrow \mathbf{F}^{[T_1,T_2]}(X_i < k_{low})) \wedge \mathbf{F}^{[0,T_2]}(X_i > k_{high})), \quad (4.2)$$

expressing the fact that low values of X_i alternate to high values, with a period between T_1 and T_2. The secondary signals are $k_{low} - X_i$, $X_i - k_{high}$, and so on. Here X_i can be one of the three proteins of the Repressilator. In the next discussion, we focus on X_1.

Again, the robustness score gives us a measure of the satisfaction/dissatisfaction of the formula. As

we can see from Figure 4(a), the robustness degree shows a bimodal behaviour also in this case. In particular, in case the formula is false, it gives some degree of information on the amplitude of oscillations, and on the stability of the period (relatively to the formula parameters). In fact, a robustness value of, say, -50 can be obtained for instance if from a point in which $X_i < k_{low}$, the system remains below $k_{high} - 50$ for a whole (half) period of the oscillation (which is constrained to be in $[T_1, T_2]$). This can happen due to low amplitude or irregular period. In Figure 4(b), we plot the average robustness degree against the satisfaction probability, varying the property parameter T_2, showing once again the correlation between the two quantities.



(a) Robustness distribution

(b) satisfaction probability vs robustness degree

Figure 4: Robustness distribution for Formula 4.2 parameters $k_{low} = 20$, $k_{high} = 60$, $T_1 = 100$, $T_2 = 4000$, $T = 7000$. Average robustness is -25.888 and estimated satisfaction probability is 0.35 (left). Satisfaction probability versus average robustness degree. T_2 was varied between 1000 and 7000 in steps of 100 units (right).

5 System Design

We now discuss an application of the robust semantics to the system design problem. The problem we want to tackle is the following:
given a population (hybrid) model, depending on a set of parameters $\theta \in K$, and a specification φ given by a STL formula, find the parameter combination θ^* such that system satisfied φ with probability at least $p \in [0,1]$ as robustly *as possible.*
We will tackle this problem by:

- rephrasing it as an unconstrained optimisation problem, where we seek to optimise the average robustness, using penalty terms to encode for probability constraints. More specifically, assuming we want to enforce the satisfaction probability to be at least q, we add a penalty term of the form $\alpha\|p-q\|$, if $p < q$, and 0 otherwise, where $\alpha < 0$ controls the penalty intensity.
- evaluating the function to optimise using statistical model checking with a fixed number of runs, usually set to 100;
- solving the optimisation problem using an optimisation strategy for reinforcement learning, based on statistical emulation and Gaussian processes regression (Gaussian Process - Upper Confidence

Bound optimisation, GP-UCB [42]).

5.1 Gaussian Processes - Upper Confidence Bound Optimisation

Gaussian Processes. The key ingredient for the design problem is an efficient estimation of the unknown objective function, i.e. the average robustness as a function of the process kinetic parameters. Function approximation is a central task in machine learning and statistics. The general regression task can be formulated as follows [7]: given a set of input-output pairs $(\mathbf{x}_i, y_i), \quad i = 1,..,N$ (*training data*), with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, determine a function $f\colon \mathbb{R}^d \to \mathbb{R}$ s.t. $f(\mathbf{x}_i)$ is optimally close to the target values y_i (usually in terms of minimising a suitable loss function). Several methods exist for addressing this task; in this paper we consider Gaussian Process (GP) regression, a popular Bayesian methodology [40]. GPs are flexible non-parametric distributions over spaces of functions which can be used as prior distributions in a Bayesian framework, where the input-output pairs represent noisy observations of the unknown function. This enables a natural quantification of the uncertainty of the estimated function at every new input value; this uncertainty will play a central role in the optimal design strategy we propose in Section 5. We now give a semi-formal definition of GP [40]:

Definition 3 *A Gaussian Process over a (portion of) $\mathbb{R}^d$ is a collection of random variables indexed by $\mathbf{x} \in \mathbb{R}^d$ such that every finite dimensional marginal distribution is multivariate normal. Furthermore, there exist two functions $\mu\colon \mathbb{R}^d \to \mathbb{R}$ (mean function) and $K\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (covariance function) such that the mean and covariance of the finite dimensional normal marginals is given by evaluating the mean and covariance functions at each point and each pair of points respectively.*

We denote a sample from a GP with mean function μ and covariance function K as

$$f \sim \mathscr{GP}(\mu, K).$$

In practice, the input-output pairs in a regression task are often different features of experimentally observed data points. In this paper, the output points correspond to true functional evaluations of an unknown (and analytically intractable) function of the inputs. In this case, the regression task is often given the special name of *emulation* in the statistics literature: the true (but unknown) function is assumed to be a draw from a GP, and the functional evaluations are used as observations to obtain a posterior estimate of the unknown function. This approach was initially introduced in order to perform sensitivity analysis for deterministic computer models in [32]; in that case, the function evaluations could be assumed to be noiseless (apart from numerical errors that were considered negligible in that paper). In our case, the function linking model parameters to average robustness cannot be computed, and we can only obtain a sampling approximation through a Statistical Model Checking procedure. This means that our function evaluations will be noisy; by virtue of the Central limit theorem we can assume that, provided sufficient samples were used for the SMC estimates, the noise in the observed robustness estimates will be approximately Gaussian.[1] This therefore enables us to obtain an analytical estimate of the posterior process [40]. Furthermore, the SMC samples also allow us to estimate the (sample) variance in the average robustness at every sampled parameter value; this information can also be included leading us to a heteroscedastic (i.e. with non identical noise) regression problem (which is however still analytically tractable).

[1] Note that here we approximate as a GP the average robustness score (or any other fitness score) as a function of parameters. We are not imposing any (Gaussian) approximation of the process itself.

Gaussian Processes Optimization: the GP-UCB algorithm As we have seen, GP emulation provides a convenient way to explore approximately the average robustness of a stochastic process for different values of the model parameters. One could then be tempted to also use the emulated robustness profile for model design, i.e. find the optimum of the emulated function. This strategy, while appealing in its simplicity, is vulnerable to local optima: the emulated function is estimated based on relatively few function evaluation, so that, while the emulator typically provides a good approximation of the true function near the sampled points, regions of parameter space far from the sampled points may contain the true maximum undetected. Using the language of reinforcement learning, maximising the emulated function would privilege exploitation (i.e. using currently available information) at the expense of exploration. Obviously, given sufficient computational power, one may consider sampling many parameter points so as to have sufficient coverage of the whole region of interest; this strategy is however bound to fail in even moderate dimensions due to the curse of dimensionality.

An elegant solution to the above conundrum can be obtained by also considering the *uncertainty* of the emulated function (which is also computed analytically in GP regression): intuitively, one should explore regions where the maximum *could plausibly be*, i.e. regions in parameter space where there is substantial posterior probability mass for the function to take a high value. We formalise these ideas in a recursive search rule, the so called Gaussian Process Upper Confidence Bound (GP-UCB) algorithm: assume we have computed the average robustness at N parameter values (so that we have N input output pairs). Let $\mu_N(\theta)$ and $v_N(\theta)$ be the mean and variance of the GP emulator at a given point in input space θ (recall that the marginal at any point will be Gaussian)[2]. We select the parameter value θ_{N+1} for the next function evaluation according to the following rule

$$\theta_{N+1} = \text{argmax}_\theta \left(\mu_N(\theta) + \beta_{N+1} v_N(\theta)\right) \tag{5.3}$$

where β_{N+1} is a parameter. Thus, the next point for exploration does not maximise the emulated function, but an upper confidence bound at a certain confidence level specified by the parameter β_{N+1} (the quantile can be obtained by applying the inverse probit transform to the parameter). [42] proved that this algorithm converges to the global maximum of the unknown function with high probability (which can be adjusted by varying the algorithm's parameters).

The primary difficulty in applying GP-UCB is that, in order to be able to apply the rule, the emulated function must be computed at a large number of points; while this is obviously not as onerous as evaluating the true unknown function (as the emulator is known analytically), it may still be problematic for high dimensional parameter spaces. Nevertheless, the algorithm can be applied effectively for moderate sized parameter spaces (of the order of 10 parameters), and modular construction may be used to extend to higher dimensional systems [28, 5].

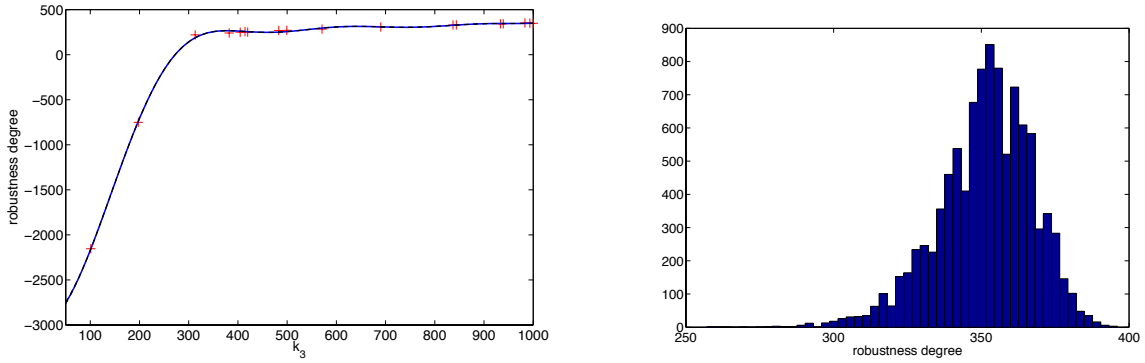
5.2 Experimental Results

Schlögl system. We set up the experiment as follows. We combine the robustness degree of the formula of Section 4.1 and the satisfaction probability in the systems design problem asking, at the same time, to maximize the robustness degree constraining the probability value to remain above 0.75. We varied k_3 uniformly in $[50, 1000]$, fixing all other parameters to the values of Table 1. We ran the GP-UCB optimisation algorithm by first estimating the robustness degree and the satisfaction probability, using statistical model checking, for 30 points sampled randomly and uniformly from the parameter space, and then using the GP-UCB strategy to estimate the maximum of the upper bound function in a grid of 200

[2] We now denote the input as θ to emphasize that they are the parameters of a stochastic process

Parameter mean	Parameter range	Mean probability
$k_3 = 997.78$	[979.31 999.99]	1
Average Robustness	Number of function evaluations	Number of simulation runs
348.97	34.4	3440

Table 2: Statistics of the results of ten experiments to optimize the parameter k_3 in the range $[50, 1000]$.



(a) The estimated robustness function

(b) Robustness distribution after the optimization

Figure 5: The emulated robustness function in the optimisation of k_3 (left). The distribution of the robustness score for $k_3 = 999.99$ (right).

points. If in this grid a point is found with a larger value than those of the observation points, we compute the robustness and satisfaction probability also for this new point, and add it to the observations (thus changing the GP approximation). Termination happens when no improvement can be made after three grid resamplings. Further integration of local maximisation can further improve the method.
In the experiment, repeated 10 times, we used a GP with radial basis kernel [7], with length scale fixed to 0.5 (after standardisation of the parameter range to $[-1, 1]$). The amplitude of the kernel was adaptively set to 60% of the difference between the maximum and the mean value of the robustness for the initial observations. The observation noise was experimentally fixed to 1, by monitoring the average standard deviation at different random parameter combinations.
Results are shown in table 2. As we can see, the result of the optimisation suggests that the more robust system satisfying the specification (i.e. remaining as much as possible above the threshold 300 for a sufficiently long amount of time) is the one obtained for $k_3 = 1000$. We can see that this is the case in Figure 5(b): the system becomes monostable, and X stably remains above 550 units (corresponding to a robustness score above 250 with very high probability).

Repressilator. We consider a different optimisation problem, in which we keep model parameters constant and we try to optimise the parameters of the formula to make the robustness score as large as possible. This can be seen as a sort of dual problem, in which the goal is to learn the emergent behaviour of the model in terms of the most robustly satisfied formula (of fixed structure). Furthermore, the parametrisation of a formula is usually an underestimated problem, as the satisfaction/robustness heavily depends on these parameters. This problem has been partially tacked e.g. in [41] for deterministic

Parameter mean	Parameter range	Mean probability
$T_1 = 231.5$ $T_2 = 6993$	[0, 500] [6963, 7000]	0.900
Average Robustness	Number of function evaluations	Number of simulation runs
17.20	35	3500

Table 3: Statistics of the results of 10 experiments to optimize the parameter T_1 in the range $[0,500]$ and the parameter T_2 in the range $[1000,7000]$.

models, but never for stochastic ones, to authors' knowledge. In particular, we consider Formula (4.2) and optimise the temporal delays T_1 in the range $[0,500]$ and T_2 in the range $[1000,7000]$. This can be seen as an attempt to learn the best bounds on the oscillatory period, through the filter of the logical specification of oscillations of equation (4.2).

In this experiment, we used the same settings of the optimisation algorithms as for the Schlóg system, save for the number of initial observations, set to 25 and constrained to lie in an equi-spaced grid. The parameters of the model are fixed to those shown in the caption of Figure 3.

In this case, due to the highly random duration of each oscillation cycle of the SHA model of the Repressilator, we expect to obtain a low value for T_1 and a large value for T_2. This is indeed the case (see Table 3) confirming the intrinsic instability of the oscillatory pattern of the model. Moreover, the large variability in the value of T_1 can indicate a scarce contribution of the parameter in the determination of the robustness. We confirmed this intuition by emulating the average robustness score as a function of T_1 alone in the range $[0,500]$ for T_2 fixed to 6963 (data not shown), obtaining an essentially flat function: the average robustness varies between 15.01 (near $T_1 = 0$) and 6.89 (near $T_1 = 500$). We expect that an heteroschedastic treatment of noise could improve this estimate of T_1.

6 Conclusion

Discussion. In this paper we investigate a notion of robustness of behaviours of stochastic models, extending the robustness degree of STL formulae in a probabilistic setting. Discussing two case studies, a bistable model and the Repressilator, we showed that the distribution of the robustness degree of a formula provides more information than the satisfaction probability alone, and its average can be used to enforce robust behaviours by optimising it. Such optimisation is carried out using state-of-the-art optimisation algorithms coming from reinforcement learning, which emulate the true function from just few samples, and perform very well in a simulation based scenario. Remarkably, the proposed approach to evaluate robustness and to system design can be applied both to CTMC and to SHA models. We also briefly considered the problem of learning the most effective parameters of a given formula, in the sense of finding the parameter combination maximising the robustness score. This hints towards a more ambitious goal, that of finding machine learning procedures to learn the emergent behaviours (described as temporal logic formulae) from models and from experimental data. Many problems need to be faced to achieve this goal, like how to learn formula structure, how to avoid overfitting (with respect to both formula structure and parameters), how to deal with the curse of dimensionality afflicting GP-UCB and other optimisaton algorithms.

Related Work. Temporal Logic (TL) is a very intuitive specification language to express formally the behavioural property emerging in a complex biological system. Several important extensions of TL, such

as Metric Interval Temporal Logic (MITL) and Signal Temporal Logic (STL) [36], have been introduced to deal with dense-time and real-valued signals, respectively.

In the last years there was a great scientific effort to enrich the classic qualitative semantics of TL or *satisfiability* (yes/no answer for the formula satisfaction of a trajectory) with more powerful and useful notions of quantitave semantics[26, 27, 23, 41, 2] (or *robustness degree*), providing a real value measuring the level of satisfaction or violation for a trajectory of the property of interest.

Several tools, such as BIOCHAM [15], S-TaLiRo [2] Breach [19], are now available to perform robustness analysis on the time series collected in wet-lab experiments or produced by simulation-based techniques. The robustness degree have been successfully employed in the analysis of ODE-based biological models, to tune the parameters that discriminates the behaviours observed experimentally (the so called design problem).

Donze et al. in [21] proposed a multi-step analysis, where they adopt STL to express dynamical properties and they use robustness and sensitivity analysis to sample efficiently the parameter space, searching for feasible regions in which the model exhibit a particular behavior. A similar method appeared also in a previous paper [20] of one of the co-authors.

In [5] the authors proposed a new approach, based on robustness degree, for the design of a synthetic biological circuit whose behaviour is specified in terms of signal temporal logic (STL) formulae. Also in this case stocasticity was not taken into account.

For what concern the stochastic models, while the satisfiability analysis has been considered as a discriminating criterion to tune the parameters in the design process using both simulation-based statistical approximated methods [13] and probabilistic exact methods [6], to the best of our knowledge, we are not aware of approaches using the robustness degree. Another related work in this sense is that of [35], where authors compute exactly upper and lower bounds on the satisfaction probability within a given region of the parameter space.

Future work. The present work uses advanced machine learning concepts to address core problems in formal modelling; this is a relatively new line of work [13] which opens significant new avenues for further research. From the practical point of view, more extensive testing and an efficient and robust implementation (exploiting some of the possible parallelisms e.g. in SMC) will be important for the tool to be adopted. From the theoretical perspective, we plan to use multi-objective optimisation to find good parametrisation for conflicting objectives. Another interesting direction is to combine the design problem with the inference problem, which has recently been addressed for a number of continuous time stochastic systems [37]; this would open the possibility of addressing the control problem for such systems, simultaneously inferring the state of the system and designing the optimal input to lead it to a desired state.

Acknowledgements. Work partially supported by the EU-FET project QUANTICOL (nr. 600708) and by FRA-UniTS.

References

[1] R. Alur, T. Feder & T.A. Henzinger (1996): *The benefits of relaxing punctuality*. *J. ACM*. Available at `http://doi.acm.org/10.1145/227595.227602`.

[2] Y. Annapureddy, C. Liu, G. Fainekos & S. Sankaranarayanan (2011): *S-TaLiRo: A Tool for Temporal Logic Falsification for Hybrid Systems*. In: *Proceedings of TACAS*, doi:10.1007/978-3-642-19835-9_21.

[3] C. Baier, E.M. Clarke, V. Hartonas-Garmhausen, M.Z. Kwiatkowska & M. Ryan (1997): *Symbolic Model Checking for Probabilistic Processes*. In: *Proc. of ICALP '97, the 24th International Colloquium on Automata, Languages and Programming, Bologna, Italy, July 7-11*, *Lecture Notes in Computer Science* 1256, Springer Berlin Heidelberg, pp. 430–440, doi:10.1007/3-540-63165-8_199.

[4] C. Baier, B. Haverkort, H. Hermanns & J.-P. Katoen (2003): *Model-Checking Algorithms for Continuous-Time Markov Chains*. *IEEE Trans. Softw. Eng.* 29(6), pp. 524–541, doi:10.1109/TSE.2003.1205180.

[5] E. Bartocci, L. Bortolussi & L. Nenzi (2013): *A temporal logic approach to modular design of synthetic biological circuits*. In: *In Proc. of CMSB 2013, the 11th International Conference on Computational Methods in Systems Biology, IST Austria, Klosterneuburg, Austria, September 23-25, 2013*, *Lecture Notes in Computer Science* 8130, Springer-Verlag, pp. 164–178, doi:10.1007/978-3-642-39176-7.

[6] E. Bartocci, R. Grosu, P. Katsaros, C. Ramakrishnan & S. A. Smolka (2011): *Model Repair for Probabilistic Systems*. In: *Proceedings of TACAS 2011, the 17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, *Lecture Notes in Computer Science* 6605, Springer Berlin / Heidelberg, pp. 326–340, doi:10.1007/978-3-642-19835-9_30.

[7] C. M. Bishop (2006): *Pattern Recognition and Machine Learning*. Springer.

[8] L. Bortolussi (2010): *Limit behavior of the hybrid approximation of Stochastic Process Algebras*. In: *Proceedings of ASMTA 2010*, doi:10.1007/978-3-642-13568-2_26.

[9] L. Bortolussi, J. Hillston, D. Latella & M. Massink (2013): *Continuous Approximation of Collective Systems Behaviour: a Tutorial*. *Performance Evaluation* 70(5), pp. 317–349, doi:10.1016/j.peva.2013.01.001.

[10] L. Bortolussi & A. Policriti (2008): *Hybrid approximation of stochastic process algebras for systems biology*. In: *Proceedings of IFAC WC*, doi:10.3182/20080706-5-KR-1001.02132.

[11] L. Bortolussi & A. Policriti (2010): *Hybrid Dynamics of Stochastic Programs*. *Theoretical Computer Science* 411(20), pp. 2052–2077, doi:10.1016/j.tcs.2010.02.008.

[12] L. Bortolussi & A. Policriti (in print): *(Hybrid) Automata and (Stochastic) Programs. The hybrid automata lattice of a stochastic program*. *Journal of Logic and Computation*.

[13] L. Bortolussi & G. Sanguinetti (2013): *Learning and Designing Stochastic Processes from Logical Constraints*. In: *Proc. of QEST 2013, 10th International Conference on Quantitative Evaluation of Systems, Buenos Aires, Argentina, August 27-30, 2013*, 8054, pp. 89–105, doi:10.1007/978-3-642-40196-1.

[14] M. L. Bujorianu, J. Lygeros & M. C. Bujorianu (2005): *Bisimulation for General Stochastic Hybrid Systems*. In: *Proceedings of HSCCl*, pp. 198–214, doi:10.1007/978-3-540-31954-2_13.

[15] L. Calzone, F. Fages & S. Soliman (2006): *BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge*. *Bioinformatics* 22, pp. 1805–1807, doi:10.1093/bioinformatics/btl172.

[16] T. Chen, M. Diciolla, M. Kwiatkowska & A. Mereacre (2011): *Time-bounded verification of CTMCs against real-time specifications*. In: *Proc. of FORMATS 2011, the 9th International Conference on Formal Modeling and Analysis of Timed Systems, Aalborg, Denmark, September 21–23*, *Lecture Notes in Computer Science* 6919, Berlin, Heidelberg, pp. 26–42, doi:10.1007/978-3-642-24310-3_4.

[17] M.H.A. Davis (1993): *Markov Models and Optimization*. Chapman & Hall.

[18] Andrea Degasperi & Stephen Gilmore (2008): *Sensitivity analysis of stochastic models of bistable biochemical reactions*. In: *Formal Methods for Computational Systems Biology*, *Lecture Notes in Computer Science* 5016, Springer, pp. 1–20, doi:10.1007/978-3-540-68894-5_1.

[19] A. Donzé (2010): *Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems*. In: *Proceedings of CAV*. Available at `http://dx.doi.org/10.1007/978-3-642-14295-6_17`.

[20] A. Donzé, G. Clermont & C.J. Langmead (2010): *Parameter Synthesis in Nonlinear Dynamical Systems: Application to Systems Biology*. *Journal of Computational Biology* 17(3), pp. 325–336, doi:10.1007/978-3-642-02008-7_11.

[21] A. Donzé, E. Fanchon, L. M. Gattepaille, O. Maler & P. Tracqui (2011): *Robustness analysis and behavior discrimination in enzymatic reaction networks*. *PLoS One* 6(9), p. e24246, doi:10.1371/journal.pone.0024246.

[22] A. Donzé, T. Ferrer & O. Maler (2013): *Efficient Robust Monitoring for STL*. In: *Proc. of CAV 2013, the 25th International Conference on Computer Aided Verification, Saint Petersburg, Russia, July 13-19*, *Lecture Notes in Computer Science* 8044, pp. 264–279, doi:10.1007/978-3-642-39799-8_19.

[23] A. Donzé & O. Maler (2010): *Robust satisfaction of temporal logic over real-valued signals*. In: *Proc. of FORMATS 2010, the 8th International Conference on Formal Modeling and Analysis of Timed Systems, Klosterneuburg, Austria, September 8–10*, 6246, pp. 92–106, doi:10.1007/978-3-642-15297-9_9.

[24] R. Durett (2012): *Essentials of stochastic processes*. Springer, doi:10.1007/978-1-4614-3615-7.

[25] M.B. Elowitz & S. Leibler (2000): *A synthetic oscillatory network of transcriptional regulators*. *Nature* 403, pp. 335–338, doi:10.1038/35002125.

[26] G. Fainekos & G. Pappas (2007): *Robust Sampling for MITL Specifications*. In: *Proc. of FORMATS 2007, the 5th International Conference on Formal Modeling and Analysis of Timed Systems*, *Lecture Notes in Computer Science* 8044, pp. 264–279, doi:10.1007/978-3-540-75454-1_12.

[27] G. E. Fainekos & G. J. Pappas (2009): *Robustness of temporal logic specifications for continuous-time signals*. *Theor. Comput. Sci.* 410(42), pp. 4262–4291, doi:10.1016/j.tcs.2009.06.021.

[28] A. Georgoulas, A. Clark, A. Ocone, S. Gilmore & G. Sanguinetti (2012): *A subsystems approach for parameter estimation of ODE models of hybrid systems*. In: *Proc. of HSB 2012, the 1st International Workshop on Hybrid Systems and Biology*, *EPTCS* 92, pp. 30–41, doi:10.4204/EPTCS.92.3.

[29] D.T. Gillespie (1977): *Exact Stochastic Simulation of Coupled Chemical Reactions*. *J. of Physical Chemistry* 81(25), doi:10.1021/j100540a008.

[30] R. Gunawan, Y. Cao, L. Petzold & F.J. Doyle III (2005): *Sensitivity analysis of discrete stochastic systems*. *Biophysical Journal* 88(4), p. 2530, doi:10.1529/biophysj.104.053405.

[31] K. D. Jones, Konrad V & D. Nickovic (2010): *Analog property checkers: a DDR2 case study*. *Formal Methods in System Design* 36(2), pp. 114–130, doi:10.1007/s10703-009-0085-x.

[32] M. Kennedy & A. O'Hagan (2001): *Bayesian Calibration of Computer Models*. *Journal of the Royal Stat. Soc. Ser. B* 63(3), pp. 425–464, doi:10.1111/1467-9868.00294.

[33] M. Komorowski, M. J. Costa, D. A. Rand & M. PH Stumpf (2011): *Sensitivity, robustness, and identifiability in stochastic chemical kinetics models*. *PNAS USA* 108(21), pp. 8645–8650, doi:10.1073/pnas.1015814108.

[34] Marta Kwiatkowska, Gethin Norman & David Parker (2004): *Probabilistic symbolic model checking with PRISM: a hybrid approach*. *Int. J. Softw. Tools Technol. Transf.* 6(2), pp. 128–142, doi:10.1007/s10009-004-0140-2.

[35] S. Drazan L. Brim, M. Ceska & D. Šafránek (2013): *Exploring Parameter Space of Stochastic Biochemical Systems using Quantitative Model Checking*. In: *Proc. of CAV 2013, the 25th International Conference on Computer Aided Verification, Saint Petersburg, Russia, July 13-19*, *Lecture Notes in Computer Science* 8044, pp. 107–123, doi:10.1007/978-3-642-39799-8_7.

[36] O. Maler & D. Nickovic (2004): *Monitoring Temporal Properties of Continuous Signals*. In: *Proc. of Joint International Conferences on Formal Modeling and Analysis of Timed Systmes, FORMATS 2004, and Formal Techniques in Real-Time and Fault -Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24*, 3253, pp. 152–166, doi:10.1007/978-3-540-30206-3_12.

[37] A. Ocone, A. J. Millar & G. Sanguinetti (2013): *Hybrid regulatory models: a statistically tractable approach to model regulatory network dynamics*. *Bioinformatics* 29(7), pp. 910–916, doi:10.1093/bioinformatics/btt06.

[38] M. Opper, A. Ruttor & G. Sanguinetti (2010): *Approximate inference in continuous time Gaussian-Jump processes*. In: *Proceedings of NIPS 2010, the 4th Annual Conference on Neural Information Processing Systems, 6–9 December 2010, Vancouver, British Columbia, Canada*, pp. 1831–1839. Available at `http://books.nips.cc/papers/files/nips23/NIPS2010_1095.pdf`.

[39] Amir Pnueli (1977): *The temporal logic of programs*. Foundations of Computer Science, IEEE Annual Symposium on 0, pp. 46–57, doi:10.1109/SFCS.1977.32.

[40] C. E. Rasmussen & C. K. I. Williams (2006): *Gaussian Processes for Machine Learning*. MIT Press.

[41] A. Rizk, G. Batt, F. Fages & S. Soliman (2008): *On a Continuous Degree of Satisfaction of Temporal Logic Formulae with Applications to Systems Biology*. In: Proc. of CMSB 2008, the 6th International Conference on Computational Methods in Systems Biology, Rostock, Germany, October 12–15, Lecture Notes in Computer Science 5307, pp. 251–268, doi:10.1007/978-3-540-88562-7_19.

[42] Niranjan Srinivas, Andreas Krause, Sham M. Kakade & Matthias W. Seeger (2012): *Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting*. IEEE Transactions on Information Theory 58(5), pp. 3250–3265, doi:10.1109/TIT.2011.2182033.

[43] H. L. S. Younes, M. Z. Kwiatkowska, G. Norman & D. Parker (2004): *Numerical vs. Statistical Probabilistic Model Checking: An Empirical Study*. In: Proc. of 2004, the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Barcelona, Spain, March 29 - April 2, doi:10.1007/978-3-540-24730-2_4.

Robustness Analysis for Value-Freezing Signal Temporal Logic

L. Brim, T. Vejpustek, D. Šafránek, and J. Fabriková [*]

Faculty of Informatics
Masaryk University
Botanická 68a, Brno, Czech Republic
safranek@fi.muni.cz

In our previous work we have introduced the logic STL*, an extension of Signal Temporal Logic (STL) that allows value freezing. In this paper, we define robustness measures for STL* by adapting the robustness measures previously introduced for Metric Temporal Logic (MTL). Furthermore, we present an algorithm for STL* robustness computation, which is implemented in the tool Parasim. Application of STL* robustness analysis is demonstrated on case studies.

1 Introduction

A particular place among formalisms adopted by systems biology is occupied by temporal logics, which serve as a language for description of biological systems behaviour. Resulting temporal formulae can be used during computer-aided system analysis, such as model checking [5], which automatically verifies whether a model satisfies given temporal formula. Methods based on temporal logics have been successfully employed to study biological phenomena [28, 25, 16] (see [3] for review).

Since most of current models developed in computational systems biology have the form of ordinary differential equations, model checking cannot be directly employed and is typically replaced with a non-exhaustive procedure of monitoring [24]. In this setting, a (finite) set of signals representing individual time-courses of the model is monitored wrt a given temporal specification. In particular, the respective temporal logics are interpreted over individual signals that are most typically simplified to discrete timed state sequences (time series) approximating the continuous trajectories by means of numerical simulation. Temporal logics fitting this interpretation are Metric Temporal Logic (MTL) [21] and Signal Temporal Logic (STL) [24], which allow quantifying modalities with the time frame represented by a closed time interval. MTL possesses both discrete and continuous semantics, as it can be interpreted over both infinite timed state sequences and continuous signals. STL is practically focused and is defined for piece-wise linear approximations of continuous signals.

Temporal logics are satisfactorily used in systems biology to express statements about a single instance of system behaviour such as *in five minutes, concentration of glucose will be greater than 0.8*. However, many biological hypotheses contain relative temporal references, e.g., *after protein P reaches the maximum concentration, a steady concentration of P is reached which is less than half of the maximum*. Such a scenario can be found, e.g., in feed-forward genetic regulatory circuits generating pulses in expression signals [18]. In common temporal logics, such a general query cannot be expressed. This is because the values in different time points cannot be compared, i.e., the property *in five minutes, concentration of glucose will rise by 0.2*, which relates glucose concentration at current time and in the

[*] The work has been supported by the Grant Agency of Czech Republic grant GAP202/11/0312 and by the EC OP project No. CZ.1.07/2.3.00/20.0256.

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 20–36, doi:10.4204/EPTCS.125.2

future, cannot be specified. Of specific interest is oscillatory behaviour, e.g., a sequence of gradually increasing peaks followed by a limit cycle with a stable amplitude [15]. In order to express the increasing amplitude, it is necessary to detect local extremes in signals and compare respective signal values. This cannot be achieved using common temporal logics. Signals with a series of increasing local maxima have been observed, e.g., in response of FGF signalling pathways transferring stimuli from mutated FGFR3 receptors to target effectors affecting bone cells growth [22]. Since the mentioned behaviour correlates with the phenotype of dysplasia, it is necessary to develop models that mechanistically capture the respective signalling pathways and to analyse circumstances under which the undesired behaviour occurs. This makes a necessary step before designing a targeted medical treatment. To this end, temporal logics and verification procedures which allow to capture and analyse such complex phenotypes have to be developed.

In [7], we have introduced a new temporal logic STL* which alleviates limitations mentioned above. Expressiveness of STL* is enhanced by signal-value freeze operator which stores values at certain time, which may be referred to in the future. This allows STL* to specify and distinguish various dynamic aspects which occur in biological systems, in addition to the phenomena mentioned above, these can be, e.g., damped oscillations [17] or local extremes in species concentration. It is worth noting that some more complex queries can be expressed in traditional temporal logic by including signal derivatives into atomic propositions. However, this does not directly apply to queries mentioned above. One can express the presence and shape of a local extreme by using the first and second derivative, but still the values in particular time points have to be compared in order to express the complex queries.

An important concept associated with biological systems and temporal logics is *robustness*, the ability of a system to maintain its function against perturbations [20]. Since system function can be expressed in the terms of temporal logic, we speak of robustness with respect to a temporal logic formula, which can be quantified and computed [14, 26]. Robustness significantly enhances model analysis and gives an optimization goal for model parameter estimation/synthesis [11, 9, 27].

This paper introduces the notion of robustness in the value-freezing logic STL* setting. In particular, we extend the continuous and discrete measure defined for MTL by Fainekos et al. [14] to the semantic domain of STL*. Robustness of the input signal with respect to STL* formula delineates the robust neighbourhood of the signal (the maximal "tube" around the signal where the formula is satisfied). The robustness measure we propose (Section 3) is defined inductively wrt the formula structure and is based on a distance metrics employed on the signal domain extended with (multiple) dimensions representing the frozen time points. The theoretical framework is computationally supported with an algorithm based on solving the optimization problem (Section 4) provided that the logic is restricted to linear predicates. Special consideration is given to optimization of the formula to overcome unnecessary computational overhead.

Implementation of our algorithm is included as a part of Parasim [12], a tool aimed as a modular environment for monitoring and robustness analysis of kinetic models. To demonstrate the usage and evaluate the performance, we present case studies of two simple kinetic models (Section 5).

1.1 Related Work

Robustness measures have been defined for three temporal logics targeting deterministic continuous systems: STL [11], MTL [14] and QFLTL [26]. We adopt the concept of behaviour-based robustness introduced on a fragment of MTL by Fainekos et al. [14], who define robustness measure for MTL formulae with discrete [13] and continuous [14] semantics. In [14], Fainekos et. al prove a theorem connecting discrete and continuous robustness, which is valuable for robustness computation. A recent

tool [2] implements the method. Donzé et al. [24] use STL to define a distinct robustness measure, albeit constructed from [14], and propose its application for space exploration [11, 9], which was implemented in the Breach Toolbox [8]. The work is further improved from the computational point of view in [10]. Our implementation (Parasim) is based on a simplified version of the robustness analysis algorithm for STL where the sensitivity-based computation of local robustness is replaced with direct computation of trajectories distance. The extension for STL* as presented in Section 4 is implemented in this setting.

Fages et al. [26] introduced property-based approach to robustness that fixes input behaviour and examines the formula. Basically, it measures the extent to which the formula can be modified while preserving its satisfaction. The tool BioCham implements this idea [4]. Extended LTL logic with constraints over real numbers (quantifier-free LTL) is employed being defined for finite discrete time-series.

It is worth noting that the problem of formula satisfiability is undecidable for MTL [21]. To achieve decidability, Alur and Henzinger specified further conditions on intervals associated with temporal operators [1]. The result, metric interval temporal logic, requires all intervals to be non-singular and is interpreted over timed state-sequences where time points are replaced with consecutive time intervals. STL was introduced by Maler and Nickovic in [24] as a basis for their monitoring procedure. Technically, it comprises a variant of MITL interpreted over real signals. Because of its practical purpose, in [7] we selected STL as a good candidate for extension with value-freezing.

2 Background

STL* is evaluated over finite time continuous signals (finite signals for short).

Definition 2.1 *Let $n \in \mathbb{N}$ and $T = [0,r]$ where $r \in \mathbb{R}^+$. Then $s : T \to \mathbb{R}^n$ is a* bounded continuous-time signal *and T its* time domain. *We denote $l(s) = r$ the length of signal s.*

Signal value freezing is facilitated by the following structure which is used to store time values at various time points which then can be referred to in predicates.

Definition 2.2 *Let $\mathscr{I}$ be a finite index set.* Frozen time vector *is a function:*

$$t^* : \mathscr{I} \to \mathbb{R}_0^+$$

The symbol $t_i^* = t^*(i)$ is referred to as i-th frozen time. For convenience reasons and without loss of generality, we will henceforth assume that an index set $\mathscr{I} = \{1,\ldots,m\}$ is given, where $m \in \mathbb{N}$.

Predicates comprise Boolean expressions over values of a signal s at time t and each frozen time t_i^*, where x_j denotes the j-the component of the signal at time t, i.e. $s(t) = (x_1,\ldots,x_j,\ldots,x_n)$, and x_j^{*i} the j-th component at time t_i^*. When $|\mathscr{I}| = 1$, we usually omit the index of asterisk, e.g. $x_i^* = x_i^{*1}$.

We consider only predicates given by linear inequalities, so that analytic expressions of predicate robustness is possible.

Definition 2.3 *Let $n \in \mathbb{N}$, $b \in \mathbb{R}$ and $a_{ij} \in \mathbb{R}$ where $i \in \{0\} \cup \mathscr{I}$, $j \in \{1,\ldots,n\}$ and not all a_{ij} are zero. A* predicate *is defined as a subset of $\mathbb{R}^n \times (\mathbb{R}^n)^{\mathscr{I}}$ such that:*

$$\sum_{j=1}^{n} a_{0j}x_j + \sum_{i=1}^{|\mathscr{I}|}\sum_{j=1}^{n} a_{ij}x_j^{*i} + b \geq 0$$

Predicates are specified by the set of associated coefficients a_{ij}, b (where coefficients a_{0j} are connected with the current time t). Therefore, for convenience reasons, we will use these coefficients to represent predicates. Predicates with all coefficients a_{ij} zero were omitted since they are of the form $b \geq 0$ and, therefore, trivially true or false.

Predicates with equality (i.e. having $=$ in place of $\geq$), although theoretically possible, lack practical value, as they are not robust (small perturbation may invalidate the property). This has been already argued in [7], albeit without defining the concept of robustness. Since robustness of predicates with strict and non-strict inequalities does not differ, we consider only non-strict inequalities.

Freeze operator is used to store the time point into frozen time vector, thus facilitating signal value freezing. The following definition introduces an auxiliary concept of storing the current time t as the ith component of the frozen time vector.

Definition 2.4 *Let t^* be frozen time vector, $i, j \in \mathscr{I}$ and $t \in \mathbb{R}_0^+$.* Freezing *i*th component of t^* in t *is denoted as $t^*[i \leftarrow t]$ and defined:*

$$t^*[i \leftarrow t](j) = \begin{cases} t & i = j \\ t_j^* & i \neq j \end{cases}$$

Definition 2.5 *Syntax of STL* is defined by the following grammar:*

$$\varphi ::= \mu \mid \top \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \boldsymbol{U}_I \varphi_2 \mid *_i \varphi$$

where $i \in \mathscr{I}$, $\top$ denotes the true constant, μ is a predicate as of Definition 2.3 and $I \subseteq \mathbb{R}_0^+$ a closed non-singular interval.

Note that all Boolean connectives and temporal operators **F** and **G** can be defined using the basic operators defined above. Similarly to predicates, when $|\mathscr{I}| = 1$, we usually omit the index of freeze operator, as in $*\mathbf{G}_I(x > x^*) = *_1 \mathbf{G}_I(x > x^{*_1})$. Henceforth, let $i, \mu, \varphi, \varphi_1, \varphi_2$ be the same as in Definition 2.5.

Definition 2.6 *Let $s \in (\mathbb{R}^n)^T$ be a signal, $t \in T$ a time point and $t^* \in T^{\mathscr{I}}$ a frozen time vector. Formula satisfaction is defined inductively:*

$$\begin{array}{lcl}
(s,t,t^*) \models \top & & \\
(s,t,t^*) \models \mu & \iff & (s(t), s \circ t^*) \in \mu \\
(s,t,t^*) \models \neg\varphi & \iff & (s,t,t^*) \not\models \varphi \\
(s,t,t^*) \models \varphi_1 \vee \varphi_2 & \iff & (s,t,t^*) \models \varphi_1 \vee (s,t,t^*) \models \varphi_2 \\
(s,t,t^*) \models \varphi_1 \boldsymbol{U}_I \varphi_2 & \iff & \exists\, t' \in t \oplus I : (s,t',t^*) \models \varphi_2 \wedge \\
 & & \forall\, t'' \in [t,t'] : (s,t'',t^*) \models \varphi_1 \\
(s,t,t^*) \models *_i \varphi & \iff & (s,t,t^*[i \leftarrow t]) \models \varphi
\end{array}$$

Operator $\circ$ is used to denote function composition, i.e. $(s \circ t^*) \in (\mathbb{R}^n)^{\mathscr{I}}$ and $(s \circ t^*)(i) = s(t_i^*)$ and $t \oplus I$ stands for $\{t + u \mid u \in I\}$.

Definition 2.7 *Let $s \in (\mathbb{R}^n)^T$ be signal and φ formula. Formula satisfaction by signal is given:*

$$s \models \varphi \iff (s, 0, \mathbf{0}) \models \varphi$$

where $\mathbf{0}$ *denotes the* zero frozen time vector*, i.e.* $\{(i,0) | i \in \mathscr{I}\}$.

Intuitively, interpretation of $*_i\,\varphi$ is the following: freeze operator stores signal values at the time of $*_i\,\varphi$ evaluation, which can then be referred to using index i in predicates of φ. An example property, *"in the next five time units, x increases by* 8*"* can be specified as:

$$*\mathbf{F}_{[0,5]}(x \geq x^* + 8)$$

where x^* refers to value of x at time 0.

When intervals associated with until operators are bounded, satisfaction of a given formula can be decided on any finite signal of sufficient length. This length can be determined from the formula structure in a way similar to [24] and corresponds to the furthest time point (among all possible signals) which has to be examined in order to determine formula satisfaction. This clearly also holds for frozen time values.

Definition 2.8 *Let φ be a formula. The* necessary input length *for φ, $l(\varphi)$ is defined inductively:*

$$\begin{aligned} l(\top) = l(\mu) &= 0 \\ l(\neg\varphi) = l(*_i\,\varphi) &= l(\varphi) \\ l(\varphi_1 \vee \varphi_2) &= \max(l(\varphi_1), l(\varphi_2)) \\ l(\varphi_1\, \boldsymbol{U}_I\, \varphi_2) &= \max(l(\varphi_1), l(\varphi_2)) + \sup I \end{aligned}$$

When $l(s) < l(\varphi)$ we state that $s \not\models \varphi$.

Frozen time indices and freeze operators share some similarities with variables and quantifiers of predicate logic. We may distinguish free and bound indices, where index i is free if it is used in a predicate (i.e. coefficient a_{ij} is not zero for some j) and is not in the scope of operator $*_i$.

Naturally, whenever i is free in φ, then $s \models \varphi$ iff $s \models *_i\,\varphi$, since t_i^* is zero in both cases.

Additionally, we may substitute for free indices of a formula in a manner similar to variable substitution. However, it only makes sense to substitute one index for another, which we will denote *index renaming* and express as $\varphi[\pi]$ where π is a total function on $\mathscr{I}$ (but not necessarily a permutation – two indices can be renamed to one) or $\varphi[k/l]$, where k is renamed to l. To preserve formula semantics, renaming is only safe when no free index becomes bound after renaming in any subformula.

3 Robustness Measures for STL*

Following from STL* semantics, robustness of signal s with respect to formula φ is given for each time point t and frozen time vector t^* and denoted by $\rho(\varphi, s, t, t^*)$. We also define $\rho(\varphi, s) = \rho(\varphi, s, 0, \mathbf{0})$. Robustness of signal s with respect to formula φ is a value, which under-approximates the distance of s from the set of signals where φ has different truth value [14]. To express this formally, we first need to define certain basic concepts (where S is a set of signals):

- Distance of signals is given by their maximum pointwise distance: $d(s,s') = \max_{t\in\mathbb{R}_0^+} d(s(t), s'(t))$
- *Set distance* is given by minimum distance to the set: $\mathbf{dist}(s,S) = \min\{d(s,s') \mid s' \in S\}$
- *Set depth* is given by set distance to the complement: $\mathbf{depth}(s,S) = \mathbf{dist}\left(s, \overline{S}\right)$
- *Signed distance* is given: $\mathbf{Dist}(s,S) = \begin{cases} -\mathbf{dist}(s,S) & s \notin S \\ \mathbf{depth}(s,S) & s \in S \end{cases}$

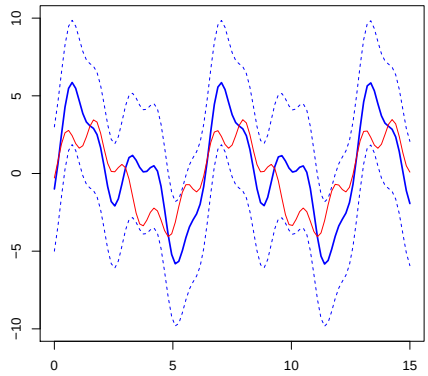


Figure 1: Signal s (blue, thick) and borders of its robust neighbourhood (blue, dashed) with an example of a signal (red) contained in the robust neighbourhood (adapted from [14]).

The value $\rho(\varphi,s)$ underapproximates the signed distance of s from the set of all signals satisfying φ, $\mathscr{L}(\varphi)$, i.e. $|\rho(\varphi,s)| \leq |\mathbf{Dist}(s,\mathscr{L}(\varphi))|$ holds while their signs are identical. The absolute value of $\rho(\varphi,s)$ thus delineates an equidistant tube where all signals satisfy φ if and only if s does – the *robust neighbourhood* of s (see Figure 1).

It would be desirable to define the robustness equal to the signed distance; however, by [14], the robustness computation would not be feasible then. In order to be sound, the robustness definition has to satisfy the following property (for any φ, s, t and t^*):

$$-\mathbf{dist}\,(s,\mathscr{L}_{t,t^*}(\varphi)) \leq \rho(\varphi,s,t,t^*) \leq \mathbf{depth}\,(s,\mathscr{L}_{t,t^*}(\varphi))\,, \tag{1}$$

where $\mathscr{L}_{t,t^*}(\varphi) = \{s \mid (s,t,t^*) \models \varphi\}$. Since $\mathbf{depth}(s,\mathscr{L}_{t,t^*}(\varphi)) = 0$ when $(s,t,t^*) \not\models \varphi$ (and analogously for **dist**), this actually requires that:

1. $s \models \varphi \implies 0 \leq \rho(\varphi,s,t,t^*) \leq \mathbf{depth}\,(s,\mathscr{L}_{t,t^*}(\varphi))$,
2. $s \not\models \varphi \implies -\mathbf{dist}\,(s,\mathscr{L}_{t,t^*}(\varphi)) \leq \rho(\varphi,s,t,t^*) \leq 0$.

Robustness is defined inductively for each logical connective from its semantics in such manner that Boolean functions $\wedge$ and $\vee$ are replaced by real functions min and max (respectively). Quantifiers in the semantics of operator **U** can then be expressed by infinite disjunction or conjunction. Robustness wrt predicate μ is defined as $\mathbf{Dist}(s,\mathscr{L}_{t,t^*}(\mu))$, i.e. the ideal value without underapproximation. If $\rho(\mu,s)$ was lower, it would diminish resulting robustness value, for robustness wrt formula cannot be greater than robustness wrt any of its predicates. Soundness of this definition (property (1)) is, naturally, proved inductively wrt formula structure.

This has already been established by Fainekos et al. in [14], albeit for MTL which does not allow signal value freezing. Nevertheless, their definition can be directly extended for STL*. Intuitively, this is due to frozen time values being only stored by freeze operators and retrieved in predicates, which does not affect other logical connectives. The full proof can be found in [29] (page 83).

Consequently, we have to define robustness for the freeze operator. It follows from its semantics:

$$\mathscr{L}_{t,t^*}(*_i\,\varphi) = \{s \mid (s,t,t^*) \models *_i\,\varphi\} = \{s \mid (s,t,t^*[i \leftarrow t]) \models \varphi\} = \mathscr{L}_{t,t^*[i\leftarrow t]}(\varphi)$$

Thus, robustness of freeze operator can be defined in the following manner:

$$\rho(*_i \varphi, s, t, t^*) = \rho(\varphi, s, t, t^*[i \leftarrow t])$$

Assume $-\mathbf{dist}(s, \mathscr{L}_{t,t^*}(\varphi)) \leq \rho(\varphi, s, t, t^*) \leq \mathbf{depth}(s, \mathscr{L}_{t,t^*}(\varphi))$ for any t, t^*. Therefore, it also holds for t and $t^*[i \leftarrow t]$ and thus:

$$-\mathbf{dist}(s, \mathscr{L}_{t,t^*[i\leftarrow t]}(\varphi)) \leq \rho(\varphi, s, t, t^*[i \leftarrow t]) \leq \mathbf{depth}(s, \mathscr{L}_{t,t^*[i\leftarrow t]}(\varphi))$$

From which follows the validity of (1) for $\rho(*_i \varphi, s, t, t^*)$. STL* robustness for logical connectives is presented in Figure 2.

$$\begin{aligned}
\rho(\top, s, t, t^*) &= +\infty \\
\rho(\neg\varphi, s, t, t^*) &= -\rho(\varphi, s, t, t^*) \\
\rho(\varphi_1 \vee \varphi_2, s, t, t^*) &= \max\left(\rho(\varphi_1, s, t, t^*), \rho(\varphi_2, s, t, t^*)\right) \\
\rho(\varphi_1 \,\mathbf{U}_I\, \varphi_2, s, t, t^*) &= \max_{t' \in t \oplus I} \min\left(\rho(\varphi_2, s, t', t^*), \min_{t'' \in [t,t']} \rho(\varphi_1, s, t'', t^*)\right) \\
\rho(*_i \varphi, s, t, t^*) &= \rho(\varphi, s, t, t^*[i \leftarrow t])
\end{aligned}$$

Figure 2: Robustness of STL* logical connectives.

3.1 Robustness of Predicates

Finding $\mathbf{Dist}(s, \mathscr{L}_{t,t^*}(\mu))$ generally constitutes a convex analysis problem [14]. Thus, it could be solved using convex programming for each t and t^*, which would, however, greatly increase computation time, and therefore, analytic solution is preferable. To this end, we have restricted STL* predicates to be linear.

For predicate μ with coefficients a_{ij}, b, the problem of finding $\mathbf{Dist}(s, \mathscr{L}_{t,t^*}(\mu))$ can be reduced to optimization of $f(\mathbf{d}) = \max_i \sum_j d_{ij}^2$ (where $i \in \mathscr{I}$ and $j \in \{1, \ldots, n\}$) under the constraint $\sum_i \sum_j a_{ij} d_{ij} + \varepsilon = 0$ for some positive ε. This is a non-trivial problem, since f is not differentiable at point $\mathbf{d}$ where $f(\mathbf{d}) = \sum_j d_{kj}^2 = \sum_j d_{lj}^2$ for some $k \neq l$. To solve it, generalized method of Lagrange multipliers from [6] was used, resulting in the following definition of the robustness ρ (detailed derivation can be found in [29] (page 47)).

Definition 3.1 *Let μ be a predicate with coefficients a_{ij}, b. Then*

$$\rho(\mu, s, t, t^*) = \frac{\sum_j a_{0j} s_j(t) + \sum_i \sum_j a_{ij} s_j(t_i^*) + b}{\sum_i \sqrt{\sum_j a_{ij}^2}}$$

for arbitrary s, t, t^*, i ranging over $\mathscr{I}$, j ranging over $\{1, \ldots, n\}$.

The numerator corresponds to the left-hand side value of the predicate.

It holds that $\rho(\mu, s, t, t^*) = \mathbf{Dist}(s, \mathscr{L}_{t,t^*}(\mu))$, unless some time points given by t and t^* are equal. This originates from the optimization problem, where $t_k^* = t_l^*$ (or $t = t_k^*$) would constitute another constraint, which might change the solution.

Suppose that $t_k^* = t_l^*$ (reasoning for $t = t_k^*$ is similar). We can merge (sum) coefficients a_{kj} and a_{lj} for any given j, which effectively reduces the number of considered frozen times. Robustness of

predicates with merged coefficients is greater, since the denominator of definition 3.1 becomes smaller as $\sqrt{\sum_j \left(a_{kj}+a_{lj}\right)^2} \leq \sqrt{\sum_j a_{kj}^2}+\sqrt{\sum_j a_{lj}^2}$ due to triangle inequality. Therefore, even if we disregard possible time point equality, property (1) still holds. However, the greater the value of $\rho(\mu,s,t,t^*)$ is, the better approximation of $\mathbf{Dist}(s,\mathscr{L}_{t,t^*}(\varphi))$ is obtained. Therefore, we will investigate two distinct cases when time points can be equal:

1. It happens consistently for given formula φ and predicate μ, i.e. φ is built in such way that the same time value is stored by freeze operator associated with both indices, such as:

$$\psi = \mathbf{G}_{I_1}(*_i \neg *_j \mathbf{F}_{I_2}(x^{*_i}+x^{*_j} \geq x))$$

2. It is a result of $\varphi \equiv *_i(\varphi_1 \mathbf{U}_I \varphi_2)$ (or similar formula) evaluation:

$$(s,t,t^*) \models \varphi \iff (s,t,t^*[i \leftarrow t]) \models \varphi_1 \mathbf{U}_{[a,b]} \varphi_2 \iff$$
$$\exists t' \in [a+t,b+t] : (s,t',t^*[i \leftarrow t]) \models \varphi_2 \wedge \forall t'' \in [t,t'] : (s,t'',t^*[i \leftarrow t]) \models \varphi_1$$

   When $a=0$, it may occur that $t'=t$. Additionally, $t'' \in [t,t']$, therefore, satisfaction of φ_1 by $(s,t,t^*[i \leftarrow t])$ has to be evaluated. The equality of t and i-th frozen time may be propagated to predicates. We have decided to omit this case in order to simplify robustness computation.

3.2 Improving Approximation

The formula ψ (see above) is obviously badly written, since it can be reformulated with only one frozen time index: $\mathbf{G}_{I_1}(\neg * \mathbf{F}_{I_2}(2x^* \geq x))$. This eliminates time point equality and thus improves robustness approximation. We have formulated three rules which can be used to automatically rewrite formula so that it does not induce consistent time point equality (while preserving its meaning):

1. Freeze operator is distributive over Boolean connectives. Consequently, freeze operators can be moved down along the formula syntax tree until they reach a temporal operator, predicate or another freeze operator.
2. Freeze operator preceding predicate can be merged with the predicate (associated coefficients being merged with coefficients for unfrozen time).
3. Two consecutive freeze operators and their associated indices can be merged. However, in order to preserve the formula meaning, a completely new index has to be chosen as the result of merging.

Subsequently, all STL* formulae can be written in such manner that each freeze operator is followed by until operator, which also ensures that all frozen time indices generally refer to distinct time points. Indeed, all meaningful formulae (i.e. not serving to illustrate semantic peculiarities) in [7] are specified in this manner.

This reinforces the connection between temporal operators and freeze operators expressiveness. Subsequently, it may be practical to define an alternate STL* syntax, where signal value freezing is directly tied to the until operator, such as $\varphi_1 \mathbf{U}_I^{*_i} \varphi_2 \equiv *_i(\varphi_1 \mathbf{U}_I \varphi_2)$. However, we do not deem it necessary, seeing that it entails no expressiveness gain. Moreover, the current syntax of STL* may permit shorter and more transparent formulae.

It should be noted that although application of previous rules may increase number of indices used in a formula (due to the rule (3) which introduces one new index), it does not increase the number of free indices in each subformula. On the contrary, the number of free indices may decrease.

4 Computation

To compute (or monitor) robustness of continuous signal, we use the approach of Fainekos et al. [14], which is based on discrete robustness semantics. The following procedure is used:

1. Sample input signal $s : T \to \mathbb{R}^m$ into a *timed state sequence* $(\tau, \sigma) : \mathbb{N} \to T \times \mathbb{R}^m$.
2. Compute robustness over points of the resulting timed state sequence (i.e. the discrete robustness).

This only approximates continuous robustness of s. When MTL robustness is concerned, Fainekos et al. give bound for error introduced by this approximation under certain conditions, which can be summarized as signal sampling being sufficiently dense with respect to given formula. We assume this strong theorem translates to STL* (as STL* robustness extends MTL robustness) and deem the previous procedure good approximation for an input signal with large enough sampling rate.

Before the robustness monitoring algorithm is described, we should note that it can also be used to decide formula satisfaction, since positive robustness implies formula satisfaction (and negative its invalidity). However, when $\rho(\varphi, s) = 0$ no information about formula satisfaction can be derived. Additionally, robustness measure only underapproximates the robust neighbourhood, and so the robustness value may be zero even if clearly s satisfies φ. Consequently, classical monitoring may produce more precise results.

Algorithm 1 computes robustness for a STL* formula and sufficiently long timed state sequence (which may constitute a sampled signal). It copies inductive definition of robustness with recursive calls of procedure MONITOR (line 4), which computes robustness only in the points of given state sequence. Therefore, instead of frozen time vector $t^* : \left(\mathbb{R}_0^+\right)^{\mathscr{J}}$, *frozen state vector* $\iota^* : \mathbb{N}^{\mathscr{J}}$ is used. The computation starts at zero index and zero frozen state vector (line 3), which ensures only robustness values needed for resulting robustness evaluation are computed.

Robustness values with respect to subformulae of input formula are not stored. Instead, they are computed every time procedure MONITOR is called on a given subformula. The reasoning behind this practise is the following: For the majority of formulae, the value of robustness for given ι and ι^* is obtained by a simple – constant-time – operation on just a single value of robustness (or two in the case of $\vee$). Additionally, the robustness with respect to predicates can be computed in constant time.

The only operator where robustness depends on robustness values over an interval is the until operator (and by extension all derived temporal operators). Consequently, robustness values associated with until operators are stored. Furthermore, when MONITOR($\varphi_1 \mathbf{U}_I \varphi_2, \iota, \iota^*$) is called for the first time, robustness values with respect to $\varphi_1 \mathbf{U}_I \varphi_2$ for ι^* and all ι' are precomputed (see lines 10–17) by the procedure PRECOMPUTEUNTIL, which constitutes an algorithmic version of robustness definition for until operator. These precomputed values are expected to be referred to later, since robustness computation is restricted to time interval $[0, l(\varphi)]$ which comprises all input values necessary to evaluate $\rho(\varphi, (\tau, \sigma))$.

4.1 Complexity

Apparently, the most time-consuming task of Algorithm 1 is the PRECOMPUTEUNTIL procedure, which is quadratic to the number of states in the input timed state sequence. In the worst case it is called for each ι^*. Therefore, the complexity of Algorithm 1 is in $\mathscr{O}\left(|\varphi| \cdot n^{2|\mathscr{J}|}\right)$ where n is the size of input timed state sequence. For sampled signals, it may be expressed using necessary length, resulting in alternate complexity formulation: $\mathscr{O}\left(|\varphi| \cdot l(\varphi)^{2|\mathscr{J}|} \cdot f^{2|\mathscr{J}|}\right)$ where f is the sampling rate of input signal, which correlates with the precision of robustness computation. Space complexity can be bounded by the same function.

Algorithm 1 Robustness Monitoring for STL*

Input: STL* formula φ and timed state sequence (τ,σ) of length greater than $l(\varphi)$ (see Definition 2.8).
Output: The value of $\rho(\varphi,(\tau,\sigma))$.

For any i free in φ, $\varphi \leftarrow *_i \varphi$.

$P \leftarrow \emptyset$ ▷ Precomupted robustness values.
return MONITOR$(\varphi, 0, \mathbf{0})$

procedure MONITOR$(\varphi, \imath, \imath^*)$
  if $\varphi \equiv \top$ **then return** $+\infty$
  else if $\varphi \equiv \mu$ **then return** $\rho(\mu,(\tau,\sigma),\imath,\imath^*)$ ▷ According to Definition 3.1.
  else if $\varphi \equiv \neg\varphi_1$ **then return** $-$MONITOR$(\varphi_1,\imath,\imath^*)$
  else if $\varphi \equiv \varphi_1 \vee \varphi_2$ **then return** $\max($MONITOR$(\varphi_1,\imath,\imath^*),$MONITOR$(\varphi_2,\imath,\imath^*))$
  else if $\varphi \equiv *_i \varphi_1$ **then return** MONITOR$(\varphi_1,\imath,\imath^*[i \leftarrow \imath])$
  else if $\varphi \equiv \varphi_1 \mathbf{U}_{[a,b]} \varphi_2$ **then**
    if $(\varphi,\imath^*) \in \mathbf{dom}(P)$ **then**
      return $P(\varphi,\imath^*)(\imath)$
    else
      $\rho \leftarrow$ PRECOMPUTEUNTIL$(\varphi_1,\varphi_2,a,b,\imath^*)$
      $P \leftarrow P \cup ((\varphi,\imath^*),\rho)$
      return $\rho_\imath$
    end if
  end if
end procedure

procedure PRECOMPUTEUNTIL$(\varphi_1,\varphi_2,a,b,\imath^*)$
  $i \leftarrow 0$
  $l \leftarrow \max(l(\varphi_1),l(\varphi_2))$
  $\rho \leftarrow \emptyset$ ▷ Sequence of robutness values.
  while $\tau_i + b + l \le l(\tau)$ **do**
    $j \leftarrow 0$
    $r_1 \leftarrow$ MONITOR$(\varphi_1,i,\imath^*)$
    while $\tau_{i+j} < \tau_i + a$ **do** ▷ Before $[\tau_i+a,\tau_i+b]$.
      $r_1 \leftarrow \min(r_1,$MONITOR$(\varphi_1,i+j,\imath^*))$
      $j \leftarrow j+1$
    end while
    $r \leftarrow r_1$
    while $\tau_{i+j} \le \tau_i + b$ **do** ▷ Inside $[\tau_i+a,\tau_i+b]$.
      $r_1 \leftarrow \min(r_1,$MONITOR$(\varphi_1,i+j,\imath^*))$
      $r_2 \leftarrow$ MONITOR$(\varphi_2,i+j,\imath^*)$
      $r \leftarrow \max(r,\min(r_1,r_2))$
      $j \leftarrow j+1$
    end while
    $\rho \leftarrow \rho \cup \{(i,r)\}$ ▷ Set the value of ρ_i.
    $i \leftarrow i+1$
  end while
end procedure

The parameter most adversely affecting the algorithm complexity is the size of frozen time index set $|\mathscr{I}|$. Naturally, $\mathscr{I}$ can be restricted to indices used in input formula. In most practical cases, their number will be small. This is supported by the following result:

Theorem 4.1 *Any formula φ can be rewritten into a semantically equivalent formula which uses only so many indices as is the maximum number of free indices in subformulae of φ.*

Note that the number of free indices may increase as we descend into subformulae.

This statement derives from the fact that an index only serves to associate one freeze operator with a set of coefficients in one or more predicates and it is free on all paths between this freeze operator and all associated predicates. Therefore, indices which are never simultaneously free need not be different.

The result of this theorem can be realized by an automatic procedure which renames frozen time indices in a formula while traversing its syntax tree (using DFS). This procedure stores pairs of indices $[k/l]$ corresponding to the renaming of source index k in the original formula φ to destination index l in its optimized version φ'. When the procedure encounters freeze operator $*_i$, new pair $[i/m]$ is introduced where m is the smallest unused destination index and the operator is changed to $*_m$. Whenever k becomes free in φ, the pair $[k/l]$ is removed and l can be reused. Upon reaching a predicate, all stored pairs are applied as a renaming.

This procedure is described in greater detail in [29] (page 44) where additional justification of its correctness can also be found.

Together with freeze operator merging described in Section 3.2 (which does not increase number of free indices), this can considerably decrease the number of indices used in a formula and thus the time complexity of robustness monitoring. Although intelligent formula specification may result in already optimal formula, the existence of automatic optimization procedures reduces demands on writers of formulae.

4.2 Implementation

The algorithm has been implemented as an extension of the tool Parasim [12]. Parasim is a highly modular Java-based open-source tool with graphical user interface for computing robustness of a model with respect to perturbations. Integrating the algorithm presented in this paper into an already existing tool has an additional advantage of facilitating the use of STL* robustness in practise.

Given a model, STL* formula and perturbation set, Parasim samples the perturbation set into points and for each point simulates the model and computes robustness of the resulting signal with respect to STL* robustness measure. In the neighbourhood of signals with low robustness, additional points are sampled. Formula optimizing algorithms are implemented to maximize efficiency.

5 Case Study

By employing the Parasim tool we have conducted several experiments on two simple population dynamics models. The experiments have also served us to briefly evaluate the algorithm performance (in the setting of the Parasim tool).

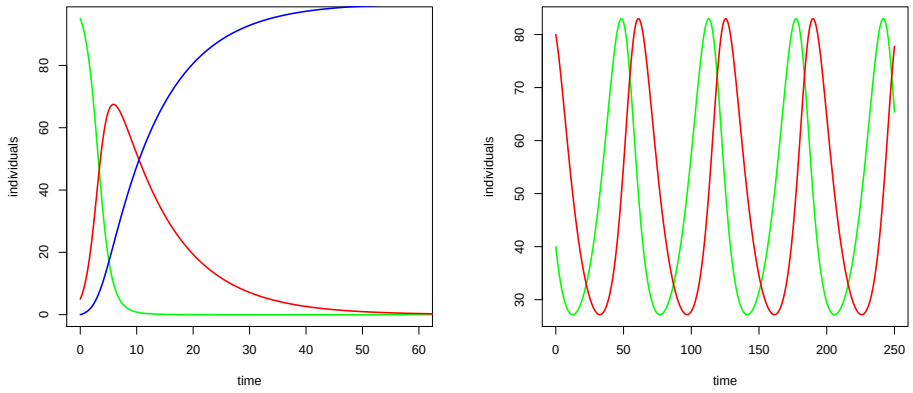
5.1 SIR Model

First, we demonstrate the robustness analysis on the model simulating an outbreak of an infectious disease in a population [19]. The simulated population is divided into three categories: *susceptible* (*S*),

infected (I) and *recovered* (R). A susceptible individual can become infected by contact with another infected individual and an infected individual may recover. The ODE model is the following:

$$\frac{dS}{dt} = -\alpha SI \qquad \frac{dI}{dt} = \alpha SI - \beta I \qquad \frac{dR}{dt} = \beta I$$

Where α is the *contact rate* which correlates to probability of disease transmission, while β, the *recovery rate*, takes into account the standard length of recovery. A typical simulation of this model (see Figure 3a) includes a rapid increase in infected individuals, which is then followed by their gradual recovery.



(a) (b)

Figure 3: (a) Typical development of SIR model, showing the number of susceptible (green), infected (red) and recovered (blue) individuals. (b) Typical development of populations in predator-prey model, showing number of prey (green) and predator (red).

In this case study, we compare robustness analysis based on a formula containing value-freezing with respect to a freezing-free formula analysis exploiting a similar behavioural pattern. In particular, we consider the following formulae:

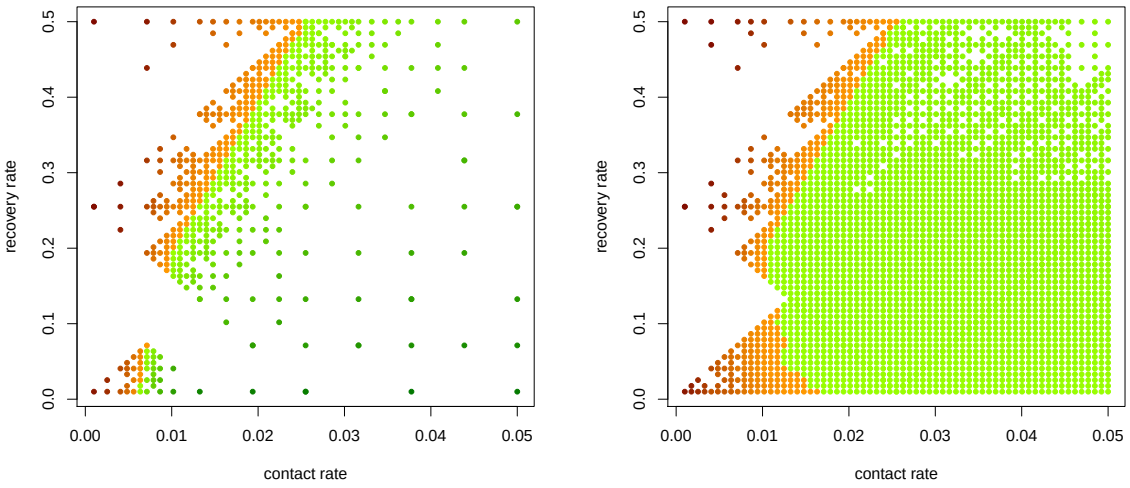
$$\text{STL} : \varphi_1 = \mathbf{F}_{[1,5]}(I \geq 50) \qquad \text{STL}^* : \varphi_2 = \mathbf{F}_{[1,5]}\left(I \geq 50 \wedge * \mathbf{G}_{[0.25,5]}(I^* \geq I)\right)$$

Both formulae require the number of infected individuals to be greater than 50 at some time in the interval $[1,5]$, while φ_2 also requires this number to be the local maximum (the number of infected individuals is required to decrease after reaching this maximum).

The robustness with respect to both properties was analysed on perturbations of both contact rate and recovery rate. Results are presented in Figure 4.

While the satisfaction sets of φ_1 and φ_2 (delineated by positive robustness) are essentially identical, the actual robustness values show a significant difference. Generally, when they are positive, the value of robustness with respect to φ_1 at given point is considerably greater than the corresponding value of robustness with respect to φ_2. In Figure 4, this can be seen as lighter shade of green points in 4b. Also, lower robustness causes the apparent increase in the number of points.

The reason for the rapid change in robustness comes from evaluation of the subformula $* \mathbf{G}_{[0.25,5]}(I^* \geq I)$ that describes the local extreme. When evaluated in time t, robustness is proportional to the difference $(I[t] - I[t+0.25])$ (by Definition 3.1). In practise, the difference is small provided that the descent of I is not extremely steep. This causes such formulae to have typically low robustness values on common signals.



(a) Robustness wrt φ_1.

(b) Robustness wrt φ_2.

Figure 4: Robustness of SIR model with respect to φ_1 and φ_2 for variable contact and recovery rates. Robustness was positive in green points and negative in orange points. Darker colour represents greater absolute value of robustness.

5.2 Predator-Prey Model

In the second case study we analyse the predator-prey model [23, 30], which attains oscillating behaviour for a wide variety of parameters. We use a variant of the Lotka-Volterra model represented by the following ordinary differential equations:

$$\frac{dX}{dt} = \nu X - \alpha XY \qquad \frac{dY}{dt} = \alpha XY - \mu Y$$

The model simulates a situation where a prey species X is hunted by a predator species Y with the simplifying assumption that predator birth rate and prey death rate are equal and proportional to the probability of prey and predator contact, and thus to the product of both species populations. We use the following coefficients: prey natality (ν), predator mortality (μ) and predation rate (α). Typical behaviour of this models constitutes periodic oscillations (see Figure 3b).

We consider perturbation of two aforementioned coefficients, ν and α, and compute robustness with respect to two properties specified by the following formulae:

$$\psi_1 = \mathbf{G}_{[0,300]} * \mathbf{F}_{[0,100]} (X \geq Y^*)$$
$$\psi_2 = \mathbf{G}_{[0,300]} \left(X \geq 1 \wedge Y \geq 1 \wedge \mathbf{F}_{[0,50]} * \left(\mathbf{F}_{[0,75]} (X^* - X \geq 25) \wedge \mathbf{F}_{[0,75]} (X - X^* \geq 25)\right)\right)$$

The property ψ_1 requires that for each time point $t \in [0,300]$, there is a subsequent time point $t' \in [t, t+100]$ such that population of prey in t' is greater than population of predators in t. According to Definition 3.1 its corresponding robustness can be expressed as follows:

$$\rho(\varphi, s) = \min_{t \in [0,300]} \max_{t' \in [t, t+100]} \frac{X[t'] - Y[t]}{2}$$

where $X[t']$ and $Y[t]$ denote values of s associated with given species at given time. The robustness value is maximized with respect to t' and minimized with respect to t, therefore, it uses maximal values of both X and Y. Consequently, this property can be interpreted as maximum population of prey being greater then maximum population of predators (restricted to given intervals).

Formula ψ_2 is based on the similar principle. While rejecting aberrant behaviour where population of one of the species drops below one individual, intuitively, it requires that there always is time in the future when population of prey can increase or decrease by 25 individuals, which is stated by the subformula $\mathbf{F}_{[0,50]} * \left(\mathbf{F}_{[0,75]}\left(X^* - X \geq 25\right) \wedge \mathbf{F}_{[0,75]}\left(X - X^* \geq 25\right)\right)$. Therefore, ψ is satisfied when the difference between maximal and minimal prey population is greater than 50 and the associated robustness is proportional to this difference. Again, we have avoided use of the extreme property, which would adversely affect robustness value.

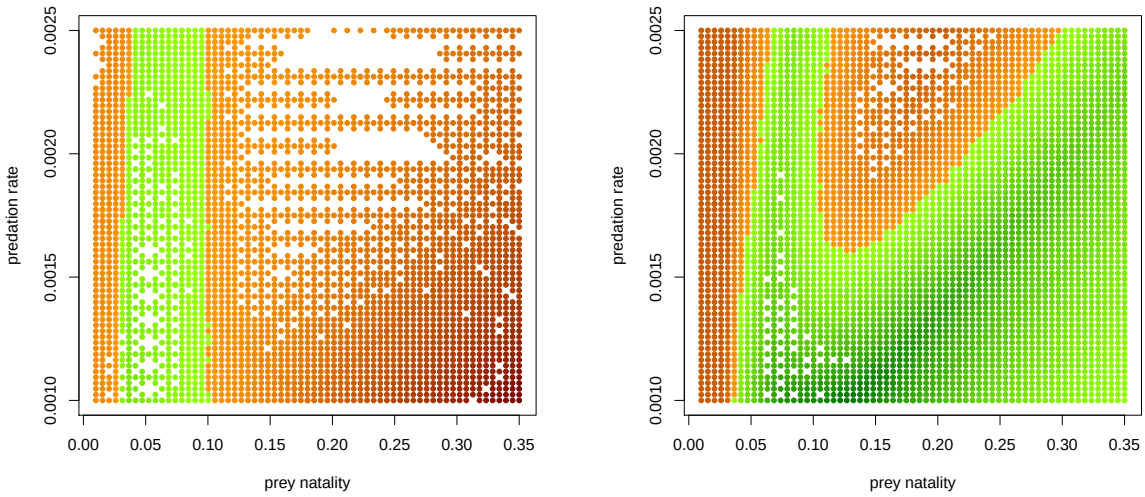


Figure 5: Robustness of predator-prey model with respect to ψ_1 (left) and ψ_2 (right) for variable prey natality and predation rate. Robustness was positive in green points and negative in orange points. Darker colour represents greater absolute value of robustness.

Results of this analysis are presented in Figure 5. Here, we should point out that small prey natality produced behaviour where predator population approached zero and period of oscillations was greatly increased. For such behaviour, intervals used in ψ_1 and ψ_2 were shorter than one period.

Apparently, satisfaction of ψ_1 is not affected by predation rate. More interestingly, when prey natality increases, predator population exceeds that of prey (see Figure 5 (left)). Figure 5 (right) shows that amplitude of prey population oscillation is affected by both prey natality and predation rate.

The above results have been confirmed by simulation.

5.3 Performance

Performance of robustness analysis is summarized in Table 1. All results have been obtained by executing the algorithm implementation on a 4 core 2 GHz CPU with 4 GB RAM. Each computation has been arranged into 8 threads. For each analysis we have set an optimal resolution of the trajectories (number of simulated points). The number of simulated trajectories has been bounded by the number of refinement iterations in the Parasim parameter space sampling procedure.

It is worth noting that all analysed properties consist only of **F** and **G** operators for which the procedure is optimized by employing Lemire queues in the same way as proposed in [10]. This is based on an optimal streaming algorithm for computing maxima (resp. minima) of a numerical sequence and allows to reduce the quadratic complexity wrt formula size to linear.

Property (model)	formula size	# trajectories	# points per a trajectory	time
φ_1 (SIR)	2	250	500	8.6 s
φ_2 (SIR)	6	1365	1000	15.2 s
ψ_1 (Predator-Prey)	4	831	400	85.4 s
ψ_2 (Predator-Prey)	12	1293	423	309.4 s

Table 1: Performance of the robustness computation measured on the prototype implementation.

The increase in computation time in the case of ψ_1 is caused by longer time intervals quantifying the temporal operators. Computation of the property ψ_2 has been slowed down due to insufficient memory.

6 Conclusion

In this paper we have set up a robustness measure for a value-freezing extension of STL. The robustness of a signal with respect to a given STL* property is based on the distance of the signal from signals violating the property. We have introduced a measure that is proved to fulfil requirements imposed on robustness measures as defined in [14]. This guarantees that the robustness measure is defined correctly. We have derived the algorithm for STL* robustness computation from the discrete robustness and implemented it as an extension of the tool Parasim [12].

Some of the properties from case studies required comparison of signal values at near frozen time points. Robustness of such properties is typically small. This is only natural as such properties represent stricter requirements on signals. However, this feature may also constitute a detriment for tools such as Parasim, which use robustness to direct perturbation set sampling. This is the exact case of analysed SIR model and property φ_2. It must be noted, though, that this problem is encompassed by the much broader issue of meaningful property design.

In [14] the authors quantify error in robustness value caused by the approximate computation. We have not yet explored this possibility for STL* robustness measures and leave this for future work. However, results in [14] imply this error is inversely proportional to the rate of input signal sampling.

References

[1] Rajeev Alur, Tomás Feder & Thomas A. Henzinger (1996): *The Benefits of Relaxing Punctuality*. *Journal of the ACM* 43(1), pp. 116–146, doi:10.1145/227595.227602.

[2] Yashwanth Singh Rahul Annapureddy, Che Liu, Georgios E. Fainekos & Sriram Sankaranarayanan (2011): *S-TaLiRo: A Tool for Temporal Logic Falsification for Hybrid Systems*. In: *Tools and Algorithms for the Construction and Analysis of Systems*, *Lecture Notes in Computer Science* 6605, Springer, pp. 254–257, doi:10.1007/978-3-642-19835-9_21.

[3] Luboš Brim, Milan Češka & David Šafránek (2013): *Model Checking of Biological Systems*. In Marco Bernardo, Erik Vink, Alessandra Pierro & Herbert Wiklicky, editors: *Formal Methods for Dynamical Systems*, *Lecture Notes in Computer Science* 7938, Springer Berlin Heidelberg, pp. 63–112, doi:10.1007/978-3-642-38874-3_3.

[4] Laurence Calzone, François Fages & Sylvain Soliman (2006): *BIOCHAM: An Environment for Modeling Biological Systems and Formalizing Experimental Knowledge*. Bioinformatics 22(14), pp. 1805–1807, doi:10.1093/bioinformatics/btl172.

[5] Edmund M. Clarke, Orna Grumberg & Doron A. Peled (2000): *Model Checking*. MIT Press.

[6] Frank H. Clarke (1983): *Optimization and Nonsmooth Analysis*. S.I.A.M.

[7] Petr Dluhoš, Luboš Brim & David Šafránek (2012): *On Expressing and Monitoring Oscillatory Dynamics*. In: Proceedings First International Workshop on Hybrid Systems and Biology, Open Publ. Assoc., pp. 73–87, doi:10.4204/EPTCS.92.6.

[8] Alexandre Donzé (2010): *Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems*. In Tayssir Touili, Byron Cook & Paul Jackson, editors: Computer Aided Verification, Lecture Notes in Computer Science 6174, Springer Berlin Heidelberg, pp. 167–170, doi:10.1007/978-3-642-14295-6_17.

[9] Alexandre Donzé, Gilles Clermont, Axel Legay & Christopher J. Langmead (2009): *Parameter Synthesis in Nonlinear Dynamical Systems: Application to Systems Biology*. In: Research in Computational Molecular Biology, Lecture Notes in Computer Science 5541, Springer, pp. 155–169, doi:10.1007/978-3-642-02008-7_11.

[10] Alexandre Donzé, Thomas Ferrre & Oded Maler (2013): *Efficient Robust Monitoring for STL*. In Natasha Sharygina & Helmut Veith, editors: Computer Aided Verification, Lecture Notes in Computer Science 8044, Springer Berlin Heidelberg, pp. 264–279, doi:10.1007/978-3-642-39799-8_19.

[11] Alexandre Donzé & Oded Maler (2010): *Robust Satisfaction of Temporal Logic over Real-Valued Signals*. In: FORMATS 2010, Springer, pp. 92–106, doi:10.1007/978-3-642-15297-9_9.

[12] Faculty of Informatics, Masaryk University (2013): *Parasim: Tool for Parallel Simulations and Verification*. Available at `https://github.com/sybila/parasim/wiki`.

[13] Georgios Fainekos & George Pappas (2006): *Robustness of Temporal Logic Specifications*. In Klaus Havelund, Manuel Nunez, Grigore Rosu & Burkhart Wolff, editors: Formal Approaches to Software Testing and Runtime Verification, Lecture Notes in Computer Science 4262, Springer Berlin Heidelberg, pp. 178–192, doi:10.1007/11940197_12.

[14] Georgios E. Fainekos & George J. Pappas (2009): *Robustness of Temporal Logic Specifications For Continuous-Time Signals*. Theoretical Computer Science 410(42), pp. 4262–4291, doi:10.1016/j.tcs.2009.06.021.

[15] Leon Glass & JoelS. Pasternack (1978): *Prediction of limit cycles in mathematical models of biological oscillations*. Bulletin of Mathematical Biology 40(1), pp. 27–44, doi:10.1007/BF02463128.

[16] Radu Grosu, Gregory Batt, FlavioH. Fenton, James Glimm, Colas Guernic, ScottA. Smolka & Ezio Bartocci (2011): *From Cardiac Cells to Genetic Regulatory Networks*. In Ganesh Gopalakrishnan & Shaz Qadeer, editors: Computer Aided Verification, Lecture Notes in Computer Science 6806, Springer Berlin Heidelberg, pp. 396–411, doi:10.1007/978-3-642-22110-1_31.

[17] Benno Hess (2000): *Periodic Patterns in Biology*. Naturwissenschaften 87(5), pp. 199–211, doi:10.1007/s001140050704.

[18] Shai Kaplan, Anat Bren, Erez Dekel & Uri Alon (2008): *The incoherent feed-forward loop can generate non-monotonic input functions for genes*. Molecular Systems Biology 4(1), pp. –, doi:10.1038/msb.2008.43.

[19] William O. Kermack & Anderson G. McKendrick (1927): *A Contribution to the Mathematical Theory of Epidemics*. Proceedings of the Royal Society of London. Series A 115(772), pp. 700–721, doi:10.1098/rspa.1927.0118.

[20] Hiroaki Kitano (2004): *Biological Robustness*. Nature Reviews Genetics 5(11), pp. 826–837, doi:10.1038/nrg1471.

[21] Ron Koymans (1990): *Specifying Real-Time Properties with Metric Temporal Logic*. Real-Time Systems 2, pp. 255–299, doi:10.1007/BF01995674.

[22] Pavel Krejci, Vitezslav Bryja, Jiri Pachernik, Ales Hampl, Robert Pogue, Pertchoui Mekikian & William R Wilcox (2004): *FGF2 inhibits proliferation and alters the cartilage-like phenotype of RCS cells*. Experimental Cell Research 297(1), pp. 152 – 164, doi:10.1016/j.yexcr.2004.03.011.

[23] Alfred J. Lotka (1925): *Elements of Physical Biology*. Williams and Wilkins, Baltimore.

[24] Oded Maler & Dejan Nickovic (2004): *Monitoring Temporal Properties of Continuous Signals*. In Yassine Lakhnech & Sergio Yovine, editors: Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Lecture Notes in Computer Science 3253, Springer Berlin Heidelberg, pp. 152–166, doi:10.1007/978-3-540-30206-3_12.

[25] Elisabetta De Maria, François Fages, Aurélien Rizk & Sylvain Soliman (2011): *Design, Optimization and Predictions of a Coupled Model of the Cell Cycle, Circadian Clock, DNA Repair System, Irinotecan Metabolism and Exposure Control under Temporal Logic Constraints*. Theoretical Computer Science 412(21), pp. 2108–2127, doi:10.1016/j.tcs.2010.10.036.

[26] Aurélien Rizk, Grégory Batt, François Fages & Sylvain Soliman (2011): *Continuous valuations of temporal logic specifications with applications to parameter optimization and robustness measures*. Theor. Comput. Sci. 412(26), pp. 2827–2839, doi:10.1016/j.tcs.2010.05.008.

[27] Aurlien Rizk, Gregory Batt, Franois Fages & Sylvain Soliman (2009): *A general computational method for robustness analysis with applications to synthetic gene networks*. Bioinformatics 25(12), pp. i169–i178, doi:10.1093/bioinformatics/btp200.

[28] Szymon Stoma et al. (2013): *STL-based Analysis of TRAIL-induced Apoptosis Challenges the Notion of Type I/Type II Cell Line Classification*. PLoS Comput Biol 9(5), p. e1003056, doi:10.1371/journal.pcbi.1003056.

[29] Tomáš Vejpustek (2013): *Robustness Analysis of Extended Signal Temporal Logic STL**. Master's thesis, Masaryk University, Faculty of Informatics. Available at `http://is.muni.cz/th/324713/fi_m/`.

[30] Vito Volterra (1928): *Variations and Fluctuations of the Number of Individuals in Animal Species living together*. Journal du Conseil 3(1), pp. 3–51, doi:10.1093/icesjms/3.1.3.

The impact of high density receptor clusters on VEGF signaling

Ye Chen
Department of Mathematics
West Virginia University
chenye@math.wvu.edu

Christopher Short
Department of Mathematics
West Virginia University
cshort3@mix.wvu.edu

Ádám M. Halász
Department of Mathematics
West Virginia University
halasz@math.wvu.edu

Jeremy S. Edwards
Department of Molecular Genetics and Microbiology
University of New Mexico
jsedwards@salud.unm.edu *

Vascular endothelial growth factor (VEGF) signaling is involved in the process of blood vessel development and maintenance. Signaling is initiated by binding of the bivalent VEGF ligand to the membrane-bound receptors (VEGFR), which in turn stimulates receptor dimerization. Herein, we discuss experimental evidence that VEGF receptors localize in caveloae and other regions of the plasma membrane, and for other receptors, it has been shown that receptor clustering has an impact on dimerization and thus also on signaling. Overall, receptor clustering is part of a complex ecosystem of interactions and how receptor clustering impacts dimerization is not well understood. To address these questions, we have formulated the simplest possible model. We have postulated the existence of a single high affinity region in the cell membrane, which acts as a transient trap for receptors. We have defined an ODE model by introducing high- and low-density receptor variables and introduce the corresponding reactions from a realistic model of VEGF signal initiation. Finally, we use the model to investigate the relation between the degree of VEGFR concentration, ligand availability, and signaling. In conclusion, our simulation results provide a deeper understanding of the role of receptor clustering in cell signaling.

1 Introduction

The topic of the spatial organization of the cell membrane and its impact on receptor clustering and signal initiation are part of a complex and very active field, illustrating the challenges faced by quantitative systems biology. There are ultimately *two different levels* of spatial and mathematical detail involved. Signaling in response to the presence of VEGF occurs on the level of *the entire cell*. Factors that enhance or inhibit signaling are of crucial importance in the quest to understand and control the progression of various types of cancer. At the other end of the spectrum, the detailed topography of the cell membrane, the mobility and binding characteristics of individual receptors, occur at scales of a few *tens of nanometers*, literally *thousands of times smaller* than the size of the cell. These aspects are actively investigated by various microscopy modalities, which provide a wealth of extremely detailed data.

One of the tasks of meaningful modeling is to bridge the gap between these scales, and to identify rational approaches to abstractions and approximations that can connect data and insights from different scales. Hybrid systems result naturally when continuous degrees of freedom are abstracted into discrete states or regimes. Here we discuss a continuous model that results as the ultimate abstraction of the

*This work was supported by NIH grants R01 GM104973 (to JSE and ÁMH) and K25 CA131558 (ÁMH).

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 37–52, doi:10.4204/EPTCS.125.3

complex biological system mentioned above. Our starting point is the microscopic observation that receptors tend to concentrate in small patches. The distribution, size, and physical characteristics of these patches can be inferred from microscopic observations. In other work [9, 17], we performed detailed, spatial simulations of receptors in a network of high- and low- density membrane patches. These models are naturally abstracted to a network of patches that act as well-mixed, communicating containers. The final abstraction is one where all high density patches are treated as a single, well mixed compartment, in contact with another one, that represents the rest of the membrane.

In this paper we focus on the final, "top level" abstraction, which becomes quite complex when one combines it with a realistic, kinetic model of signal initiation. We use a recently developed approach [7] to identify and investigate the steady states of the model, and discuss the implications of high density patches on the phenomenology of signaling.

This rest of this paper is organized as follows. We first provide some background on the role of VEGF, its signaling mechanism, and the potential modulation of VEGF signaling by the spatial structure of the cell membrane. The following subsection is devoted to the phenomenology of receptor clustering and the available experimental data. We conclude the introduction by sketching the sequence of abstractions and approximations required to extract high, cell level behaviors from the detailed microscopic observations. Sec. 2 is devoted to the definition of the model, and to the derivation of analytical expressions for the steady states, that require solving a one dimensional algebraic equation. Section 3 discusses results obtained by numerically solving the steady state expressions.

Background

Angiogenesis, the growth of new blood vessels from preexisting vessels, is switched on or off by the dynamic balance among numerous angiogenic stimulators and inhibitors (the 'angiogenesis switch' hypothesis) [2, 8]. Among the various growth factors, vascular endothelial growth factor (VEGF) and its receptors (VEGFR) have received much attention, because of their fundamental role in tumorigenesis and other pathologies [2, 10, 19]. Initially identified as a vascular permeability factor that increased leakiness of blood vessels [25], the role of VEGF in regulating angiogenesis was discovered later [4, 21].

Signaling by VEGFR is initiated by binding of the ligand dimer to the extracellular domain of the receptor, which stimulates receptor homo- and hetero-dimerization [14, 23, 27]. Receptor dimerization is followed by protein kinase activation, trans-autophosphorylation, recruitment of signaling molecules, and activation of distinct pathways. Due to its bivalence, VEGF binding may precede and induce the dimerization of its receptors, by the binding of a second receptor to the free binding site of the ligand (see Figure 1 for the explicit process). Ligand-induced or -enhanced receptor dimerization is a feature present in several other receptor-ligand families including EGF and immune receptors.

Mathematical models of VEGF binding [5] generally represent the cell membrane as a single, homogeneous entity, equivalent to a "well-mixed compartment" whose state is sufficiently characterized by a single concentration value for each of the substances of interest. This is justified if there are no significant inhomogeneities and all molecules can diffuse and mix freely over the entire membrane surface, as in the classic Singer-Nicholson fluid mosaic model [26]. However, our understanding of the cell membrane has evolved significantly since 1972. The current picture [28] is more structured, with microdomains of lipids and proteins [6, 13, 24]. Modern microscopy techniques [22, 29] provide direct evidence of the effect of these structures on membrane receptor localization and movement [1, 18, 20], revealing receptor clusters in static images, and intervals of confinement in small areas separated by jumps or "hops" in single particle tracking.

Spatial organization in the membrane can potentially have a major impact on signaling pathways

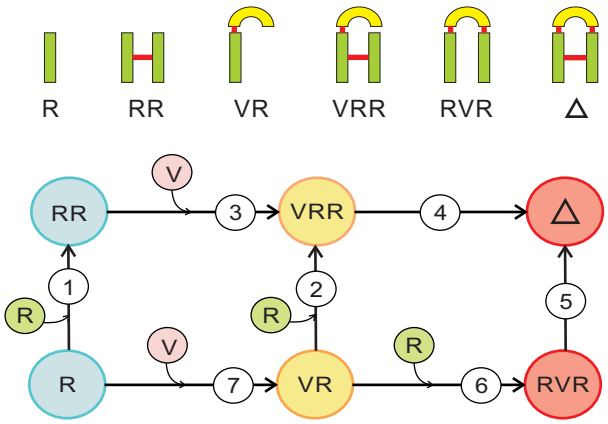


Figure 1: The dimerization and ligand binding reactions form a network of 7 reactions in the VEGF signal initiation model of [5]. Receptors (R) may bind one of the two poles of a VEGF ligand (V), and may form a direct bond with another receptor. In the ligand-induced dimerization (LID) sequence, receptors can not form a direct bond outside a pre-existing complex; signal initiation progresses through reactions $(7,6,5)$. In the dynamic pre-dimerization (DPD) sequence, receptors may dimerize before binding ligand (adding reactions 1 and 2).

that rely on interaction between membrane-bound molecules. Receptor dimerization, either through (ligand-dependent or independent) direct receptor-receptor binding, or by crosslinking through ligand[s], requires the collision of two membrane-bound receptors, and is thus influenced by the mobility and possible confinement of receptors. In turn, receptor dimerization is a necessary step in signal initiation, and therefore the mobility and spatial organization of membrane receptors must be part of the quantitative understanding of many cell signaling pathways.

The microscopic picture

VEGF receptors share many properties of other receptor tyrosine kinases. Similarly to EGF receptors, they form ligand-bound dimers in order to activate their intracellular tyrosine kinase domains [12]. Experimental and theoretical investigation of EGF binding [3, 9, 17] emphasized the importance of spatial distribution of receptors. Ample experimental evidence indicates that EGF receptors can have a highly inhomogeneous distribution characterized by small areas of high density [30], and exhibit anomalous diffusion [20]. There are other examples of receptors that exhibit clustering and anomalous diffusion [1]. Receptor accumulation in high density patches has an impact on dimerization and on signaling [3,9,15–17].

The data analysis pipeline in this case begins with detailed microscopic observations, that provide either static images of a large fraction of the receptors of interest, or, in the case of single particle tracking (SPT), time histories of the positions of a small fraction of the receptors. In the first case (Fig. 2) the imaging modality is transmission electron microscopy (TEM); receptors are tagged with small (6-10 nanometer) sized gold particles, and one image covers a few μm^2 capturing up to a few hundred receptors. In the second case (SPT), tracking is typically performed using fluorescent tags, but the technique can only identify the position of well separated molecules; this modality provides up to several hundred snapshots covering a few seconds, yielding a few tens of trajectories.

Static images of receptors (even in the absence of ligand) typically reveal a *clustering* pattern, where receptors tend to accumulate in groups ranging from a few to a few tens of receptors. This occurs for VEGFR and also other receptor types, for which there is no evidence of a collective binding mechanism. The generally accepted explanation is that receptors accumulate in *microdomains*: small, physically

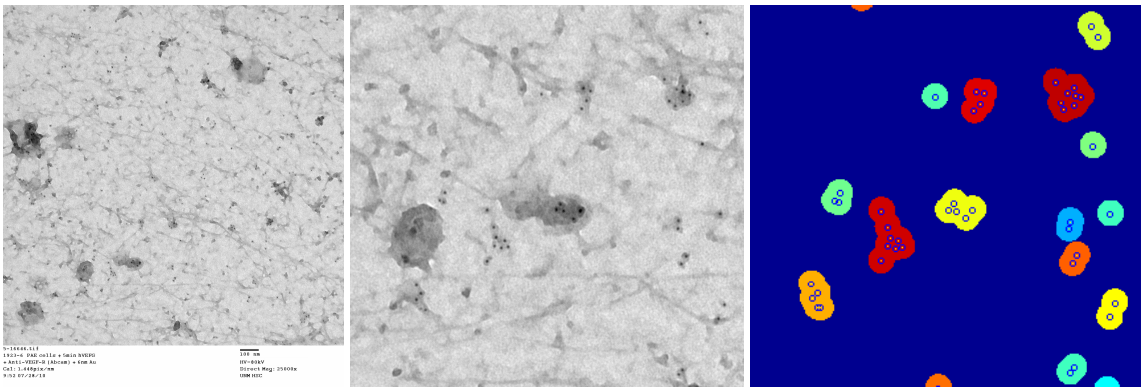


Figure 2: (*L*) Transmission electron microscopy (TEM) images of nano-gold labelled VEGF receptors on the membrane of PAE-KDR cells, courtesy of the Wilson lab at UNM. The second image is a detail of the first one. Gold particles appear as dark spots, whose coordinates are extracted in a semi-automatic procedure. (*R*) The result of a distance based cluster identification algorithm applied to the second image.

delimited areas of the cell membrane that result from partitioning by actin filaments (elements of the cytoskeleton) or are formed by local aggregations of specific types of lipids and / or membrane proteins. However, there is no clear understanding of the mechanism of receptor accumulation. One hypothesis is that some microdomains have a specific molecular composition that results in an affinity for the receptors; receptors may diffuse in and out of them, but the crossing probability is asymmetric. This hypothesis of *confining [micro]domains* is supported by time resolved tracking data, that reveals anomalous (non-Brownian) diffusion and under some circumstances, spatial confinement.

From microscopic details to global behavior

Our model building program relies on a sequence of models, with three different levels of detail. Abstractions and/or average behaviors obtained from one level serve as inputs to the next, higher level. We use the idea of high affinity patches as a working hypothesis.

1. At the *microscopic* level, we investigate the localization, motion and interactions of *individual receptors*. Static distributions (Fig. 2), exhibit clusters that are not consistent with a random distribution. The identification of clusters can be done by a hierarchic clustering algorithm. The distribution of nearest neighbor distances, as well as other statistical measures, point toward a structure of *high density patches*, essentially identified with the observed clusters. Receptors are distributed randomly *within* the patches, and the patches themselves also appear to be distributed randomly.

The observed receptor trajectories exhibit anomalous diffusion. We model this with *random walks* in the presence of various geometries of semi-permeable barriers. Comparisons of simulated and experimental step size distributions also support the high density patch hypothesis. In summary, the microscopic data combined with a Brownian motion model can provide estimations of the *individual and combined size*, as well as the *attractiveness* of the high density patches. In addition, direct measurements based on SPT can provide *exit and entrance rates* as well as *dimerization and dissociation rates* for molecular species of interest.

2. The information on the size and properties of high affinity patches is used at the intermediate, *mesoscopic* level, to simulate the reactions and interchange of receptors and receptor-ligand complexes. At this level, each high density patch is abstracted into a single, well mixed compartment. Since receptors tend to diffuse quickly through the non-attractive region until they are [re]trapped by an attractive patch, the entire non-attractive region is represented as a single compartment. The mesoscopic model

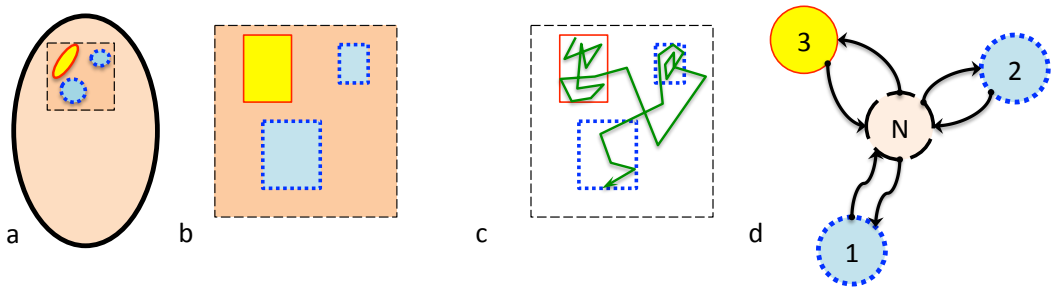


Figure 3: (a,b) Attractive microdomains occupy a small fraction of the cell membrane, and their measurements may be extracted from experimental images. (c) Although the receptors can move through the non-attractive regions, they tend to remain in the smaller clusters. (d) In the mesoscopic approach each microdomain, as well as the rest of the membrane (the "normal" region), are represented as well-mixed compartments that may exchange particles.

is an abstraction of the microscopic models, where *spatial degrees of freedom have been discretized*. Mobility information is encapsulated in the particle exchange rates between domains, the capacity of the corresponding compartments, as well as the effective dimerization rates with the compartments. In terms of implementation, this level requires the composition of a spatial network of domains, defined by an oriented, weighted adjacency graph, and a chemical reaction network.

3. The third, highest level of abstraction is obtained by (1) merging all attractive patches into a single one and (2) making the continuum approximation. The resulting *ODE system* is discussed in the remainder of this paper.

2 Model

Consistent with the emerging experimental picture, we make two assumptions. First, we assume that a fraction $f \leq 1$ of the membrane is covered by domains that have a physical affinity for receptors. As receptors diffuse throughout the membrane, the probability of crossing the boundary of such an attractive domain is asymmetric - all else being equal, inbound crossing is $\alpha \geq 1$ times more likely than outbound. We will consider the aggregate of the high affinity patches as a single *high density* domain, and refer to the rest as the low density domain or sector. Second, we will set the mobility of receptor dimers lower than that of monomers. These two ingredients result in the preferential accumulation of receptors in the high density patches. We investigate the effect of this accumulation (clustering) on dimerization and signal initiation. We are especially interested in establishing whether there is a postive feed-back between dimerization and receptor clustering.

2.1 Reactions and Equations

We follow the mathematical modeling framework of MacGabhann, Popel and coworkers [5] to describe free (R) and ligand-bound (VR) receptors, receptor dimers (RR), and three ligand-bound dimer complexes (VRR, RVR, Δ); the ligand is considered constant. Their structure and reactions among them are illustrated in Figure 1. For simplification, we assume that there is a region in the membrane with high affinity for VEGF receptors, and describe the rest as a second,"normal" or low affinity one. Each of the six species is presented in both domains; similary, each of the 7 reactions has a copy in each domain, see Figure 4. Assuming the free VEGF concentration is kept constant at V_0, we have a 12-dimensional state

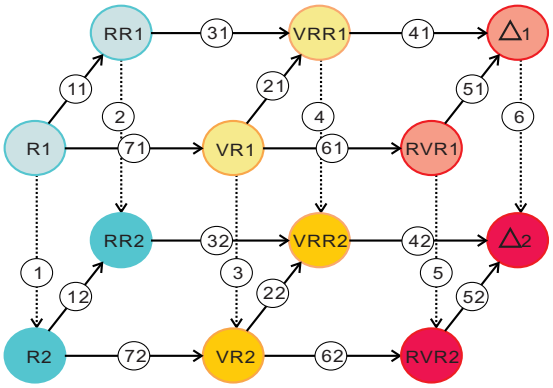


Figure 4: Reactions in the two-compartment model. Each horizontal sheet contains the reactions in one of the domains; transport reactions are "vertical". Here we omitted the added monomer receptors in reactions $\mathscr{C}_{1\mathrm{x}}$ and $\mathscr{C}_{2\mathrm{x}}$, as well as the added VEGF (V) in reactions $\mathscr{C}_{3\mathrm{x}}$ and $\mathscr{C}_{7\mathrm{x}}$.

vector,

$$\mathbf{X} = ([\mathrm{R}_1],[\mathrm{R}_2],[\mathrm{RR}_1],[\mathrm{RR}_2],[\mathrm{VR}_1],[\mathrm{VR}_2],[\mathrm{VRR}_1],[\mathrm{VRR}_2],[\mathrm{RVR}_1],[\mathrm{RVR}_2],[\Delta_1],[\Delta_2])^T \quad . \tag{1}$$

In addition to the 28 (irreversible) reactions that represent molecular transformations, we describe the transfer of every molecular species between domains as a separate reaction, bringing the total to 40 (irreversible) reactions. It is convenient to group pairs of opposing reactions into single reversible reactions [11], leaving us with 20 reversible reactions, as illustrated in Figure 4. The arrows represent the conventional direction for the corresponding fluxes. The 20 reactions with reaction rates assuming mass-action are denoted as follows (where $x = 1,2$):

$$\begin{array}{lll}
\mathscr{C}_{1\mathrm{x}}:\ \mathrm{R_x}+\mathrm{R_x}\underset{d}{\overset{b}{\rightleftharpoons}}\mathrm{RR_x} & \mathscr{C}_{2\mathrm{x}}:\ \mathrm{VR_x}+\mathrm{R_x}\underset{d}{\overset{b}{\rightleftharpoons}}\mathrm{VRR_x}; & \mathscr{C}_{3\mathrm{x}}:\ \mathrm{RR_x}+\mathrm{V_0}\underset{c}{\overset{2a}{\rightleftharpoons}}\mathrm{VRR_x};\\
\mathscr{C}_{4\mathrm{x}}:\ \mathrm{VRR_x}\underset{c_i}{\overset{a_i}{\rightleftharpoons}}\Delta_\mathrm{x}; & \mathscr{C}_{5\mathrm{x}}:\ \mathrm{RVR_x}\underset{d_i}{\overset{b_i}{\rightleftharpoons}}\Delta_\mathrm{x}; & \mathscr{C}_{6\mathrm{x}}:\ \mathrm{VR_x}+\mathrm{R_x}\underset{c}{\overset{a_s}{\rightleftharpoons}}\mathrm{RVR_x};\\
 & \mathscr{C}_{7\mathrm{x}}:\ \mathrm{R_x}+\mathrm{V_0}\underset{c}{\overset{a}{\rightleftharpoons}}\mathrm{VR_x}; & \\
\mathscr{D}_1:\ \mathrm{R}_1\underset{k_2}{\overset{k_1}{\rightleftharpoons}}\mathrm{R}_2; & \mathscr{D}_2:\ \mathrm{RR}_1\underset{\beta k_2}{\overset{\beta k_1}{\rightleftharpoons}}\mathrm{RR}_2; & \mathscr{D}_3:\ \mathrm{VR}_1\underset{k_2}{\overset{k_1}{\rightleftharpoons}}\mathrm{VR}_2;\\
\mathscr{D}_4:\ \mathrm{VRR}_1\underset{\beta k_2}{\overset{\beta k_1}{\rightleftharpoons}}\mathrm{VRR}_2; & \mathscr{D}_5:\ \mathrm{RVR}_1\underset{\beta k_2}{\overset{\beta k_1}{\rightleftharpoons}}\mathrm{RVR}_2; & \mathscr{D}_6:\ \Delta_1\underset{\beta k_2}{\overset{\beta k_1}{\rightleftharpoons}}\Delta_2.
\end{array} \tag{2}$$

The corresponding stoichiometry matrix is

$$\Gamma = \begin{bmatrix}
-2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0\\
0 & -2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0\\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}.$$

Effective concentrations: We will use *effective concentrations* to describe the amounts of each species found in the two domains; $[\mathrm{S_x}]^{\mathrm{eff}}$ is defined as the ratio of the amount (number of mols) of substance S

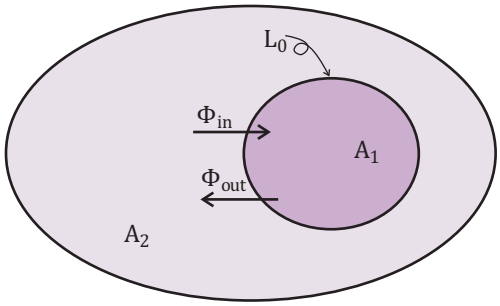


Figure 5: Schematic and notations for the high- and low-density areas on the cell surface. We assume that a fraction of the membrane (area A_1) has a higher affinity for VEGF receptors than the rest of the membrane. This translates into asymmetric rate 'constants' for the Φ_{in}, and Φ_{out} fluxes.

in domain x ($x = 1,2$), divided by the *total* area of the cell membrane A_{cell}. We will refer to the usual concentrations as *physical*, $[\text{S}_\text{x}]^{\text{phys}}$. Generally, the meaning of the concentrations and rate constants is similar to the standard approach in [5], with some important differences as discussed below.

Consider first the exchange reactions ($\mathscr{D}_1 \ldots \mathscr{D}_6$ in eq.(2)), exemplified by reaction $\mathscr{D}_1 : \text{R}_1 \rightleftharpoons \text{R}_2$. Let the fraction of the area that has high affinity to VEGF receptors be f. The size of the high (VEGF) density area is $A_1 = f \cdot A_{\text{cell}}$, and the remaining area is $A_2 = (1-f) \cdot A_{\text{cell}}$ (see Figure 5). Let us derive the flux of unbound receptors R between A_1 and A_2, represented by the reaction $\text{R}_1 \rightleftharpoons \text{R}_2$. Let $[\text{R}_1]^{\text{phys}}$ and $[\text{R}_2]^{\text{phys}}$ be the physical concentrations of R in A_1 and A_2, respectively, defined as the amount (in fmol) of R in A_1 (respectively A_2), divided by the area A_1 (resp. A_2). The effective concentrations, denoted as $[\text{R}_1]^{\text{eff}}$ ($[\text{R}_2]^{\text{eff}}$ resp.), are the amounts of R in A_1 (A_2 resp.), but divided by the *total* area A_{cell}. Therefore,

$$[\text{R}_1]^{\text{phys}} = \frac{[\text{R}_1]^{\text{eff}}}{f} \quad \text{and} \quad [\text{R}_2]^{\text{phys}} = \frac{[\text{R}_2]^{\text{eff}}}{(1-f)} \quad , \tag{3}$$

with units of fmol/cm^2 for all concentrations.

We assume that the *flux* of receptors, Φ_{out} (amount of substance per unit time and boundary length, in fmol/(cm $\cdot$ s) in our case), from A_1 to A_2 is proportional to the *physical* concentration, $\Phi_{\text{out}} = \gamma_{\text{out}}[\text{R}_1]^{\text{phys}}$; similarly, the receptor flux into A_1 is $\Phi_{\text{in}} = \gamma_{\text{in}}[\text{R}_2]^{\text{phys}}$. The factors γ_{in} and γ_{out} reflect the physical permeability of the boundary and have units of cm/s. We define the *attractiveness*, $\alpha \equiv \gamma_{\text{in}}/\gamma_{\text{out}}$, to reflect the asymmetry of the permeabilities; so we have $\Phi_{\text{in}} = \gamma_{\text{in}}[R_2]^{phys} = \alpha\gamma_{\text{out}}[R_2]^{phys}$, and $\alpha \geq 1$ means that a receptor or a dimer is more easily transferred into the high concentration area A_1 than into A_2. Consequently, the exchange fluxes between the two domains will balance when the ratio of the respective *physical* concentrations is α, i.e. $\Phi_{\text{in}} = \Phi_{\text{out}} \Leftrightarrow \alpha\gamma_{\text{out}}[\text{R}_2]^{\text{phys}} = \gamma_{\text{out}}[\text{R}_1]^{\text{phys}} \Leftrightarrow \alpha[\text{R}_2]^{\text{phys}} = [\text{R}_1]^{\text{phys}}$.

Consider the net rate of change of concentrations $[\text{R}_1]^{\text{phys}}$ and $[\text{R}_2]^{\text{phys}}$, due to the exchange of receptor monomers between the two compartments, we have

$$\left(\frac{d[\text{R}_1]^{\text{phys}}}{dt}\right)_\Phi = \frac{(\Phi_{\text{in}} - \Phi_{\text{out}}) \cdot L_0}{A_1} \;\; ; \;\; \left(\frac{d[\text{R}_2]^{\text{phys}}}{dt}\right)_\Phi = \frac{(\Phi_{\text{out}} - \Phi_{\text{in}}) \cdot L_0}{A_2}, \tag{4}$$

where L_0 is the length of the boundary between A_1 and A_2. Substitute A_1, A_2, Φ_1 and Φ_2 into (4):

$$\left(\frac{d[\text{R}_1]^{\text{phys}}}{dt}\right)_\Phi = (\alpha[\text{R}_2]^{\text{phys}} - [\text{R}_1]^{\text{phys}})\frac{L_0\gamma_{\text{out}}}{fA_{\text{cell}}} \;\; , \;\; \left(\frac{d[\text{R}_2]^{\text{phys}}}{dt}\right)_\Phi = ([\text{R}_1]^{\text{phys}} - \alpha[\text{R}_2]^{\text{phys}})\frac{L_0\gamma_{\text{out}}}{(1-f)A_{\text{cell}}}. \tag{5}$$

Defining a common time constant $\delta \equiv A_{\rm cell}/(L_0\gamma_{\rm out})$ in (5), we have

$$\left(\frac{d[\mathrm{R}_1]^{\rm phys}}{dt}\right)_\Phi = (\alpha[\mathrm{R}_2]^{\rm phys} - [\mathrm{R}_1]^{\rm phys})\frac{1}{f\delta} \quad , \quad \left(\frac{d[\mathrm{R}_2]^{\rm phys}}{dt}\right)_\Phi = ([\mathrm{R}_1]^{\rm phys} - \alpha[\mathrm{R}_2]^{\rm phys})\frac{1}{(1-f)\delta}. \quad (6)$$

Finally, substituting the effective concentrations from (3) yields

$$\begin{aligned}\left(\frac{d[\mathrm{R}_1]^{\rm eff}}{dt}\right)_\Phi &= \frac{\alpha}{\delta(1-f)}[\mathrm{R}_2]^{\rm eff} - \frac{1}{\delta f}[\mathrm{R}_1]^{\rm eff},\\ \left(\frac{d[\mathrm{R}_2]^{\rm eff}}{dt}\right)_\Phi &= -\frac{\alpha}{\delta(1-f)}[\mathrm{R}_2]^{\rm eff} + \frac{1}{\delta f}[\mathrm{R}_1]^{\rm eff}.\end{aligned}$$

The above result implies the identity $\frac{d[\mathrm{R}_1]^{\rm eff}}{dt} + \frac{d[\mathrm{R}_2]^{\rm eff}}{dt} = 0$, which reflects particle number conservation. This is the main advantage of using effective concentrations.

We follow the same line of reasoning for the other transfer reactions. The exit rate constants also reflect the generic mobility of particles; in a more detailed simulation, one could relate them to the diffusion constants and the permeability of the membranes. Here we will assume that exit rate constants are the same as above for ligand-bound monomer species VR_x. For dimer species, RR_x, VRR_x, RVR_x and Δ_x, we will use reduced exit rate constants, proportional to $\gamma_{\rm in}$ and $\gamma_{\rm out}$, and denote the coefficient as β. In summary, the six exchange fluxes are

$$\begin{aligned}&\phi_1 = k_1[\mathrm{R}_1]^{\rm eff} - k_2[\mathrm{R}_2]^{\rm eff}, && \phi_2 = \beta(k_1[\mathrm{RR}_1]^{\rm eff} - k_2[\mathrm{RR}_2]^{\rm eff}),\\ &\phi_3 = k_1[\mathrm{VR}_1]^{\rm eff} - k_2[\mathrm{VR}_2]^{\rm eff}, && \phi_4 = \beta(k_1[\mathrm{VRR}_1]^{\rm eff} - k_2[\mathrm{VRR}_2]^{\rm eff}),\\ &\phi_5 = \beta(k_1[\mathrm{RVR}_1]^{\rm eff} - k_2[\mathrm{RVR}_2]^{\rm eff}), && \phi_6 = \beta(k_1[\Delta_1]^{\rm eff} - k_2[\Delta_2]^{\rm eff}),\end{aligned}$$

where $k_1 = \frac{1}{\delta f}$ and $k_2 = \frac{\alpha}{\delta(1-f)}$.

Next, we consider the rates of chemical reactions, molecular transformations that take place within each area. As an example, consider a reaction in the high density area. For $\mathscr{C}_{21} : \mathsf{VR}_1 + \mathsf{R}_1 \underset{d}{\overset{b}{\rightleftharpoons}} \mathsf{VRR}_1$, we have

$$\begin{aligned}\frac{d[\mathrm{R}_1]^{\rm phys}}{dt} = \frac{[\mathrm{dVR}_1]^{\rm phys}}{dt} &= -b[\mathrm{R}_1]^{\rm phys}[\mathrm{VR}_1]^{\rm phys} + d[\mathrm{VRR}_1]^{\rm phys},\\ \frac{d[\mathrm{VRR}_1]^{\rm phys}}{dt} &= b[\mathrm{R}_1]^{\rm phys}[\mathrm{VR}_1]^{\rm phys} - d[\mathrm{VRR}_1]^{\rm phys}. \quad (7)\end{aligned}$$

Substituting $[\mathrm{R}_1]^{\rm phys} = [\mathrm{R}_1]^{\rm eff}/f$, $[\mathrm{VR}_1]^{\rm phys} = [\mathrm{VR}_1]^{\rm eff}/f$ and $[\mathrm{RVR}_1]^{\rm phys} = [\mathrm{RVR}_1]^{\rm eff}/f$, we have

$$\frac{d[\mathrm{VRR}_1]^{\rm eff}}{dt} = -\frac{d[\mathrm{R}_1]^{\rm eff}}{dt} = -\frac{d[\mathrm{VR}_1]^{\rm eff}}{dt} = \frac{b}{f}[\mathrm{R}_1]^{\rm eff}[\mathrm{VR}_1]^{\rm eff} - d[\mathrm{VRR}_1]^{\rm eff} \quad , \quad (8)$$

therefore the flux for reaction r_{21} is

$$\phi_{21} = \frac{b}{f}[\mathrm{R}_1]^{\rm eff}[\mathrm{VR}_1]^{\rm eff} - d[\mathrm{VRR}_1]^{\rm eff} \quad .$$

The only difference between the above rate law and the one in terms of physical concentrations (7), is that the dimerization rate constant is scaled by the relative size of the domain, $b \to b/f$. This reflects the effect of clustering on dimerization; if the same number of reacting molecules are forced into a

smaller space, their collision rate and implicitly, the absolute dimerization rate, will increase. Similar considerations give the following for the 14 reversible reactions:

$$\begin{array}{ll}
\phi_{11} = \frac{2b}{f}[\mathrm{R}_1]^{\mathrm{eff}2} - d_1[\mathrm{RR}_1]^{\mathrm{eff}}, & \phi_{12} = \frac{2b}{1-f}[\mathrm{R}_2]^{\mathrm{eff}2} - d[\mathrm{RR}_2]^{\mathrm{eff}} \\
\phi_{21} = \frac{b}{f}[\mathrm{R}_1]^{\mathrm{eff}}[\mathrm{VR}_1]^{\mathrm{eff}} - d[\mathrm{VRR}_1]^{\mathrm{eff}}, & \phi_{22} = \frac{b}{1-f}[\mathrm{R}_2]^{\mathrm{eff}}[\mathrm{VR}_2]^{\mathrm{eff}} - d_2[\mathrm{VRR}_2]^{\mathrm{eff}} \\
\phi_{31} = 2aV_0[\mathrm{RR}_1]^{\mathrm{eff}} - c[\mathrm{VRR}_1]^{\mathrm{eff}}, & \phi_{32} = 2aV_0[\mathrm{RR}_2]^{\mathrm{eff}} - c[\mathrm{VRR}_2]^{\mathrm{eff}} \\
\phi_{41} = a_i[\mathrm{VRR}_1]^{\mathrm{eff}} - 2c_i[\Delta_1]^{\mathrm{eff}}, & \phi_{42} = a_i[\mathrm{VRR}_2]^{\mathrm{eff}} - 2c_i[\Delta_2]^{\mathrm{eff}} \\
\phi_{51} = b_i[\mathrm{RVR}_1]^{\mathrm{eff}} - d_i[\Delta_1]^{\mathrm{eff}}, & \phi_{52} = b_i[\mathrm{RVR}_2]^{\mathrm{eff}} - d_i[\Delta_2]^{\mathrm{eff}} \\
\phi_{61} = \frac{a_s}{f}[\mathrm{R}_1]^{\mathrm{eff}}[\mathrm{VR}_1]^{\mathrm{eff}} - c[\mathrm{RVR}_1]^{\mathrm{eff}}, & \phi_{62} = \frac{a_s}{1-f}[\mathrm{R}_2]^{\mathrm{eff}}[\mathrm{VR}_2]^{\mathrm{eff}} - c[\mathrm{RVR}_2]^{\mathrm{eff}} \\
\phi_{71} = aV_0[\mathrm{R}_1]^{\mathrm{eff}} - c[\mathrm{VR}_1]^{\mathrm{eff}}, & \phi_{72} = aV_0[\mathrm{R}_2]^{\mathrm{eff}} - c[\mathrm{VR}_2]^{\mathrm{eff}} \ .
\end{array} \tag{9}$$

We denote by $\Phi(X) = (\phi_{11}, \phi_{12}, \phi_{21}, \cdots, \phi_{71}, \phi_{72}, \phi_1, \cdots, \phi_6)^T$, then the system of differential equation assuming mass-action is

$$\frac{d\mathbf{X}}{dt} = \Gamma \cdot \Phi(\mathbf{X}). \tag{10}$$

2.2 Parameter values

Receptor Diffusivity and Boundaries: For the diffusivity of VEGFR, we used the exit rate of $\gamma_{\mathrm{out}} = 8.23 \cdot 10^{-6}$ cm/s based on the expression given in [5]. The cells are assumed to have a surface area of $1000\mu\mathrm{m}^2$ [5]; assuming a spherical shape, the radius of a cell works out to approximately $r_{cell} = 8.9\ \mu$m, and the length of the high density area boundary is $L_0 = 2\pi\sqrt{r_{cell}^2 - \left(r_{cell} - \frac{1000f}{2\pi \cdot r_{cell}}\right)^2}\,\mu$m, where $f \equiv A_1/A_{\mathrm{cell}}$ is the relative fraction of the HD area. The L_0 and γ_{out} can be readily substituted into the definition $\delta = A_{\mathrm{cell}}/(L_0\gamma_{\mathrm{out}})$. The graph of δ as a function of f is shown in Figure 6. We can see that δ decreases faster in the beginning as f is increasing from 0 to 0.5.

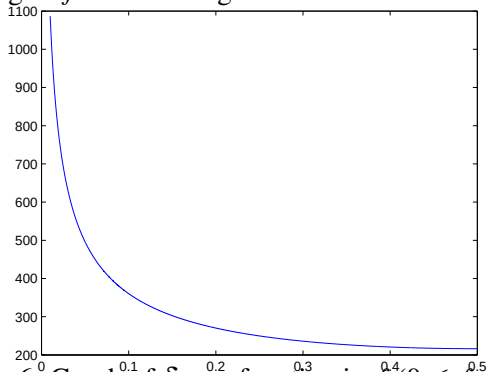


Figure 6: Graph of δ as a function in f ($0 \le f \le 0.5$).

Following [5], we assume that there are 40,000 receptor monomers on one cell membrane, corresponding to $R_{\mathrm{total}} \approx 6.6\ \mathrm{fmol} \cdot \mathrm{cm}^{-2}$.

Reaction Rates: We use the model of [5], with base units as follows: volume concentration (of VEGF only, V_0), in nM; all surface concentrations in $\mathrm{fmol/cm}^2$; time in s. Except for VEGF, whose concentration is *not* a variable, all molecular species in the model are surface bound. Consequently, the units of mass-action rate constants are s^{-1} for unimolecular reactions, $(\mathrm{nM} \cdot \mathrm{s})^{-1}$ for reactions involving VEGF, and $\mathrm{cm}^2/(\mathrm{fmol} \cdot \mathrm{s})$ for on-surface dimerizations. The rate constants are summarized in the table below.

Reaction	Direction	Notation	Value	Unit	Reaction	Direction	Notation	Value	Unit
$R_x + R_x \rightleftharpoons RR_x$	$\rightarrow$	b	0.1	$cm^2/(fmol \cdot s)$	$R_1 \rightleftharpoons R_2$	$\rightarrow$	k_1	0.0277	s^{-1}
	$\leftarrow$	d	0.01	s^{-1}		$\leftarrow$	k_2	0.0154	s^{-1}
$VR_x + R_x \rightleftharpoons VRR_x$	$\rightarrow$	b	0.1	$cm^2/(fmol \cdot s)$	$RR_1 \rightleftharpoons RR_2$	$\rightarrow$	βk_1	0.01385	s^{-1}
	$\leftarrow$	d	0.01	s^{-1}		$\leftarrow$	βk_2	0.0077	s^{-1}
$RR_x + V_0 \rightleftharpoons VRR_x$	$\rightarrow$	$2a$	0.0044	$(nM \cdot s)^{-1}$	$VR_1 \rightleftharpoons VR_2$	$\rightarrow$	k_1	0.0277	s^{-1}
	$\leftarrow$	c	0.026	s^{-1}		$\leftarrow$	k_2	0.0154	s^{-1}
$VRR_x \rightleftharpoons \Delta_x$	$\rightarrow$	a_i	0.949	s^{-1}	$VRR_1 \rightleftharpoons VRR_2$	$\rightarrow$	βk_1	0.01385	s^{-1}
	$\leftarrow$	c_i	0.026	s^{-1}		$\leftarrow$	βk_2	0.0077	s^{-1}
$RVR_x \rightleftharpoons \Delta_x$	$\rightarrow$	b_i	0.446	s^{-1}	$RVR_1 \rightleftharpoons RVR_2$	$\rightarrow$	βk_1	0.01385	s^{-1}
	$\leftarrow$	d_i	0.02	s^{-1}		$\leftarrow$	βk_2	0.0077	s^{-1}
$VR_x + R_x \rightleftharpoons RVR_x$	$\rightarrow$	a_s	0.21	$cm^2/(fmol \cdot s)$	$\Delta_1 \rightleftharpoons \Delta_2$	$\rightarrow$	βk_1	0.01385	s^{-1}
	$\leftarrow$	c	0.026	s^{-1}		$\leftarrow$	βk_2	0.0077	s^{-1}
$R_x + V_0 \rightleftharpoons VR_x$	$\rightarrow$	a	0.0044	$(nM \cdot s)^{-1}$	NOTE: the values for k_1 and k_2 given here correspond to				
	$\leftarrow$	c	0.026	s^{-1}	$f = 0.1$, $\alpha = 5$, $\beta = 0.5$.				

2.3 Steady States

To solve for the closed form steady-state of the differential equation system (10), we use the method introduced in [7]. The steady states of this system are sets of concentration values $\{[R_1],[R_2],[RR_1],[RR_2]$ $[VR_1],[VR_2],[VRR_1],[VRR_2],[RVR_1],[RVR_2],[\Delta_1],[\Delta_2]\}$ for which the expression on the right-hand side of (10) is identically zero. Define an expanded vector $\mathbf{X}_E$ that consists of the original variables of $\mathbf{X}$ plus the binomials $[R_1]^2$, $[R_2]^2$, $[R_1]\cdot[VR_1]$ and $[R_2]\cdot[VR_2]$ as

$$\mathbf{X}_E \equiv ([R_1],[R_2],[RR_1],[RR_2][VR_1],[VR_2],[VRR_1],[VRR_2],[RVR_1],[RVR_2], \\ [\Delta_1],[\Delta_2],[R_1]^2,[R_2]^2,[R_1]\cdot[VR_1],[R_2]\cdot[VR_2])^T \quad . \tag{11}$$

As the rate law vector $\Phi(\mathbf{X}) = (\phi_{11},\phi_{12},\phi_{21},\cdots,\phi_{71},\phi_{72},\phi_1,\cdots,\phi_6)^T$ is a linear combination of $\mathbf{X}_E$, it can be interpreted as a linear expression: $\Phi(\mathbf{X}) = \mathbf{A}_E \cdot \mathbf{X}_E$. So we have

$$\frac{d\mathbf{X}}{dt} = \Gamma \cdot \Phi(\mathbf{X}) = \Gamma \cdot \mathbf{A}_E \cdot \mathbf{X}_E = \bar{\mathbf{A}}_E \cdot \mathbf{X}_E,$$

where $\bar{\mathbf{A}}_E = \Gamma \cdot \mathbf{A}_E$ (the 12×16 dimensional expanded system matrix $\bar{\mathbf{A}}_E$ is too large to reproduce within normal text). We substitute $X_1 \equiv [R_1]^2$, $X_2 \equiv [R_2]^2$, $Y_1 \equiv [R_1]\cdot[VR_1]$ and $Y_2 \equiv [R_2]\cdot[VR_2]$ into $\mathbf{X}_E$, and denote the new vector as $\bar{\mathbf{X}}_E$. Then all elements in $\bar{\mathbf{X}}_E$ are linear variables, and the steady state problem is equivalent to that for a linear system $\frac{d\mathbf{X}}{dt} = \Gamma \cdot \Phi(\mathbf{X}) = \bar{\mathbf{A}}_E \cdot \bar{\mathbf{X}}_E$, find the set of $\bar{\mathbf{X}}_E$ such that

$$\frac{d\mathbf{X}}{dt} = 0 \;\Leftrightarrow\; \bar{\mathbf{A}}_E \cdot \bar{\mathbf{X}}_E = 0. \tag{12}$$

By Theorem 1 in [7], $\bar{\mathbf{A}}_E$ has the same rank as the original system, i.e. $rank(\bar{\mathbf{A}}_E) = rank(\Gamma) = 11$. For the linear equation system $\bar{\mathbf{A}}_E \cdot \bar{\mathbf{X}}_E = 0$, as $rank(\bar{\mathbf{A}}_E) = 11$, and there are 16 variables in $\bar{\mathbf{X}}_E$, we can solve 11 variables (dependent variables) as a function of the other 5 (free variables). To achieve that, we first discard a row of $\bar{\mathbf{A}}_E$, whose loss would not reduce the rank of $\bar{\mathbf{A}}_E$. In this case, we select row 1.

Next, we select the 11 dependent variables. The set of dependent variables has to be determined carefully to make the method given in [7] work. We select $\bar{\mathbf{X}}_D = ([RR_1],[RR_2],[VR_1],[VR_2],[VRR_1],[VRR_2],$ $[RVR_1],[RVR_2],[\Delta_1],[\Delta_2],Y_2)^T$ as the set of dependent variables, and $\bar{\mathbf{X}}_F = ([R_1],[R_2],X_1,X_2,Y_1)^T$ as the set of free variables. We use Cramer's Rule to solve for $\bar{\mathbf{X}}_D$ in terms of $\bar{\mathbf{X}}_F$. Denote the solution as

$$y_i = a_{i1}[R_1] + a_{i2}[R_2] + a_{i3}X_1 + a_{i4}X_2 + a_{i5}Y_1,$$

where $i = 1, 2, \cdots, 11$, $y_i \in X_D$ and a_{ij} are algebraic combinations of reaction rate constants. Substitute the bilinears $X_1 = [\mathrm{R}_1]^2$, $X_2 = [\mathrm{R}_2]^2$, $Y_1 = [\mathrm{R}_1][\mathrm{VR}_1]$ and $Y_2 = [\mathrm{R}_2][\mathrm{VR}_2]$ back to $\bar{\mathbf{X}}_D$ and $\bar{\mathbf{X}}_F$, then the solution can be rewritten as

$$y_i = a_{i1}[\mathrm{R}_1] + a_{i2}[\mathrm{R}_2] + a_{i3}[\mathrm{R}_1]^2 + a_{i4}[\mathrm{R}_2]^2 + a_{i5}[\mathrm{R}_1][\mathrm{VR}_1], \tag{13}$$

where $i = 1, 2, \cdots, 11$. We carefully select two solutions

$$[\mathrm{VR}_1] = a_{31}[\mathrm{R}_1] + a_{32}[\mathrm{R}_2] + a_{33}[\mathrm{R}_1]^2 + a_{34}[\mathrm{R}_2]^2 + a_{35}[\mathrm{R}_1][\mathrm{VR}_1] \tag{14}$$

$$[\mathrm{VR}_2] = a_{41}[\mathrm{R}_1] + a_{42}[\mathrm{R}_2] + a_{43}[\mathrm{R}_1]^2 + a_{44}[\mathrm{R}_2]^2 + a_{45}[\mathrm{R}_1][\mathrm{VR}_1]. \tag{15}$$

It is easy to solve (14) for $[\mathrm{VR}_1]$. Denote the solution as $[\mathrm{VR}_1] = \varphi_1([\mathrm{R}_1], [\mathrm{R}_2])$. We then substitute the solution of $[\mathrm{VR}_1]$ to (15) and let $[\mathrm{VR}_2] = \varphi_2([\mathrm{R}_1], [\mathrm{R}_2])$. As

$$y_2 = [\mathrm{R}_2][\mathrm{VR}_2] = a_{11,1}[\mathrm{R}_1] + a_{11,2}[\mathrm{R}_2] + a_{11,3}[\mathrm{R}_1]^2 + a_{11,4}[\mathrm{R}_2]^2 + a_{11,5}[\mathrm{R}_1][\mathrm{VR}_1], \tag{16}$$

we substitute $[\mathrm{VR}_1] = \varphi_1([\mathrm{R}_1], [\mathrm{R}_2])$ and $[\mathrm{VR}_2] = \varphi_2([\mathrm{R}_1], [\mathrm{R}_2])$ into (16), and this will reduce the variables of (16) to $[\mathrm{R}_1]$ and $[\mathrm{R}_2]$. The resulting identity is a cubic function that only has $[\mathrm{R}_1]$ and $[\mathrm{R}_2]$ as variables. By Cardano's method, we solve the cubic function symbolically for $[\mathrm{R}_2]$, and denote the only positive real root by $[\mathrm{R}_2] = \psi([\mathrm{R}_1])$. Substituting $[\mathrm{R}_2] = \psi([\mathrm{R}_1])$ to $[\mathrm{VR}_1]$ solution, we then have

$$[\mathrm{VR}_1] = \varphi_1([\mathrm{R}_1], [\mathrm{R}_2]) = \varphi_1([\mathrm{R}_1], \psi([\mathrm{R}_1])).$$

As $[\mathrm{R}_2]$ and $[\mathrm{VR}_1]$ are expressed as algebraic functions with variable $[\mathrm{R}_1]$, all the variables in $X_F = \{[\mathrm{R}_1], [\mathrm{R}_2], [\mathrm{R}_1]^2, [\mathrm{R}_2]^2, [\mathrm{R}_1][\mathrm{VR}_1]\}$ can be represented as an explicit function of $[\mathrm{R}_1]$, and consequently all the solutions of (13) can be rewritten as functions of $[\mathrm{R}_1]$. We expressed all 12 variables as functions of $[\mathrm{R}_1]$. With the conservation law

$$\begin{aligned} R_{total} = [\mathrm{R}_1] + [\mathrm{R}_2] + 2[\mathrm{RR}_1] + 2[\mathrm{RR}_2] + [\mathrm{VR}_1] + [\mathrm{VR}_2] + 2[\mathrm{VRR}_1] + 2[\mathrm{VRR}_2] \\ + 2[\mathrm{RVR}_1] + 2[\mathrm{RVR}_2] + 2[\Delta_1] + 2[\Delta_2] \ , \end{aligned} \tag{17}$$

we can solve the equation for $[\mathrm{R}_1]$ numerically for any given value of R_{total}. For all parameter values used in this paper, the dependence on $[\mathrm{R}]$ was consistent with a single real root, leading us to the conclusion that the system had a unique steady state. This does not exclude the possibility of multiple steady states for other parameter sets.

3 Results and Discussion

We obtained steady states for the differential equations (10) by numerically solving the steady state equations as outlined above, for various values of the relative size of the high density area ($f = 5\% \ldots 30\%$), the attractiveness parameter ($\alpha = 1 \ldots 10$), and for VEGF concentrations ranging from 0.01 nM to 5 nM. We considered three situations for the relative mobility of dimers $\beta = 0.5, 0.25, 0$ (note that the $\beta = 0.5$ case corresponds to equivalent monomer and dimer mobilities).

We first performed calculations using the full model of [5]. The equilibrium values for the total number of receptors and signaling complexes in the two domains, as a function of the three parameters (α, f, V_0), are shown in Figures 7 and 8. Not surprisingly, increasing the attractiveness parameter α,

relative size f of the HD area results in an increasing fraction of receptors and signaling complexes in the high density area. Increasing the concentration of VEGF leads to overall increased singaling but no significant shifts between the domains.

In Figure 7, the total amount of signaling complexes increases only weakly as a function of the attractiveness parameter α. This set calculations was performed including both the ligand-induced dimerization as well as the pre-dimerization (DPD) mechanism of the Mac Gabhann-Popel model. The DPD rate constant $b = 0.1$ results in a high degree of dimerization (more than 90% dimers), even in the absence of ligand or a high affinity domain.

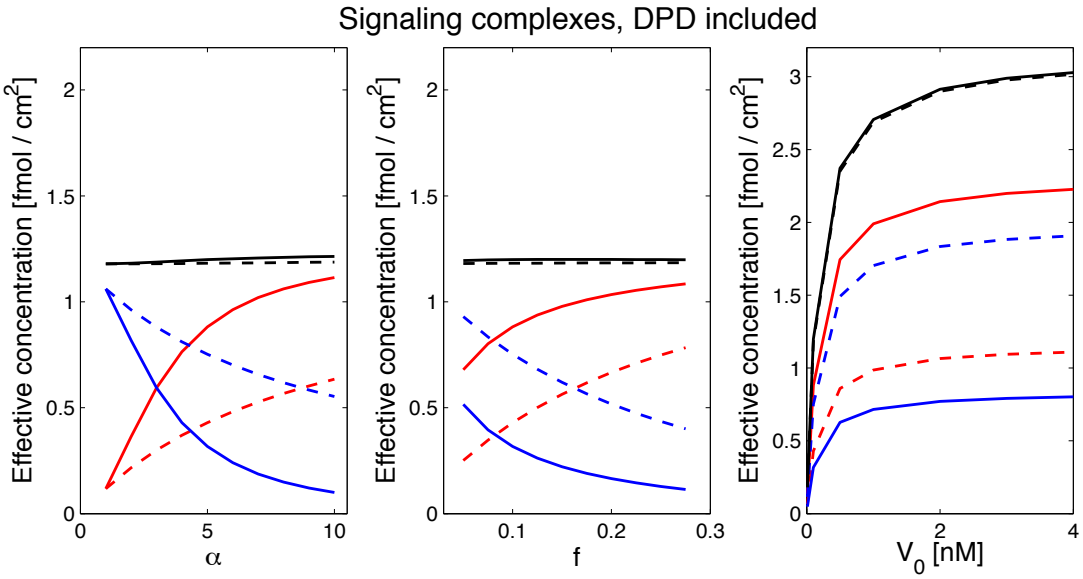


Figure 7: Signaling complexes (RVR and Δ) in the high-affinity (red) and normal (blue) domains, as well as total signal (black) as a function of the attractiveness factor α, the relative size of the HD domain f, and the concentration of VEGF ligand V_0. The values for the fixed parameters were $\alpha = 5, f = 0.1, V_0 = 0.1$ nM. Solid lines correspond to the case when dimers are not allowed to cross domain boundaries, and dashed lines correspond to fully mobile dimers. The total signal (black lines) depends weakly on the affinity and size of the attractive domain due to the combined effects of the relatively high ($V_0 = 0.1$ nM) VEGF concentration value used in the calculations, as well as due to the presence of strong ligand-independent dimerization (DPD) in the model.

We were especially interested in the effect of dimerization on the preferential accumulation of receptors. While the results in Fig. 7 and 8 indicate that the accumulation effect is stronger when dimers are not allowed to cross domain boundaries, the ligand dose response curves (rightmost panels) show only a marginal effect due to the presence of ligand.

The explanation for the weakness of these effects is the presence of ligand-independent dimerization (or pre-dimerization) in the Mac Gabhann-Popel model, as well as the high value of the on-surface VEGF-receptor binding constant a_S. The effect of dimerization on clustering is revealed when pre-dimerization is turned off (by setting $b = 0.0001$ in the rate laws) and the on-surface ligand binding rate a_s is reduced. These results are shown in Fig. 9 and 10. The signal is clearly increased as the affinity of the HD domain increases. There is an optimum in the size of the HD area (middle panel, Fig. 9). The effect on the signal persists when dimers are allowed to move. While the dependence of localization on attractiveness and domain size are similar to the previous set, the depedendence on V_0 is dramatically different in Fig. 10, showing a significant increase in the number of receptors in the HD area as V_0 is increased. This effect is completely absent when dimers are allowed to move at the same rate as monomers and is very significantly weakened at the intermediate mobility value we used (those results not shown).

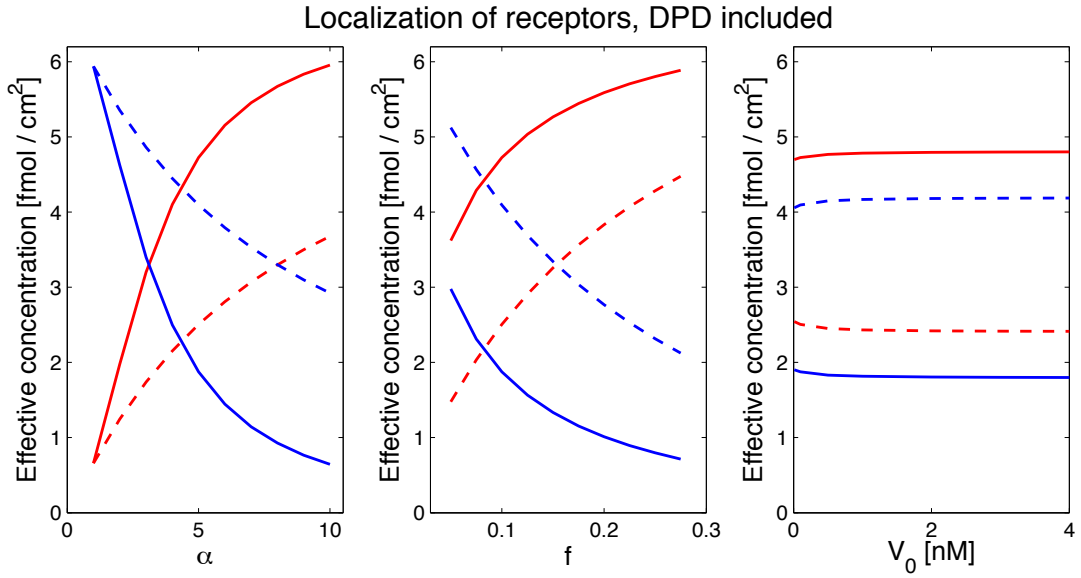


Figure 8: Total number of receptors in the high-affinity (red) and normal (blue) domains, in the same set of calculations as in Figure 7. The affinity and size of the HD domain strongly influence clustering (represented by the accumulation of receptors in the HD domain), however, the effect of VEGF is marginal, because the model includes *ligand-independent* dimerization (DPD).

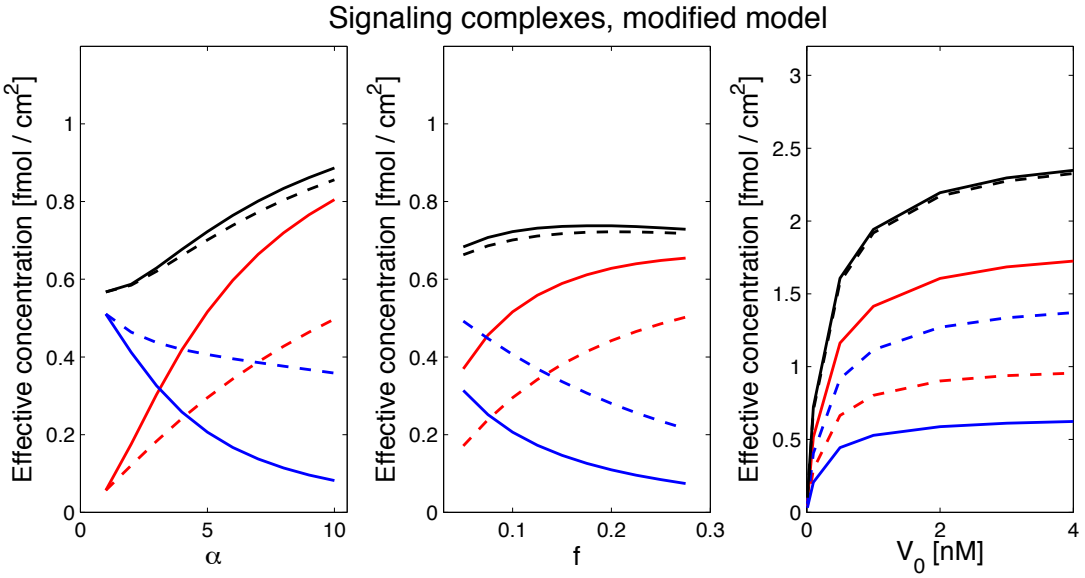


Figure 9: Signaling complexes (*RVR* and Δ) in the high-affinity (red) and normal (blue) domains, as well as total signal (black) as a function of the attractiveness factor α, the relative size of the HD domain f, and the concentration of VEGF ligand V_0. The values for the fixed parameters were $\alpha = 5, f = 0.1, V_0 = 0.1$ nM. Solid lines correspond to the case when dimers are not allowed to cross domain boundaries, and dashed lines correspond to fully mobile dimers. This calculation used significantly reduced *on-surface dimeriztion rates*, namely $a_s = 0.0021$ and $b = 0.0001$ (essentially eliminating DPD). By contrast with Figure 7, the affinity of the HD domain strongly enhances the signal, as dimers are formed at a higher rate in the HD domain.

In summary, our simple model shows a positive feedback between dimerization / signaling and receptor clustering. Accumulation of receptors in a high affinity patch enhances dimerization and signaling. On the other hand, increased dimerization, in the presence of ligand, inceased the accumulatin of receptors in the high affinity patch. The latter effect also requires a dramatic reduction in the mobility of dimers. Finally, we should point out that the empirically obtained model parameters lead to dimerization

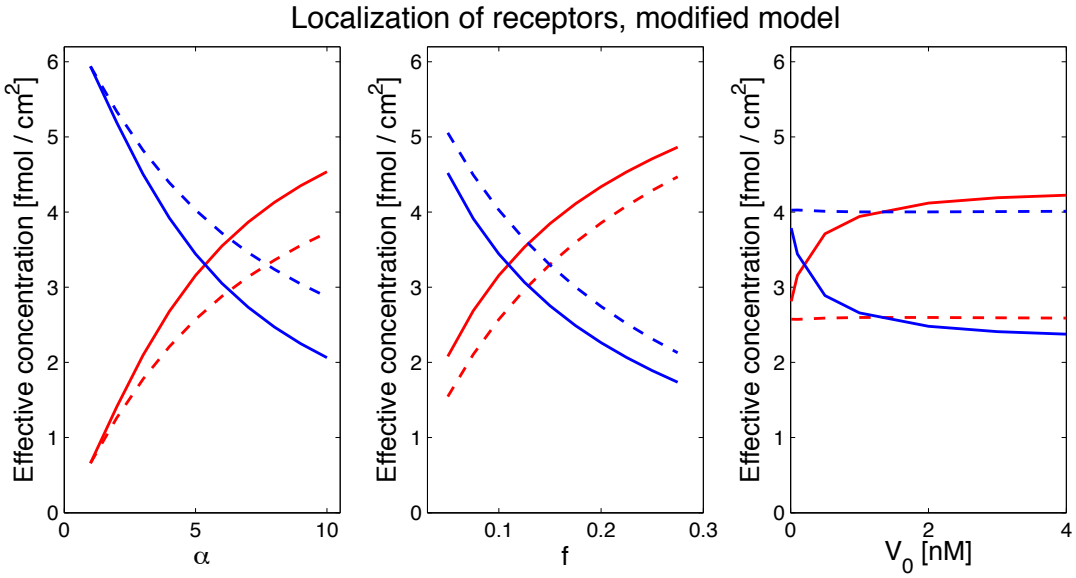


Figure 10: Total number of receptors in the high-affinity (red) and normal (blue) domains, in the same set of calculations as in Figure 9. By contrast with Figure 8, the presence of VEGF strongly promotes the accumulation of receptors in the HD domain. Dimers are formed at a higher rate in the HD domain, and become trapped, causing the HD domain to act as a receptor sink.

rates that are so high that the effects we described here would be marginal; however, the empirically determined parameters are what one would observe *as a result of* affinity-induced clustering. In other words, in the presence of high density domains, significanly lower dimerization rate constants may be sufficient to achieve the observed signaling.

References

[1] N. L. Andrews, K. A. Lidke, J. R. Pfeiffer, A. R. Burns & B. S. Wilson (2008): *Actin restricts FCeRI diffusion and facilitates antigen-induced receptor immobilization*. Nat Cell Biol 10, pp. 955–963, doi:10.1038/ncb1755.

[2] D. A. Birk, J. Barbato, L. Mureebe & R. A. Chaer (2010): *Current insights on the biology and clinical aspects of VEGF regulation*. Vasc Endovascular Surg 42, pp. 517–530, doi:10.1177/1538574408322755.

[3] M. N. Costa, K. Radhakrishnan, B. S. Wilson, D. G. Vlachos & J. S. Edwards (2009): *Coupled Stochastic Spatial and Non-Spatial Simulations of ErbB1 Signaling Pathways Demonstrate the Importance of Spatial Organization on Signal Transduction*. PLOS One 4(7), p. 6316, doi:10.1371/journal.pone.0006316.s001.

[4] N. Ferrara, K. J. Hilla, H. P. Gerber & W. Novotny (2004): *Discovery anddevelopment of bevazicumab, an anti-VEGF antibody for treating cancer*. Net Rev Drug Discov 3, pp. 391–400, doi:10.1038/nrd1381.

[5] F. Mac Gabhann & A. S. Popel (2007): *Dimerization of VEGF receptors and implications for signal transduction: A computational study*. Biophysical Chemistry 128, pp. 125–139, doi:10.1016/j.bpc.2007.03.010.

[6] A. M. Gallegos, S. M. Storey, A. B. Kier, F. Shroeder & J. M. Ball (2006): *Structure and cholesterol dynamics of caveloae/raft and nonraft plasma membrane domains*. Biochemistry 45, pp. 12100–12116, doi:10.1021/bi0602720.

[7] Á. M. Halász, H.-J. Lai, M. M. McCabe Pryor, K. Radhakrishnan & J. S. Edwards (2013): *Analytical Solution of Steady State Equations for Chemical Reaction Networks with Bilinear Rate Laws*. IEEE/ACM Transactions on Computational Biology and Bioinformatics, doi:10.1109/TCBB.2013.41. Epub before print.

[8] D. Hanahan & J. Folkman (1996): *Patterns and emerging mechanisms of the angiogenetic switch during tumorigenesis*. Cell 86, pp. 353–364, doi:10.1016/S0092-8674(00)80108-7.

[9] M.-Y. Hsieh, S. Yang, M. Raymond-Stinz, S. Steinberg, D. Vlachos, W. Shu, B. Wilson & J. S. Edwards (2008): *Stochastic Simulations of ErbB Homo- and Hetero-Dimerization: Potential IMpacts of Receptor Conformational State and Spatial Segregation*. IET Systems Biology 2, pp. 256–272, doi:10.1049/iet-syb:20070073.

[10] A. F. Karamysheva (2008): *Mechanisms of angiogenesis*. Biochemistry (Mosc) 73, pp. 751–762, doi:10.1134/S0006297908070031.

[11] L. B. Kleiman, T. Maiwald, H. Conzelman, D. A. Lauffenburger & P. K. Sorger (2011): *Rapid phospho-turnover by receptor tyrosine kinases impacts downstream signaling and drug binding*. Molecular Cell 43, p. 723C737, doi:10.1016/j.molcel.2011.07.014.

[12] M. A. Lemon & J. Schlessinger (2010): *Cell Signaling by Receptor Tyrosine Kinases*. Cell 141, pp. 1117–1134, doi:10.1016/j.cell.2010.06.011.

[13] B. F. Lillemeier, J. R. Pfeiffer, Z. Surviladze, B. S. Wilson & M. M. Davis (2006): *Plasma membrane-associated proteins are clustered into islands attached to the cytoskeleton*. Proc Natl Acad Sci USA 103, pp. 18992–18997, doi:10.1073/pnas.0609009103.

[14] M. Lohela, M. Bry, T. Tammela & K. Alitalo (2009): *VEGFs and receptors involved in angiogenesis versus lymphangiogenesis*. Curr Opin Cell Biol 21, pp. 154–165, doi:10.1016/j.ceb.2008.12.012.

[15] K. Mayawala, D. G. Vlachos & J. S. Edwards (2005): *Computational modeling reveals molecular details of epidermal growth factor binding*. BMC Cell Biol 6, p. 41, doi:10.1186/1471-2121-6-41.

[16] K. Mayawala, D. G. Vlachos & J. S. Edwards (2005): *Heterogeneities in EGF rceptor density at the cell surface can lead to concave up Scatchard plot of EGF binding*. FEBS Letters 579, pp. 3043–3047, doi:10.1016/j.febslet.2005.04.059.

[17] H. Ming-Yu, S. Yang, M. A. Raymond-Stinz, J. S. Edwards & B. S. Wilson (2010): *Spatiotemporal modeling of signalingprotein recruitment to EGFR*. BMC Systems Biology 4, p. 57.

[18] K. Murase, T. Fujiwara, Y. Umemura, K. Suzuki, R. Iino & H. Murakoshi (2004): *Ultrafine membrane compartments for molecular diffusion as revealed by single molecule techniques*. Biophys J 75, pp. 4075–4093, doi:10.1529/biophysj.103.035717.

[19] D. A. Olsson, J. Kreuger & L. Claesson-Welsh (2006): *VEGF receptor signaling - in control of vascular function*. Nat Rev Mol Cell Biol 7, pp. 359–371, doi:10.1038/nrm1911.

[20] G. Orr, D. Hu, S. Ozcelik, L. K. Opresko & H. S. Wiley (2005): *Cholesterol dictates the freedom of EGF receptors and HER2 in the plane of the membrane*. Biophys J 89, pp. 1362–1373, doi:10.1529/biophysj.104.056192.

[21] J. Plouet, J. Schilling & D. Gospodarowicz (1989): *Isolation and characterization of a newly identified endothelial cell mitogen produced by AtT-20 cels*. EMBO J 8, pp. 3801–3806.

[22] K. Ritchie & A. Kusumi (2003): *Single-particle tracking image microscopy*. Methods Enzymol 360, pp. 618–634, doi:10.1016/S0076-6879(03)60131-X.

[23] R. J. Roskoski (2008): *VEGF receptor protein-tyrosine kinases: structure and regulation*. Biochem Biophys Res Commun 375, pp. 287–291, doi:10.1016/j.bbrc.2008.07.121.

[24] F. Schoreder, A. M. Gallegos, B. P. Atshaves, S. M. Storey & A. L. McIntosh (2001): *Recent advances in membrane microdomains: rafts, caveolae, and intracellular cholesterol trafficking*. Exp Bio9l Med (Maywood) 226, pp. 873–890.

[25] D. R. Senger, S. J. Galli, A. M. Dvorak, C. A. Perruzzi, V. S. Harvey & H. F. Dvorak (1983): *Tumor cells secrete a vascular permeability factor that promotes accumulation of ascites fluid*. Science 219, pp. 983–985, doi:10.1126/science.6823562.

[26] S. J. Singer & G. L. Nicholson (1972): *The Fluid Mosaic Model of the Structure of Cell Membranes*. Science 175, pp. 720–731, doi:10.1126/science.175.4023.720.

[27] E. Stuttfield & K. Ballmer-Hofer (2009): *Structure and Function of VEGF Receptors*. *IUBMB Life* 61(9), pp. 915–922, doi:10.1002/iub.234.

[28] G. Vereb, J. Szollosi, J. Matko, P. Nagy, T. Farkas, L. Vigh, L. Matyus, T. A. Waldmann & S. Djanovich (2003): *Dynamic, yet structured: The cell membrane threedecades after the Singer-Nicolson model*. *Proc Natl Acad Sci USA* 100(14), pp. 8053–8058, doi:10.1073/pnas.1332550100.

[29] B. S. Wilson, J. R. Pfeiffer, M. A. Raymond-Stintz, D. Lidke & N. Andrews (2007): *Exploring membrane domains using native membrane sheets and transmission electron microscopy*. *Methods Mol Biol* 398, pp. 245–261, doi:10.1007/978-1-59745-513-8_17.

[30] S. Yang, M. A. Raymond-Stinz, W. Ying, J. Zhang, D. S. Lidke, S. L. Steinberg, L. Williams, J. M. Oliver & B. S. Wilson (2007): *Mapping ErbB rceptors on breast cancer cell membranes during signal transduction*. *J Cell Sci* 120, pp. 2763–2773, doi:10.1242/jcs.007658.

Falsifying Oscillation Properties of Parametric Biological Models

Thao Dang
CNRS-VERIMAG, Grenoble, France
Thao.Dang@imag.fr

Tommaso Dreossi
VERIMAG, University Joseph Fourier, Grenoble, France
Dept. of Mathematics and Computer Science
University of Udine, Udine, Italy
Tommaso.Dreossi@imag.fr

We propose an approach to falsification of oscillation properties of parametric biological models, based on the recently developed techniques for testing continuous and hybrid systems. In this approach, an oscillation property can be specified using a hybrid automaton, which is then used to guide the exploration in the state and input spaces to search for the behaviors that do not satisfy the property. We illustrate the approach with the Laub-Loomis model for spontaneous oscillations during the aggregation stage of Dictyostelium.

1 Introduction

Understanding periodic responses in living organisms is an important problem since such oscillations are a common phenomenon in biology. To reveal possible molecular mechanisms underlying this phenomenon, mathematical models have been developed. These models require validation before they can be used to make predictions. Such validation is often based on a comparison between the model behavior and experimental data obtained by temporal measurements. One major difficulty in biological model validation is that biological models often require many parameters, and most parameter values are neither measurable nor available in literature. Since there are often many sets of parameter values that can match the data, parameter identification is based not only on the error between the model simulation output and the data, but also on model robustness with respect to parameter variation. From a modelling point of view, robust parameters allow the model to fit new data without compromising the fit to the previous data. From a biological point of view, with robust parameters the system is resilient to perturbations.

The focus of this work is twofold. On one hand, we are interested in studying biological oscillating behaviors. On the other hand, we want to study the influence of parameters on the system behavior, that is how much the parameters can be varied without violating a given property. Typical behavioral changes include self-oscillations (that is the developments of periodical orbits from an equilibrium) and the occurence of a bifurcation. To illustrate this, we consider a dynamical system described by the following differential equations:

$$\dot{x} = f(x,k)$$

where $x \in \mathbb{R}^n$ is the state variables, and $k \in \mathbb{R}$ is a real-valued parameter. In a more general context, the dynamics of the system can be hybrid and contain more than one parameter. To characterize the impact of parameter variation, we want to know under which condition the two systems $\dot{x} = f(x,k)$ and $\dot{x} = f(x,k')$ (under two different parameter values k and k') are qualitatively similar, that is there exists an inversible

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 53–67, doi:10.4204/EPTCS.125.4

and continuous homeomorphism that maps a trajectory of one system to a trajectory of the other. In this case, an oscillating trajectory and a steady state equilibrium of one system correspond respectively to an oscillating trajectory and a steady state equilibrium of the other. When the parameter changes and reaches a value at which the behaviors are no longer qualitatively similar, such a behavioral change is often called a bifurcation and this value is called a critical value or a value of bifurcation. This can be illustrated with a linear dynamical system when the real parts of its eigenvalues change their sign under a parameter variation.

A set of parameter values is called *robust* if the system does not undergo a bifurcation under any variation of the parameter within that set. In this paper we propose to use a testing approach to analyze the robustness of biological models with respect to preservation of oscillation properties under admissible parameter variations. When applied to a model, this testing approach can be seen as systematic simulation that can check whether the model can replicate some essential behaviors observed during experiments. However, in general it can also be applied to a biological system (viewed as a black-box system). The key steps of the approach we propose are the following:

1. *Specifying the property*. A hybrid automaton $\mathcal{A}$ is used to describe the expected oscillating behaviors. We call $\mathcal{A}$ a *property automaton*. This automaton also encodes the satisfaction/violation of the property and incorporates realistic variations of the parameter values.

2. *Generating test cases for property falsification.* The generation of test cases from the automaton $\mathcal{A}$ is randomized but guided by the property, that is it favors the exploration of the trajectories leading to a violation of the property of interest.

We choose hybrid automata as specification formalism for two reasons. First, numerous phenomena in biology exhibit switching behaviors. Second, hybrid automata can naturally describe transitions between different qualitative behaviors, as we will show later. In the hybrid systems research, formal specification of oscillation properties of biological systems are considered in [3, 6].

Concerning bifurcation detection, the theory of bifurcation in smooth systems is well developed. The existing methods (such as using analysis of the eigenvalues of the Jacobian matrix [10], Routh-Hurwitz stability criteria [8, 9], the Floquet multipliers [14]) are developed for continuous systems and it is not easy to extend them to hybrid systems with discontinuities in the dynamics. Another approach (such as [7]) involves first generating the model outputs by simulation and then finding the parameters by fitting the simulation outputs to the experimental data, based on a grid over the parameter space. Our testing-based approach with a property guided search enables a quick detection without exploring a large number of parameter values. In addition, the approach has the potential to be more scalable than analytical and grid-based methods.

The rest of the paper is organized as follows. We first describe how to use hybrid automata to specify oscillation properties. This specification formalism can be applied to a large class of temporal properties due to the expressiveness of hybrid automata. We then show how to generate test cases from a property automaton for falsification purposes. The approach is applied to analyze the robustness of the Laub-Loomis model under parameter variation. This model has been proposed for describing the dynamical behavior of the molecular network underlying adenosine 3'5'-cyclic monophosphate (cAMP) [11].

2 Using Hybrid Automata to Specify Oscillation Properties

We first present a commonly used definition of hybrid automata and then show how they can be used to for oscillating property specifications.

2.1 Hybrid Automata

In the development of formal models for designing engineering systems, hybrid automata [1] emerged as an extension of timed automata [2] with more general dynamics. Unlike in a timed automaton where a clock c is a continuous variable with time derivative equal to 1, that is $\dot{c} = 1$, in a hybrid automaton its continuous variables x can evolve according to some differential equations, for example $\dot{x} = f(x)$. This allows hybrid automata to capture the evolution of a wide range of physical entities.

Definition 1 (Hybrid automaton)**.** *A* hybrid automaton *is a tuple* $\mathcal{A} = (\mathcal{X}, Q, E, F, \mathcal{I}, \mathcal{G}, (q_0, x_0))$ *where*

- $\mathcal{X} \subseteq \mathbb{R}^n$ *is the continuous state space;*
- Q *is a finite set of locations (or discrete states);*
- $E \subseteq Q \times Q$ *is a set of discrete transitions;*
- $F = \{F_q \mid q \in Q\}$ *specifies for each location a* continuous vector field*. In each location* $q \in Q$ *the evolution of the continuous variables* x *are governed by a differential equation* $\dot{x}(t) = f_q(x(t), u(t))$ *where* $u(\cdot) \in \mathcal{U}_q$ *is an input function of the form* $u : \mathbb{R}^+ \to U_q \subset \mathbb{R}^m$*. The set* $\mathcal{U}_q$ *is the set admissible inputs and consists of piecewise continuous functions. We assume that all* f_q *are Lipschitz continuous;*
- $\mathcal{I} = \{\mathcal{I}_q \subseteq \mathcal{X} \mid q \in Q\}$ *is a set of invariants. The* invariant *of a location* q *is defined as a subset* $\mathcal{I}_q$ *of* $\mathcal{X}$*. The system can evolve inside* q *if* $x \in \mathcal{I}_q$*;*
- $\mathcal{G} = \{\mathcal{G}_e \mid e \in E\}$ *is a set of* guards *specifying the conditions for switching between locations. For each discrete transition* $e = (q, q') \in E$*,* $\mathcal{G}_e \subseteq \mathcal{I}_q$*;*
- $\mathcal{R} = \{\mathcal{R}_e \mid e \in E\}$ *is a set of* reset *maps. Each transition* $e = (q, q') \in E$ *is associated with a reset map* $\mathcal{R}_e : \mathcal{G}_e \to 2^{\mathcal{I}_{q'}}$ *that defines how* x *may change when the automaton* $\mathcal{A}$ *switches from* q *to* q'*;*
- *The initial state of the automaton is denoted by* (q_0, x_0).

A state (q, x) of $\mathcal{A}$ can change in the following two ways:

1. by a *continuous evolution*, where the continuous state x evolves according to the dynamics f_q while the location q remains constant;
2. by a *discrete evolution*, where x satisfies the guard of an outgoing transition and the system changes location by taking this transition and updating the values of x accordingly to the associated reset map.

It is important to note that hybrid automata allow modelling non-determinism in both continuous and discrete evolutions. This non-determinism is useful for describing disturbance from the environment or under-specified control, as well as for taking into account imprecision in modeling.

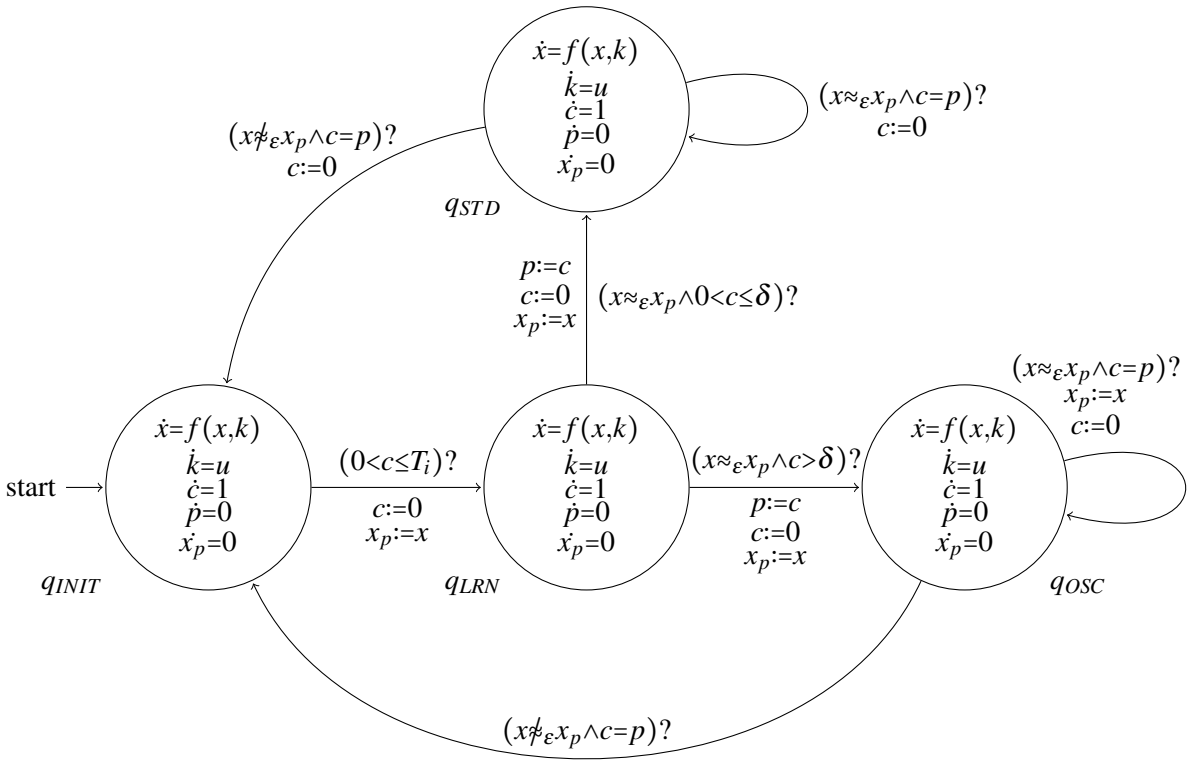


Figure 1: An oscillation property automaton.

2.2 Property Automata

We now show how to formalize some common temporal properties of particular interest for biological systems using hybrid automata. We will call them *property automata*.

A dynamical system starting from a given initial state can evolve to a steady state or to an irregular behavior. The steady state may be stationary (that is, the system remains in the same state as time passes), which is also called an equilibrium. The system can also evolve to a periodic state (or a limit cycle). Stationary states and periodic states can be stable (that is, attracting neighboring trajectories), unstable (that is, repelling neighboring trajectories), or non-stable (saddle). The stationary and periodical states are important since they help determine the long-term behavior of the system. It is often of great interest, in particular for biological systems, to know how the stationary and periodic states change when the parameters of the system change.

Suppose that we are interested in checking whether a given dynamical system exhibits the following behavioral pattern: the system from a given initial state evolves to a limit cycle and then under a some admissible parameter perturbation it evolves only from one limit cycle to another one. In other words, this parameter perturbation does not make the system undergo a structural behavior change (or a bifurcation). Another question is to know under which parameter changes the system moves from an oscillating behavior to a steady state.

The hybrid automaton depicted in Figure 1 can be used to specify the above-described oscillating behaviors. In this property automaton, the parameters $k \in \mathbb{R}^m$ form part of the continuous state of the automaton. In the remainder of this paper, we assume that the evolution of x can be described by:

$$\dot{x} = f(x,k)$$
$$\dot{k} = u$$

where u is the input. This can be thought of as an abstract view of the dynamics of x, which can be described using complex concrete models, such as a hybrid automaton. Additionally, as we will show later, for test generation purposes, the continuous dynamics in the property automaton is abstracted away, which results in a discrete abstraction. This abstraction retains only important information about the expected temporal behavioral patterns of the variables under study.

In this example, we restrict the derivatives of the parameters to be constant and they can take values in some set U. Therefore, this allows capturing piecewise linear evolution of the parameters. It is also worth noting that one can use other classes of functions to describe the parameter change.

In addition, to describe the desired temporal behavioral pattern, we augment the continuous state of the property automaton with three special variables: c, p and x_p, where c is a clock used to measure time lapses, p is used to store the oscillation period, and x_p is used to memorize a point to which the system should return after a period.

The discrete structure of the property automaton consists of four locations q_{INIT}, q_{LRN}, q_{OSC} and q_{STD}. The location q_{INIT} corresponds to transient behaviors (between different qualitative behaviors) that can have a maximal duration T_i. After this amount T_i of transient time, the automaton jumps to q_{LRN} and while doing so, as specified by the associated reset map, it stores the current value of x in the variable x_p which is used as an expected periodic point.

The role of the location q_{LRN} is to "learn" the period of a limit cycle that the system is expected to enter. At location q_{LRN}, if after a strictly positive time δ the system returns to the point x_p, then the automaton resets the clock c after storing its value in the variable p. We use a strictly positive amount of time lapse here to exclude Zeno behaviors. Therefore, if the system has entered a limit cycle, the value of the variable p is exactly the period of that limit cycle (see Figure 2). In case the system reaches x_p after exactly δ time, the automaton switches to the location q_{STD} which is used to model a steady state (see Figure 3). Note that the test generation algorithm interacts with the system under test in discrete time, and the value of δ represents the smallest clock period that the test generation algorithm can handle.

After the learning phase at the location q_{LRN} the variable p contains the value of the expected period. When the automaton switches from the location q_{LRN} to the location q_{OSC}, the variable x_p is updated with the current value of x. At the location q_{OSC}, the automaton checks after every p time whether x returns to the periodic point x_p. There are two cases:

- If x is in the ε-neighborhood of the periodic point x_p, the system is considered oscillating and the clock is reset and the self-loop transition is traversed in order to check the next oscillation cycle.
- Otherwise, the automaton jumps back to the initial location q_{INIT}. This models the scenario where the system leaves the current limit cycle and may then evolve to another limit cycle.

To allow measurement imprecision, in the guard conditions x is not required to return exactly to x_p but to some ε-neighborhood of x_p. This is denoted by $x \approx_\varepsilon x_p$.

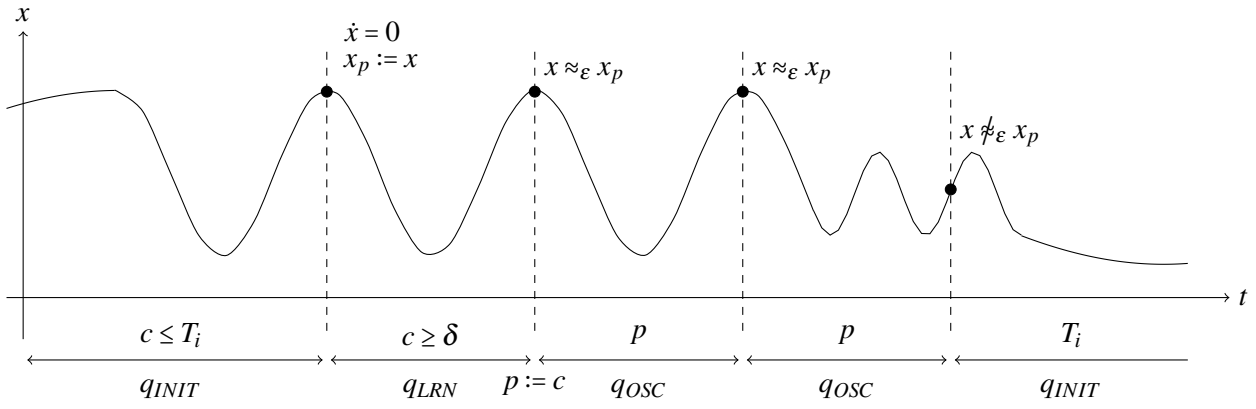


Figure 2: Detection of an oscillation using a periodic point x_p.

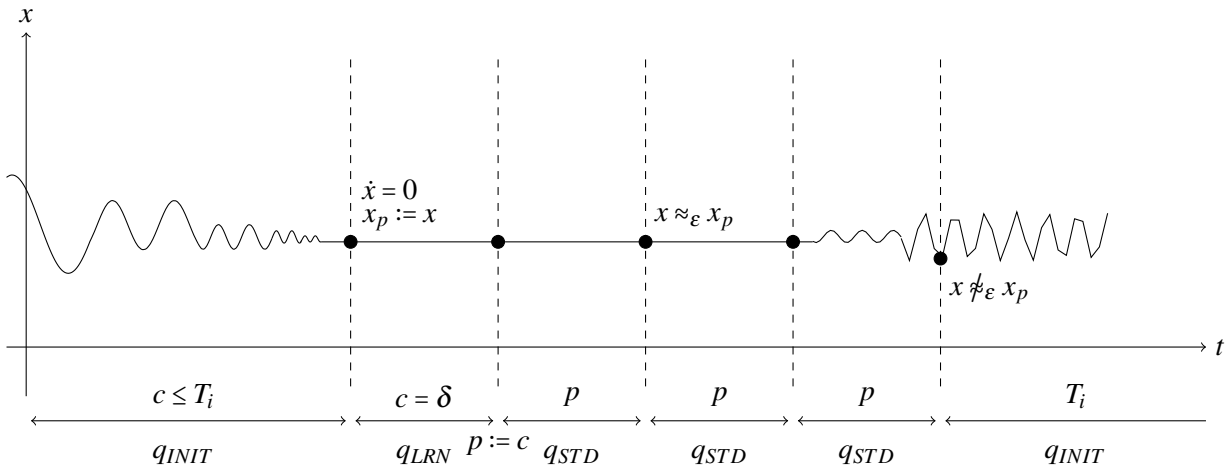


Figure 3: Detection of a steady state.

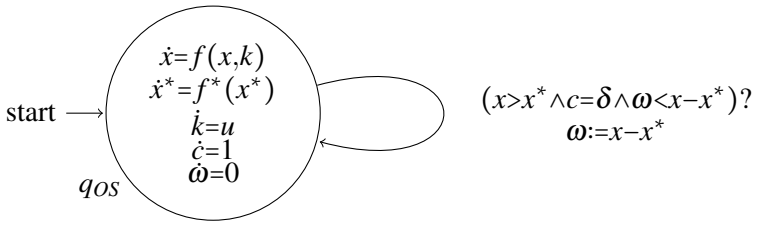


Figure 4: Overshoot detection.

Another property that can be easily expressed using hybrid automata is the maximal *overshoot*, that is the maximum peak value of x measured from a desired response of the system. The automaton has only one location q_{OS} and is equipped with an auxiliary variable ω which stores the maximum distance from the desired response x^* (see Figure 4).

3 Guided Exploration to Falsify a Property

Given a property automaton, our problem now is to explore the parameter space to detect behaviors that do not satisfy the property expressed by this automaton. To do so, we make use of the test generation algorithm gRRT [5]. This algorithm is based on the star discrepancy coverage notion and allows achieving good coverage of the reachable state space. When the objective is not to cover the whole reachable space but to quickly detect some specific behavioral patterns, we can use on top of the gRRT algorithm a property-based guiding tool. The goal of this tool is to specify some critical regions to visit and then the algorithm gRRT can be used to cover those regions. Before continuing, let us briefly recall the algorithm gRRT.

Given a hybrid automaton $\mathcal{A}$, the algorithm gRRT generates a *test case* represented by a tree where each node is associated with a state of $\mathcal{A}$ and each arc is associated with a control input action, which is either a continuous input value or a discrete control action (that is, the action of traversing a transition). Note that in the context of this work, both continuous inputs (described by u in the definition of hybrid automata) and transitions are controllable by the tester. To execute such a test case, the tester applies a control input sequence to the system, measures the variables of interest and decides whether the system under test satisfy the property. The algorithm thus can be thought as a procedure to find input signals that correspond to the beahviors we want to observe. The main steps of the coverage-guided test generation algorithm gRRT [5] are the following:

- Step 1: a goal state (q_{goal}, x_{goal}) is sampled from the state space;
- Step 2: a neighbor state (q_{near}, x_{near}) of the goal state is determined;
- Step 3: from the neighbor state, an appropriate continuous input u is applied for a time step h, or a transition is taken, in order to steer the system towards the goal state.

Step 2 can be done using a notion of distance between two hybrid states that capture the effects of discrete transitions. The choice of continuous input u in Step 3 can be done by a random selection from a discretization of the input set U. Indeed, more sophisticated methods based on trajectory sensitivity to input variation can be used but they cost more computation effort. It is important to note that a good

selection of goal states is key to a good coverage result, because the success of randomized algorithms depends on finding good starting states. For a more thorough description of the algorithm gRTT and its properties, the reader is referred to [5]. For the example in Section 4 a uniform randomized selection of u is used, which already allows an efficient exploration.

To bias the goal state sampling while taking into account the property to falsify, we first construct a discrete abstraction of the property automaton $\mathcal{A}$ that reflects the expected behavioral patterns. This abstraction is then used to biased the goal state samplings, so that it favors the exploration of the behavioral patterns of interest. As an example, to falsify the oscillation property presented in the previous section, that is to show that after a given initial transient time there exists a parameter change that leads the system out of an expected limit cycle, the trajectories that lead to the transition from q_{OSC} to q_{INIT} are favored. In other words, the exploration is biased in a way to increase the probability of sampling the goal states in the guard set of this transition.

To define a discrete abstraction, we need some additional definitions. A n-dimensional predicate is defined as $\pi(x) := g(x) \sim 0$ where $g : \mathbb{R}^n \to \mathbb{R}$ is a function of n variables, and $\sim \in \{\geq, >\}$. Let λ be a function that specifies for each location $q \in Q$ a vector of m_q predicates, that is $\lambda(q) = (\pi_1, \ldots, \pi_{m_q})$.

We define for each location q an abstraction function $\alpha_q : \mathcal{X} \to \mathbb{B}^{m_q}$ such that

$$\alpha_q(x) = (\pi_1(x), \ldots, \pi_{m_q}(x)).$$

We say that the Boolean abstraction vector of x with respect to α_q is the Boolean vector $(\pi_1(x), \ldots, \pi_{m_q}(x))$. The abstraction function α_q associated with a location $q \in Q$ partitions the set of continuous states at location q into at most 2^{m_q} subsets of continuous states such that all the continuous states in each subset have the same Boolean abstraction vector with respect to the abstraction function α_q.

In the other direction, for each location q we define the concretization function $\gamma_q : \mathbb{B}^{m_q} \to 2^{\mathcal{X}}$ such that for a given Boolean vector $b \in \mathbb{B}^{m_q}$, $\gamma_q(b) = \{x \in \mathcal{X} \mid \alpha_q(x) = b\}$.

The discrete abstraction of $\mathcal{A}$ with respect to λ is a transition system $\mathcal{D} = \{S, \rightsquigarrow, s_0\}$.

- Each location q of the hybrid automaton corresponds to a set S_q of abstract states, each of which corresponds to a pair (q, b) where $b \in \mathbb{B}^{m_q}$ is a value of the Boolean abstraction vector. For convenience, we call them q-abstract states. Two q-abstract states $s = (q, b)$ and $s' = (q, b')$ are adjacent if their corresponding sets of concrete states, that is $\gamma_q(b)$ and $\gamma_q(b')$, have non-empty intersection and they intersect only their boundaries. The whole abstract state space S is the union

$$S = \bigcup_{q \in Q} S_q.$$

- The transition relation $\rightsquigarrow \subset S \times S$ between the abstract states is defined as the union of the following two relations $\rightsquigarrow_d$ and $\rightsquigarrow_c$. Let $s = (q, b)$ and $s' = (q', b')$ be two abstract states; the transition relation between them is defined as follows:
  - $s \rightsquigarrow_c s'$ if $q = q'$ and s_1 and s_2 are adjacent.
  - $s \rightsquigarrow_d s'$ if $q \neq q'$ and $\gamma_q(s) \cap \mathcal{G}_{qq'} \neq \varnothing$ and $\mathcal{R}(\gamma_q(s) \cap \mathcal{G}_{qq'}) \subseteq \mathcal{I}_{q'}$.

  The relation $\rightsquigarrow_d$ represents the transitions in the abstract state space due to discrete switches in the original hybrid automaton $\mathcal{A}$, the relation $\rightsquigarrow_c$ represents the continuous evolution in $\mathcal{A}$.
- The initial abstract state $s_0 = (q_0, b_0)$ where $b_0 = \alpha_{q_0}(x_0)$.

The abstraction $\mathcal{D}$ can be thought of as an over-approximation of $\mathcal{A}$ since it is easy to see that any execution of $\mathcal{A}$ corresponds to an execution of $\mathcal{D}$. Moreover, it can be refined based on the exploration results in order to distinguish different qualitative behaviors that are important with respect to the property to validate.

In order for such a discrete abstraction to reflect the behavioral patterns we want to explore, we should choose for each location a set of predicates that can capture the discrete transitions of $\mathcal{A}$ and separate critical regions from the rest; therefore the set should include the predicates defining the guard and invariant conditions. This will be illustrated by the example in Section 4.

To biased the search, we use the Metropolis-Hastings method to perform a random walk [12] on $\mathcal{D}$ starting at the abstract state s_0. We first specify a target probability distribution over the abstract states

$$\pi = \{\pi_s \mid s \in S\}.$$

We then construct the following transition matrix $P(\mathcal{D})$. Between two abstract states s and s', we assign a probability to the transition from s to s':

$$\begin{cases} p_{ss'} = \dfrac{1}{deg(s)} min\{\dfrac{deg(s)\pi_{s'}}{deg(s')\pi_s}, 1\} & \text{if } s \rightsquigarrow s' \\ p_{ss'} = 1 - \sum\limits_{w \neq s} p_{sw} & \text{if } s = s' \\ p_{ss'} = 0 & \text{otherwise} \end{cases}$$

The above transition matrix $P(\mathcal{D})$ guarantees that the stationary distribution of the resulting random walk on the abstraction $\mathcal{D}$ is the target distribution π [13]. Therefore the abstract states corresponding to the region we want to visit are assigned with high target probabilities.

The Metropolis-Hastings method was proved to have good hitting times, which allows quickly reaching a desired abstract state, indeed the hitting time from s to s' of this random walk is of $\mathcal{O}(rN_v^2)$ where N_v is the number of abstract states and $r = \max\{\dfrac{\pi_s}{\pi_{s'}} \mid s, s' \in \rightsquigarrow\}$.

4 Application

4.1 Laub-Loomis Model

In this section we apply on the Laub-Loomis model [11] the techniques previously exposed. The model consists of a parametrized ODE system extracted from a molecular network that describes the aggregation stage of Dictyostelium. Our main intent is to show that for some parameter variation with bounded derivatives, the spontaneous oscillations of the system do not occur any more. Roughly speaking, we want to falsify the oscillation robustness of the system.

To this end, we derive a discrete abstraction from the property automaton in Figure 1 and guide the simulation of the ODE system towards the areas in the state space of the property automaton where the oscillation disappears. The derivatives u of the parameter variables are the inputs that we use to guide the exploration.

A revisited model that slightly differs from the original one presented by Laub and Loomis [11] is the following [8]:

$$\dot{x} = f(x,k) = \begin{bmatrix} k_1x_7 - k_2x_1x_2 \\ k_3x_5 - k_4x_2 \\ k_5x_7 - k_6x_2x_3 \\ k_7 - k_8x_3x_4 \\ k_9x_1 - k_{10}x_4x_5 \\ k_{11}x_1 - k_{12}x_6 \\ k_{13}x_6 - k_{14}x_7 \end{bmatrix}$$

Par.	Val.	Par.	Val.
k_1	2.0 min^{-1}	k_8	1.3 min^{-1}
k_2	0.9 min^{-1}	k_9	0.3 min^{-1}
k_3	2.5 min^{-1}	k_{10}	0.8 $min^{-1}\mu M^{-1}$
k_4	1.5 min^{-1}	k_{11}	0.7 min^{-1}
k_5	0.6 min^{-1}	k_{12}	4.9 min^{-1}
k_6	0.8 $min^{-1}\mu M^{-1}$	k_{13}	23.0 min^{-1}
k_7	1.0 $min^{-1}\mu M^{-1}$	k_{14}	4.5 $min^{-1}\mu M^{-1}$

Table 1: Oscillations parameter values.

The variables x correspond to seven protein concentrations: $x_1 = [ACA]$, $x_2 = [PKA]$, $x_3 = [ERK2]$, $x_4 = [REGA]$, $x_5 = [Internal\ cAMP]$, $x_6 = [External\ cAMP]$ and $x_7 = [CAR1]$. The coefficient vector $k = [k_1, \ldots, k_{14}]$ contains the system parameters. Table 1 shows the parameter values for which spontaneous oscillations occur [11].

4.2 Constructing a Discrete Abstraction

In the property automaton in Figure 1, the transition from q_{OSC} to the location q_{INIT} is critical since it takes the system from an oscillation phase to a non-oscillation phase. We thus want to control the system's behavior in order to satisfy the condition $(x \not\approx_\varepsilon x_p)?$.

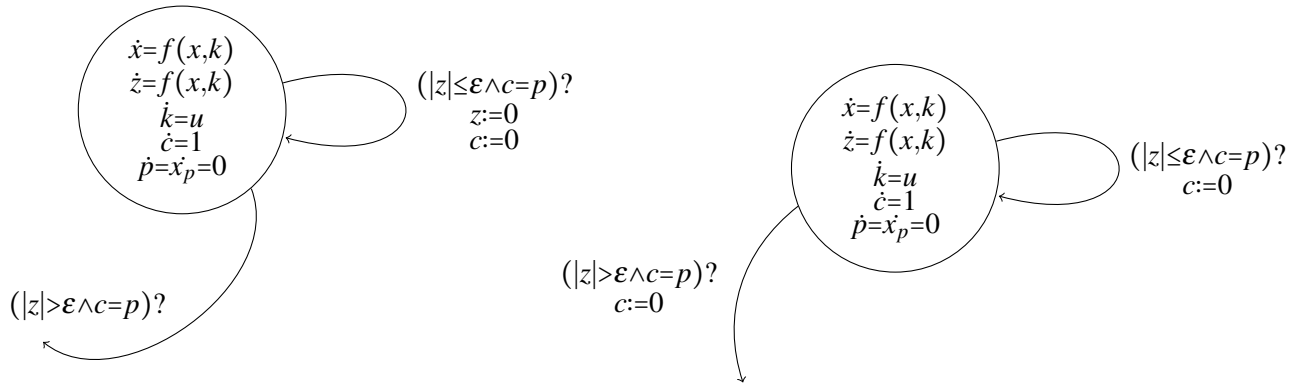
In addition, we modify the property automaton so that it results in an abstraction with predicates involving only one state variable, which is more suitable for the algorithm gRRT. Indeed the star discrepancy is defined for states inside some rectangular sets; for more general sets, box approximations are required. To do so, we modify the condition $(x \approx_\varepsilon x_p)?$ by introducing a new variable $z = x - x_p$, the derivative of which is $\dot{z} = \dot{x} - \dot{x_p} = \dot{x}$ (recall that by definition $\dot{x_p} = 0$). The guard on the self-loop transition over q_{OSC} becomes $(x \approx_\varepsilon x_p)? \equiv (|x - (x - z)| < \varepsilon)? \equiv (|z| \leq \varepsilon)?$, while the reset is rewritten as $(x_p := x) \equiv (x - z := x) \equiv (z := 0)$. Similarly the guard condition of the transition from q_{OSC} to q_{INIT} becomes $(|z| > \varepsilon)?$ and $z := 0$, respectively (see Figure 5a). The guard conditions and resets concerning the clock c remain unchanged. The same reasoning can be easily applied to the location q_{STD} (see Figure 5b).

We now proceed with the definition of the function λ which is the basis of the abstraction. Let $\lambda : Q \to 2^\Pi$ be defined as follows:

$$\lambda(q) = \begin{cases} (z \geq \varepsilon, \varepsilon > z, z > -\varepsilon, -\varepsilon \geq z) & \text{if } q = q_{OSC}, \\ (\top) & \text{otherwise.} \end{cases}$$

Note that abstraction function partitions the space of z in q_{OSC} into the sets $(+\infty; +\varepsilon]$, $(+\varepsilon, -\varepsilon)$ and $[-\varepsilon, -\infty)$ with the respective Boolean abstraction vectors $(1,0,0,0)$, $(0,1,1,0)$ and $(0,0,0,1)$. From λ we obtain the transition system $\mathcal{D} = \{S, \rightsquigarrow, s_0\}$, with abstract states

$$S = \{s_i = (q_i, \top) \mid q_i \in Q . q_i \neq q_{OSC}\} \cup \{s'_{OSC} = (q_{OSC}, (1,0,0,0)); s''_{OSC} = (q_{OSC}, (0,1,1,0)); s'''_{OSC} = (q_{OSC}, (0,0,0,1))\},$$



(a) q_{OSC} modified. (b) q_{STD} modified.

Figure 5: Modified property automaton.

transition relation

$$\rightsquigarrow = \rightsquigarrow_C \cup \rightsquigarrow_D =$$
$$\{(s'_{OSC}, s''_{OSC}); (s''_{OSC}, s'_{OSC}); (s''_{OSC}, s'''_{OSC}); (s'''_{OSC}, s''_{OSC})\} \cup$$
$$\{((q,b),(q',b')) \mid (q,q') \in E.q, q' \neq q_{OSC}\} \cup$$
$$\{(s'_{OSC}, s_{INIT}); (s''_{OSC}, s''_{OSC}); (s'''_{OSC}, s_{INIT})\},$$

and the initial abstract state $s_0 = s_{INIT} = (q_{INIT}, \top)$.

Before specifying the target probabilities over the abstract states, it is necessary to make another modification to the abstract transition system, in order to be able to distinguish the self-loop transitions originated from the abstraction process from those introduced by the transition probability definition. This modification consists in duplicating the locations that with self-loop transitions and replacing these self-loop transitions with the transitions connecting the original location to its copy, and vice versa. Hence, for this example we add two locations s_{STD^L} and s''_{OSC^L} to S and we replace in $\rightsquigarrow$ the transitions (s_{STD}, s_{STD}) and (s''_{OSC}, s''_{OSC}) with (s_{STD}, s_{STD^L}), (s_{STD^L}, s_{STD}), (s''_{OSC}, s''_{OSC^L}) and (s''_{OSC^L}, s''_{OSC}). Figure 6 shows the resulting actraction without such self-loop transitions.

We now define the target probabilities over the abstract states. Since we are interested in detecting that the system stops oscillating, it makes sense to attribute higher probabilities to those abstract states which bring the system from an oscillation phase to a non-oscillation one, i.e., the states s'_{OSC} and s'''_{OSC}. Thus, defining the target probabilities as $\pi_{s_{INIT}} = \pi_{s_{LRN}} = \pi_{s_{STD}} = \pi_{s_{STD^L}} = \pi_{s''_{OSC}} = \pi_{s''_{OSC^L}} = 0.1$ and $\pi_{s'_{OSC}} = \pi_{s'''_{OSC}} = 0.25$, we obtain the following probability transition matrix:

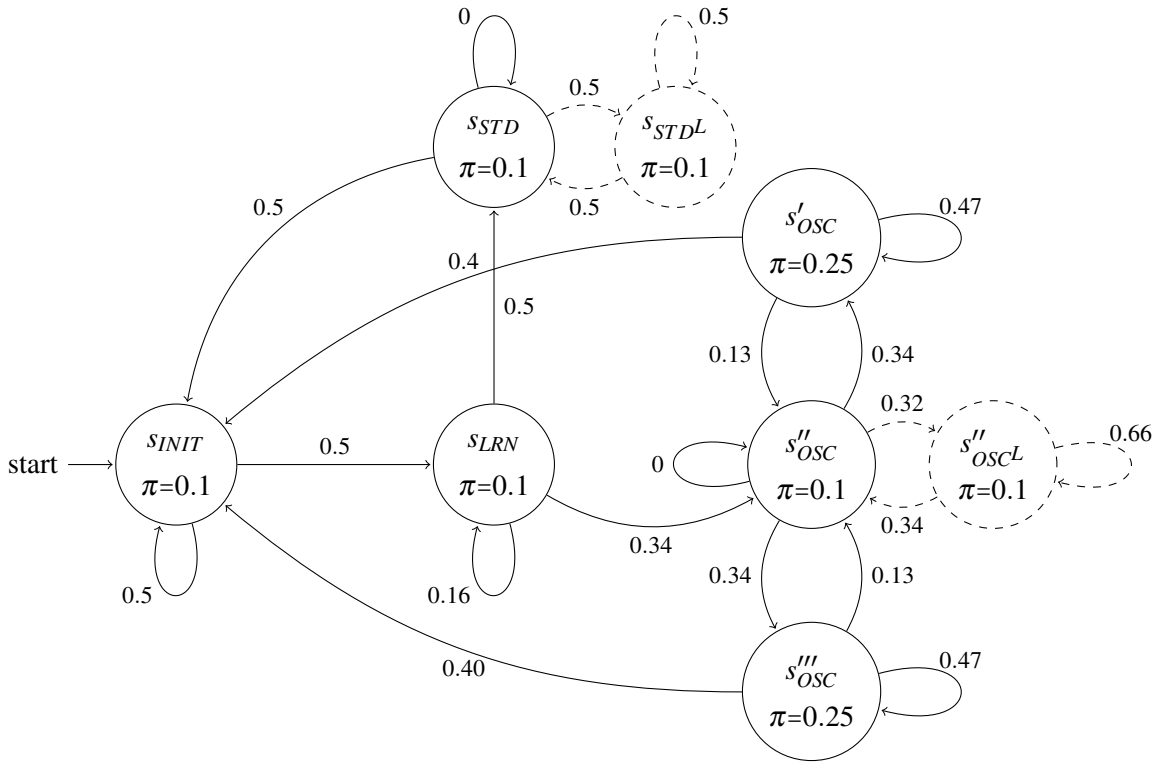


Figure 6: Abstract transition system of the property automaton. Dashed states and transitions are introduced to eliminate self-loop transitions in the property automaton.

	s_{INIT}	s_{LRN}	s_{STD}	s_{STD^L}	s'_{OSC}	s''_{OSC}	s''_{OSC^L}	s'''_{OSC}
s_{INIT}	0.50	0.50	0	0	0	0	0	0
s_{LRN}	0	0.16	0.50	0	0	0.34	0	0
s_{STD}	0.50	0	0	0.50	0	0	0	0
s_{STD^L}	0	0	0.50	0.50	0	0	0	0
s'_{OSC}	0.40	0	0	0	0.47	0.13	0	0
s''_{OSC}	0	0	0	0	0.34	0	0.32	0.34
s''_{OSC^L}	0	0	0	0	0	0.34	0.66	0
s'''_{OSC}	0.40	0	0	0	0	0.13	0	0.47

that leads to the system shown in Figure 6.

4.3 Experimental Results

We have implemented the above described method and incorporated it in the HTG tool [4], which is our previous C++ implementation of the gRRT algorithm. In particular, we extended HTG with the following new functions: defining a discrete abstraction over the considered hybrid automaton, specifying the target probabilities for each abstract state and performing a random walk on the abstract transition system in order to identify the areas that need to be explored.

In our experiments, we focused on the parameter k_1 and on its derivative modelled by the input variable u_1. Moreover, we monitor the two variables x_1 and z_1 since k_1 is involved in both of their dynamics. The values of the other parameters of the automaton in Figure 1 are fixed as follows: T_i = 7.3781, $\delta = 0.05$ and $\varepsilon = 0.2$. As an initial value of k_1 we choose its oscillating nominal value 2.0 (see Table 1).

We performed three experiments with different ranges of the input u_1. In the first case u_1 can be sampled within the interval $[-0.01, 0.01]$ (see Figure 7), in the second within $[-0.1, 0.1]$ (see Figure 8), while in the third within [-1.0,1.0]. In all the experiments the state space of k_1 is $[1.8, 2.2]$. In the first case, even if at the end of each period the value of z_1 is not exactly equal to zero, it is always included in the interval defined by $\varepsilon = 0.2$ and thus, for all the simulation runs, the system is considered oscillating. Differently, in the case where k_1 can evolve faster, the variable z_1 ends an oscillation phase at a value smaller than $-\varepsilon$. This means that already for values of $k_1 \in [1.8, 2.2]$ and $\dot{k_1} = u_1 \in [-0.1, 0.1]$ the system leaves the current limit cycle. We can interpret such a behavioral change under a small variation of the nominal parameter values as weak robustness of the Laub-Loomis model. Finally, for values of $u_1 \in [-1.0, 1.0]$ we found that, not very surprisingly, the variable z_1 drifts very far away from zero, showing that the system has stopped oscillating.

All the experiments were performed on a MacBook 3, 1 having 2GB RAM. Each experiment involved the computation of 30000 points, with integration time step equal to 0.05. In the first experiment, the tool required 6.23s, in the second 5.94s and in the third 6.53s. To give an idea of the scalability of the technique, a simulation with 10000 points requires 1.22s, with 25000 points 4.93s, while with 50000 points 17.42s.

5 Conclusion

In this paper, we described a framework for falsifying oscillation properties and study the robustness of biological models. The experimental results are encouraging and we intend to pursue this work in two

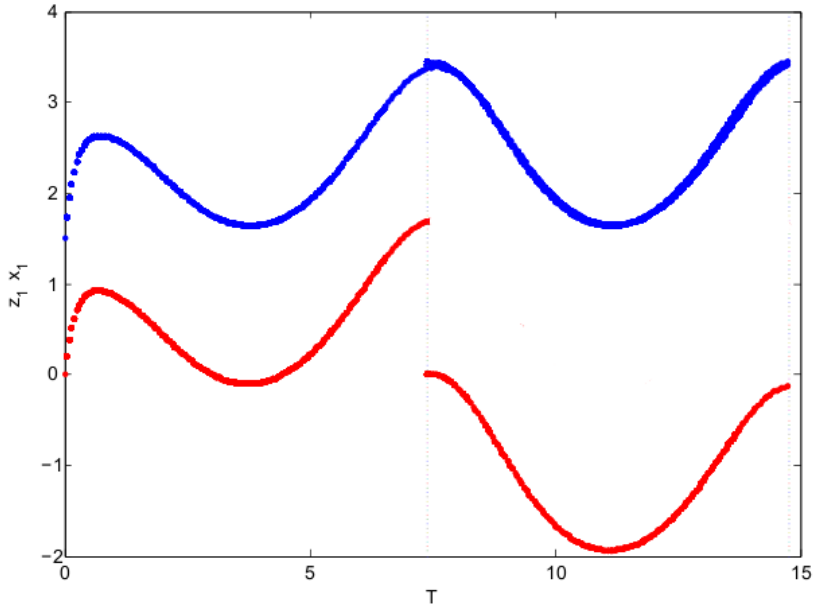


Figure 7: Evolutions of x_1 (blue) and z_1 (red) for $u_1 \in [-0.01, 0.01]$

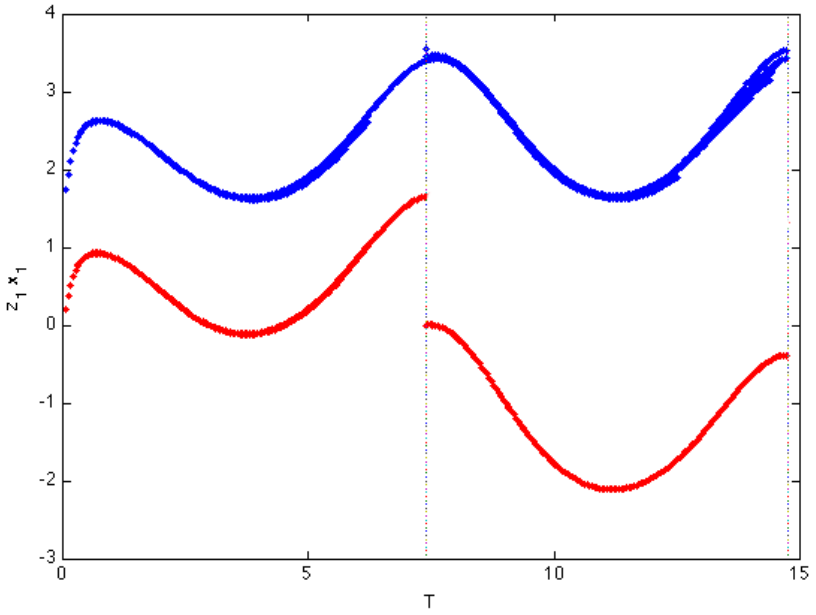


Figure 8: Evolutions of x_1 (blue) and z_1 (red) for $u_1 \in [-0.1, 0.1]$

directions. One is a more efficient parameter sampling, which can be guided by local analysis using Floquet theory. The other direction concerns the application of this approach to analyze other types of bifurcation in biological systems.

References

[1] R. Alur, C. Courcoubetis, T. A. Henzinger & P. H. Ho (1992): *Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems*. In R. L. Grossman, A. Nerode, A. P. Ravn & H. Richel, editors: *Hybrid Systems*, LNCS, Springer, pp. 209–229, doi:10.1007/3-540-57318-6_30.

[2] R. Alur & D. L. Dill (1994): *A Theory of Timed Automata*. *Theoret. Comput. Sci.* 126(2), pp. 183–235, doi:10.1016/0304-3975(94)90010-8.

[3] E. Bartocci, F. Corradini, E. Merelli & L. Tesei (2009): *Model Checking Biological Oscillators*. *Electronic Notes in Theoretical Computer Science* 299(1), pp. 41–58, doi:10.1016/j.entcs.2009.02.004.

[4] T. Dang (2010): *Model-based testing of hybrid systems*. Model-Based Testing for Embedded Systems, CRC Press, doi:10.1201/b11321-15.

[5] T. Dang & T. Nahhal (2009): *Coverage-guided test generation for continuous and hybrid systems*. *Form. Methods Syst. Des.* 34(2), pp. 183–213, doi:10.1007/s10703-009-0066-0.

[6] P. Dluhos, L. Brim & D. Safrnek (2012): *On Expressing and Monitoring Oscillatory Dynamics*. In: *HSB 2012*, doi:10.4204/EPTCS.92.6.

[7] M. M. Donahue, G. Buzzard & A. E. Rundell (2009): *Robust parameter identification with adaptive sparse grid-based optimization for nonlinear systems biology models*. In: *ACC Conference*, doi:10.1109/ACC.2009.5160512.

[8] R. Ghaemi & D. Del Vecchio (2007): *Evaluating the robustness of a biochemical network model*. In: *Decision and Control, 2007 46th IEEE Conference on*, pp. 615–620, doi:10.1109/CDC.2007.4434348.

[9] R. Ghaemi, J. Sun, P. A. Iglesias & D. Del Vecchio (2009): *A Method for determining the robustness of bio-molecular oscillator models*. *BMC Systems Biology* 3(95), doi:10.1186/1752-0509-3-95.

[10] J. Kim, D. G. Bates, I. Postlethwaite, L. Ma & P. A. Iglesias (2006): *Robustness analysis of biochemical network models*. *IEE Proc. Systems Biology* 153(2), pp. 96–104, doi:10.1049/ip-syb:20050024.

[11] M.T. Laub & W.F. Loomis (1998): *A molecular network that produces spontaneous oscillations in excitable cells of Dictyostelium*. *Molecular biology of the cell* 9(12), pp. 3521–3532, doi:10.1091/mbc.9.12.3521.

[12] R. Motwani & P. Raghavan (1995): *Randomized algorithms*. Cambridge University Press, New York, NY, USA, doi:10.1017/CBO9780511814075.

[13] Y. Nonaka, H. Ono, K. Sadakane & M. Yamashita (2010): *The hitting and cover times of Metropolis walks*. *Theoret. Comput. Sci.* 411(1618), pp. 1889 – 1894, doi:10.1016/j.tcs.2010.01.032.

[14] Kuznetsov Y. (2004): *Elements of Applied Bifurcation Theory* . Springer.

A hybrid mammalian cell cycle model

Vincent Noël
Université de Rennes 1
vincent.noel.pro@gmail.com

Sergey Vakulenko
Saint Petersburg State University of Technology and Design
vakulenfr@mail.ru

Ovidiu Radulescu
Université de Montpellier 2
ovidiu.radulescu@univ-montp2.fr

Hybrid modeling provides an effective solution to cope with multiple time scales dynamics in systems biology. Among the applications of this method, one of the most important is the cell cycle regulation. The machinery of the cell cycle, leading to cell division and proliferation, combines slow growth, spatio-temporal re-organisation of the cell, and rapid changes of regulatory proteins concentrations induced by post-translational modifications. The advancement through the cell cycle comprises a well defined sequence of stages, separated by checkpoint transitions. The combination of continuous and discrete changes justifies hybrid modelling approaches to cell cycle dynamics. We present a piecewise-smooth version of a mammalian cell cycle model, obtained by hybridization from a smooth biochemical model. The approximate hybridization scheme, leading to simplified reaction rates and binary event location functions, is based on learning from a training set of trajectories of the smooth model. We discuss several learning strategies for the parameters of the hybrid model.

1 Introduction

Systems biology employs a large number of formalisms to represent the dynamics of biochemically interacting molecules in signal transduction, metabolic and gene regulatory networks. Some of these formalisms, such as the systems of ordinary differential equations (ODE), are based on continuous representations of the phase space, whereas others, such as boolean networks, employ discrete dynamical variables. New approaches, based on hybrid models and using both continuous and discrete variables, are emerging as alternative descriptions of biochemical networks.

Hybrid modelling allows a good compromise between realistic description of mechanisms of regulation and the possibility of testing the model in terms of state reachability and temporal logics [12, 13]. Threshold dynamics of gene regulatory networks [2, 21] or of excitable signaling systems [24] has been modelled by piecewise-linear and piecewise-affine models. These models have relatively simple structure and can, in certain cases, be identified from data [19, 8]. Some methods were proposed for computing the set of reachable states of piecewise affine models [3].

The use of hybrid models in systems biology is justified when some events, such as rapid protein modifications occur on very short time scales and produce significant changes of the systems dynamics. The regulatory machinery of the cell cycle of eukaryotic organisms provides a remarkable example of such a situation. Indeed, the advancement through the cell cycle consists of a well defined sequence of stages, separated by checkpoint transitions. During each one of this stages, different sets of dynamical variables and biochemical reactions are specifically active, and change from one stage to another. A hybrid model of mammalian cell cycle has been previously proposed by Tyson's group [23]. This model is based on a Boolean automaton whose discrete transitions trigger changes of kinetic parameters in a

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 68–83, doi:10.4204/EPTCS.125.5

set of ODEs. The construction method is ad hoc and therefore difficult to generalize. Similar hybrid cell cycle models can be found elsewhere [1].

Recently, we have proposed a hybridization method for systematically deriving hybrid models from smooth ODE models [16, 17]. In this method, non-linear reaction rate functions of biochemical reactions are approximated by simpler, piecewise linear functions. The hybrid model contain new parameters that can be estimated by a combination of linear programming and least squares optimization. In this paper we discuss an application of the method to a medium size cell cycle model. Our method has some similarities to the method proposed in [10] to learn hybrid models from action potentials, but there are also differences, such as the definition of the modes and of the mode switching, and the optimization scheme.

2 Piecewise smooth hybrid models

We consider piecewise smooth hybrid dynamical systems (HDS) for which the continuous variables, u, satisfy the equations

$$\begin{aligned} \frac{du_i}{dt} &= \sum_{k=1}^{N_i} s_k P_{ik}(u) + P_i^0(u) - \sum_{l=1}^{M_i} \tilde{s}_l Q_{il}(u) - Q_i^0(u), \\ s_j &= H(\sum_{k\in C_j} w_{jk}u_k - h_j), \quad \tilde{s}_l = H(\sum_{k\in \tilde{C}_l} \tilde{w}_{lk}u_k - \tilde{h}_l), \end{aligned} \tag{1}$$

where H is the unit step function $H(y) = 1, y \geq 0$, and $H(y) = 0, y < 0$, $P_{ik}, P_i^0, Q_{il}, Q_i^0$ are positive, smooth functions of u representing production, basal production, consumption, and basal consumption, respectively. Here $w, \tilde{w}$ are matrices describing the interactions between the u variables, $i = 1,2,...,n$, $j = 1,2,...,N$, $l = 1,...,M$ and $h, \tilde{h}$ are thresholds, and C_j, $\tilde{C}_l$ are indices subsets corresponding to continuous variables controlling the discrete variables.

One will usually look for solutions of the piecewise-smooth dynamics (1) such that trajectories of $\boldsymbol{u}$ are continuous. However, we can easily extend the above definitions in order to cope with jumps of the continuous variables. Similarly to impact systems occurring in mechanics [7], the jumps of the continuous variables can be commanded by the following rule: $\boldsymbol{u}$ instantly changes to $\boldsymbol{p}_j^{\pm}(\boldsymbol{u})$ whenever a discrete variable $\hat{s}_j = H(\sum_{k\in\hat{C}_j} \hat{w}_{jk}u_k - \hat{h}_j)$ changes. The $\pm$ superscripts correspond to changes of $\hat{s}_j$ from 0 to 1 and from 1 to 0, respectively. We can consider reversible jumps in which case the functions $\boldsymbol{p}_j^{\pm}(\boldsymbol{u})$ satisfy $\boldsymbol{p}^+ \circ \boldsymbol{p}^- = Id$. The typical example in molecular biology is the cell cycle. In this case, the command to divide at the end of mitosis is irreversible and corresponds to $\boldsymbol{p}_j^+(\boldsymbol{u}) = \boldsymbol{u}/2$. No return is possible, $\boldsymbol{p}_j^-(\boldsymbol{u}) = \boldsymbol{u}$.

The class of models (1) is too general. We will restrict ourselves to a subclass of piecewise smooth systems where smooth production and degradation terms are assumed multivariate monomials in u, plus some basal terms that we try to make as simple as possible. A system with constant basal production and linear basal consumption is the following:

$$\begin{aligned} P_{ik}(\boldsymbol{u}) &= a_{ik}u_1^{\alpha_1^{ik}} \dots u_n^{\alpha_n^{ik}}, \\ P_i^0(\boldsymbol{u}) &= a_i^0, \\ Q_{il}(\boldsymbol{u}) &= \tilde{a}_{il}u_1^{\tilde{\alpha}_1^{il}} \dots u_n^{\tilde{\alpha}_n^{il}}, \\ Q_i^0(\boldsymbol{u}) &= \tilde{a}_i^0 u_i. \end{aligned} \tag{2}$$

Multivariate monomial rates represent good approximations for nonlinear networks of biochemical reactions with multiple separated timescales [20, 9]. Such examples are abundant in chemical kinetics. For instance, Michaelis Menten, Hill, or Goldbeter-Koshland reactions switch from a saturated regime where rates are constant to a small concentration regime where rates follow power laws. The definition of the rates reminds that of S-systems, introduced by Savageau [22]. Finally, as discussed in [14] monomial approximations occur naturally in "tropically-truncated" polynomial systems, ie in systems where polynomial or rational rate functions are replaced by a few dominating monomials. As compared with our previous work [14, 18], in this paper we use the tropicalization only heuristically to obtain simpler reaction kinetic laws, whose parameters are then fitted. The switching of the monomial terms is not given by the max-plus rule as in [14, 18], but is commanded by thresholding functions depending on parameters to be fitted. This allows for more flexibility and corrects the errors introduced by the tropicalization.

3 Hybridization of the generic mammalian cell cycle model

This model has been proposed by the group of Tyson [5] and is designed to be a generic model of the cell cycle for eukaryotes. The cell cycle being an old, but important system that evolved, there have to be homologies, i.e. common mechanisms shared by the cell cycle regulation of all eukaryotes. The goal of this model is to bring to light these mechanisms, while producing models that reproduce experimental results. Four different eukaryotic organisms were modelled : budding yeast, fission yeast, Xenopus embryos, and mammalian cells. For each of theses organisms, a set of parameters is provided. By changing parameter sets, one can activate or deactivate some modules, fine tune some mechanisms, in order to reproduce the behaviour of the cell cycle in the chosen organism.

We analyse here only the model describing mammalian cells. This model uses twelve variables (eleven of them being concentrations of proteins, and one being the mass of the cell) and forty reactions. We briefly discuss the steps of the algorithm applied to this model.

Choice of the hybrid scheme. Five of these reactions are typically switch-like, following Goldbeter-Koshland kinetics, defined as follows:

$$GK(v_1, v_2, J_1, J_2) = \frac{2v_1 J_2}{B + \sqrt{B^2 - 4(v_2 - v_1)v_1 J_2}}, \tag{3}$$

with $B = v_2 - v_1 + J_1 v_2 + J_2 v_1$.

These kinetics describe a steady-state solution for a 2-state biological system, meaning that this reaction will have two basics modes : active or inactive. These reactions are replaced by switched reactions whose rates are simplified monomial rates multiplied by a boolean variable.

For instance the reaction that produces Cyclin-B, induced by the cell mass, has the following kinetic rate:

$$\mathrm{R} = \mathrm{ksb}_{pp}\,[\mathrm{Mass}]\,\mathrm{GK}(\mathrm{kafb}\,[\mathrm{CycB}], \mathrm{kifb}, \mathrm{Jafb}, \mathrm{Jifb}) \tag{4}$$

In this case we replace the Goldbeter-Koshland (GK) function by a step function and obtain the following simpler rate:

$$\mathrm{R}' = \mathrm{k}'\,[\mathrm{Mass}]\,\mathrm{s}, \tag{5}$$

where s is a boolean variable.

We apply the same method for all the five GK reactions of the model. The original and hybridized reaction rates can be find in the Table 2.1.

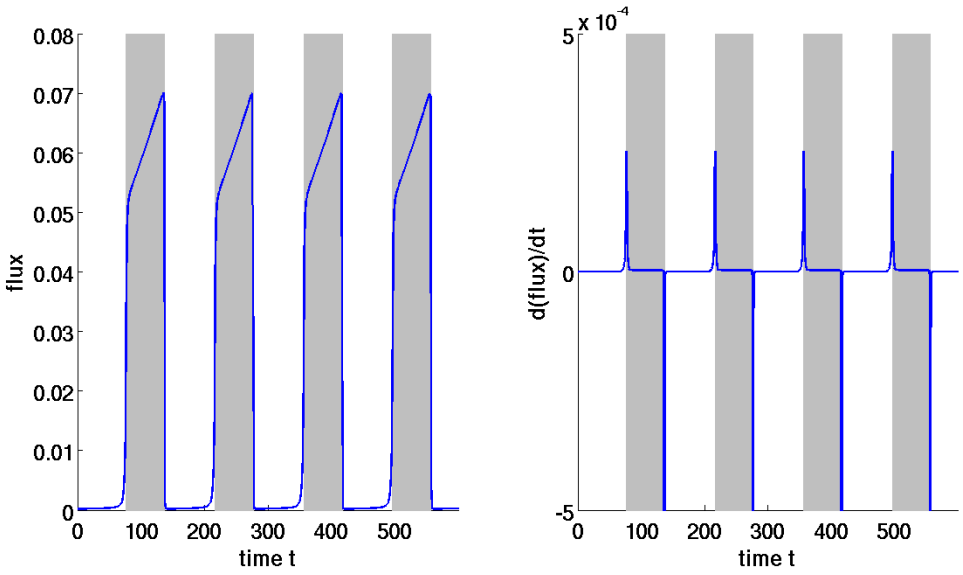


Figure 1: Flux and derivative of the flux for the Goldbeter-Koshland reaction R4. The shaded areas correspond to value where the inequation $v_1 > v_2$ is true.

Another set of reactions reactions we want to modify in this model are Michaelis-Menten (MM) reactions. We want to reproduce the two functioning modes of Michaelis-Menten kinetics, namely the linear and the saturated behaviour. The linear behaviour is observed when the substrate is in low supply. In this case, the flux of the MM reaction will be linear with respect to the substrate supply. The saturated behaviour is observed when the substrate supply is in excess, and produce a constant flux. Our goal is to obtain a hybrid reaction which switches between these two modes, controlled by boolean variables.

There are ten such reactions within this model. A classic MM reaction rate would be the following :

$$MM(X) = \frac{k.X}{X + k_m}, \tag{6}$$

and we propose to replace it by the following reaction :

$$MM(X) = s.k' + \tilde{s}.k''.X, \tag{7}$$

where s is a boolean variable, and $\tilde{s}$ is the complementary of s.

We apply this transformation to all the Michaelis-Menten reactions. The original and hybridized reaction rates can be found in the Table 2.1.

Detection of the transitions. Static event locations follow from the positions of sharp local maxima and minima of the derivatives of the reactions rates with respect to time (these correspond to sharp local maxima and minima of the second derivatives of the species concentrations, with respect to time). We have checked numerically that in the case of GK functions, these positions are close to the solutions of the equation $v_1 = v_2$. This property follows from the sigmoidal shape of the GK regulation functions. It is indeed well known that GK sigmoidal functions have an inflexion point defined by the condition $v_1 = v_2$, when the activation and inhibition input rates are equal. If the case of MM functions, these positions are close to the solutions of the equation $X = k_m$.

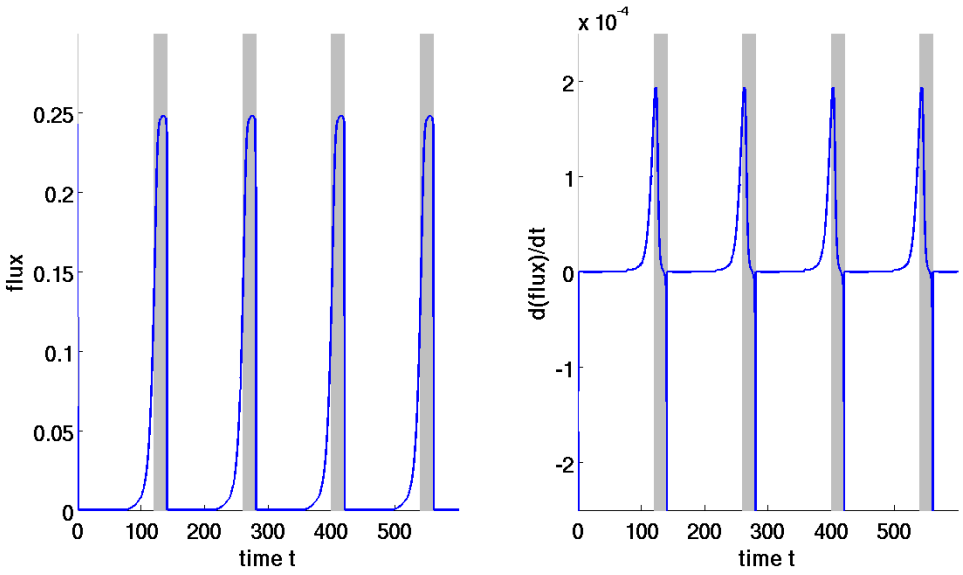


Figure 2: Flux and derivative of the flux for the Michaelis-Menten reaction R10. The shaded areas correspond to value where the inequation $X > k_m$ is true.

These findings are illustrated in Figs. 1,2. We can deduce the value of the boolean variables by checking the inequation $v_1 > v_2$ in the GK case, and $X > k_m$ in the MM case. We can observe that the change of the result of these inequations corresponds to the maxima and minima of the derivative (Fig. 1 and Fig. 2).

The structure of the model can be used to reduce the number of boolean control variables. In the case of reactions R11,12 or R13,14,15, we can see looking at reaction rates in the Table 2.1 that the inequations controlling their behavior should be the same. Thus, we can use the same boolean variable to control these reactions. Furthermore, we found out while looking for these transitions that for some MM reactions these transitions do not occur along the limit cycle trajectories. In the case of reactions R7 and R9, the behaviour is always saturated. We chose not to represent these reactions as hybrid (switched) reactions, and represented only their saturated behaviour.

We can use these inequalities and hybrid model description to fit parameters of the hybrid model in one of three ways :

i) Statically, meaning that the discrete variables times series $s(t)$ will be calculated at the detection step of the algorithm and will not change during the fit. In this case one fits only the parameters describing the modes. This has the benefit of simplicity, but comes with problems. The simplification in the representation of the reactions will introduce a difference between the original and the hybrid model, and such a difference should impact on the position of transitions.

ii) Statically, but allow for modifications of the discrete variables time series $s(t)$. We could try to include the positions of these transitions in the fitting parameters, but it would increase the complexity of the cost function. It would notably be a problem to modify all transitions occurring in a single reaction accordingly, which is important for the computation of mode control parameters.

iii) Dynamically. We could use the inequations defining the positions of the transitions dynamically, by evaluating them during the optimization. The transitions positions will be determined according

to the original model conditions applied to the hybrid model trajectories. This solves the problem of adapting transitions positions of one reaction with respect to the others. The problem is that the transition conditions from the original model are imported to the hybrid model with new parameters (thresholds) that have also to be fitted. As thresholds parameters are generally more sensitive, this choice increases the optimization difficulty.

Fitting the hybrid model parameters.

Once defining the model structure and the parameters to be fitted we can define a cost function representing the distance between the trajectories of the hybrid and smooth model. We use a parallel version of Lam's simulated annealing algorithm [11, 4] to minimize this cost function with respect to the parameters of the hybrid model. We limit the parameters search space to those involved in the hybridized reactions (a more extensive search is nevertheless possible). For the cost function, we have decided to test both species trajectories and reaction fluxes. When we limit ourselves to species trajectories, since some reactions have transitions that are close in time, there is a risk that some hybridized reaction will compensate for others. We wanted each hybridized reaction to be as much as possible a replica of the original reaction.

When using the definition with static discrete variables $s(t)$ and fitting only the mode parameters (cases i) above), we were not able to obtain even an imperfect fit of the model (in this case the trajectories of the hybrid model are very different from the ones of the original model and even become instable). We chose to include transition positions to the parameters of the fitting (case ii)), and were able to obtain a reasonable fit. However, the imperfections in the localisation of these new transition positions made difficult to find good control parameters (see next step) for all the hybridized reactions. The trajectories of the hybrid model fitted using this method are shown in Fig. 3. One can notice important differences between the trajectories of the hybrid and original model, although these differences remain bounded and the stability of the limit cycle oscillations is preserved.

When using the definition with the original model conditions for transitions (case iii)), we were able to obtain a working hybrid model, but the fit can still be improved by modifying slightly the mode control parameters. We can observe on Fig. 4 that while the dynamics of the model is preserved, there are differences in the transition positions.

Thus, when we included the parameters of transitions conditions, we obtained a model which fits better the original one. As a control we can see the results of the fitting on both the trajectories of the four main variables (Fig.5) and the fluxes of some hybridized reactions (Fig.6).
An interesting result of this optimization is that some hybridized reactions stopped having transitions, suggesting that the best fit would be obtained without these transitions. The reaction R6 (Fig. 7) is one of these reactions. This could be the result of the sensitivity of transition control parameters and a selection of a more robust solution.

Computing the mode control parameters. If we chose the static method of representing transitions during the fit, we now have to determine a regulation matrix, which will allow a dynamic definition of the events location.

Let $s_m = H(\sum_{j\in C_m} w_{mj}u_j - h_j)$ be the discrete variables and s_k^m the constant values of s_m piecewisely, on time intervals I_k identified at the detection step. Consider now the optimal trajectories $u_i^*(t_l)$, calculated piecewisely with fitted mode parameters.

Then, one should have

$$(\sum_{j\in C_m} w_{mj}u_j^*(t_l) - h_j)s_k^m > 0, \text{for all} t_l \in I_k. \tag{8}$$

This is a problem of linear programming, and is solved using the simplex algorithm [6].

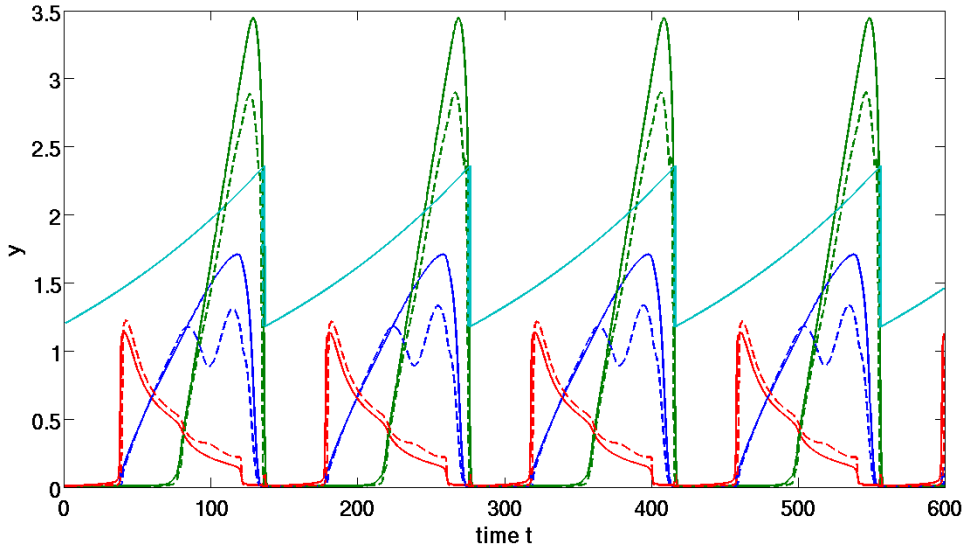


Figure 3: Comparison of the trajectories of the four main variables. (blue: Cyclin-A, green: Cyclin-B, red: Cyclin-C, aqua: cell size) (Plain lines) Original model (Dashed lines) Hybrid model without mode control parameters fitting (case ii)).

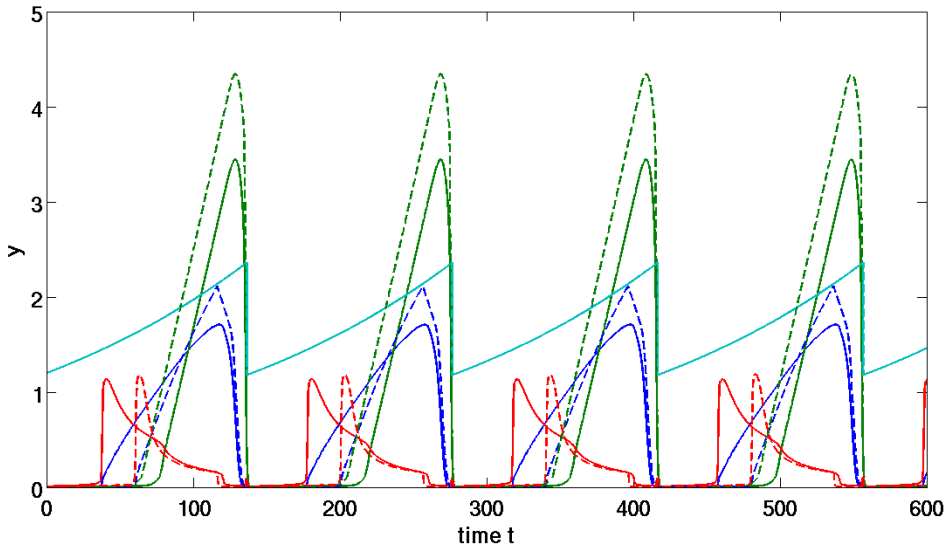


Figure 4: Comparison of the trajectories of the four main variables. (blue: Cyclin-A, green: Cyclin-B, red: Cyclin-C, aqua: cell size) (Plain lines) Original model (Dashed lines) Hybrid model without mode control parameters fitting (case iii)).

At this step, it is interesting to note that we have some choice on which variable can control a given reaction, i.e. on the subsets C_m. This potentially leads to multiple solutions of the inequations. The

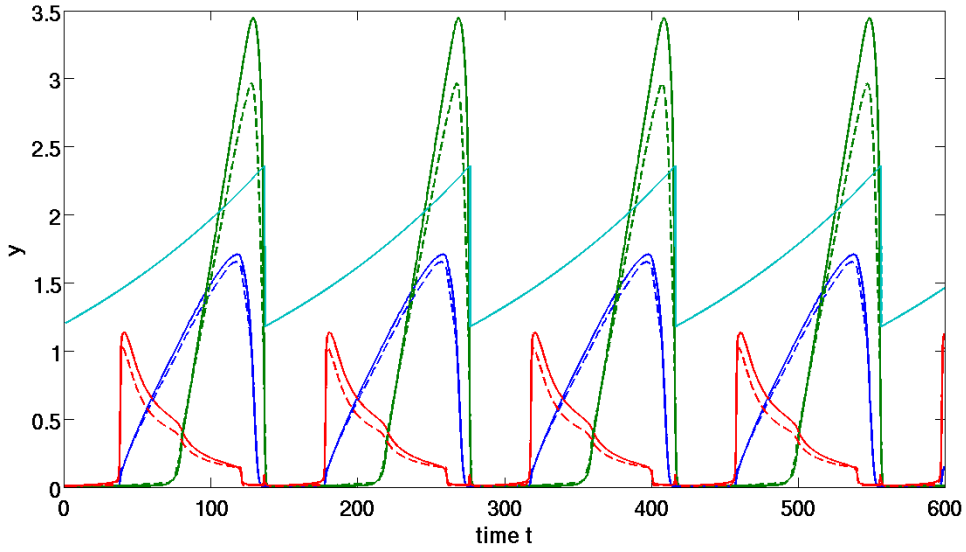


Figure 5: Comparison of the trajectories of the four main variables. (blue: Cyclin-A, green: Cyclin-B, red: Cyclin-C, aqua: cell size) (Plain lines) Original model (Dashed lines) Hybrid model with mode control parameters fitting (case iii))W.

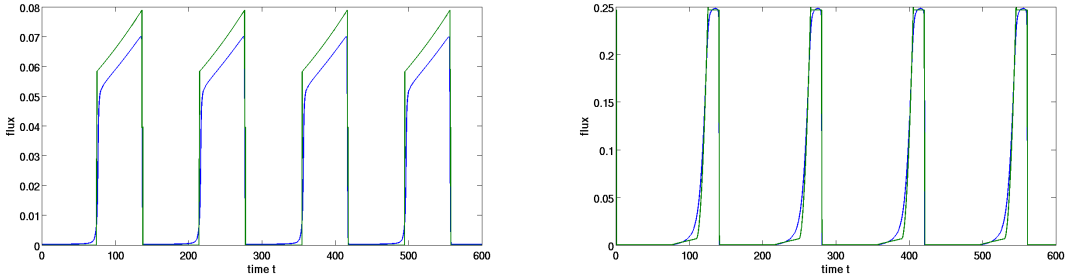


Figure 6: Comparison of original and hybridized reaction fluxes. Top : GK Reaction R4. Bottom : MM Reaction R10. Blue : flux of original reaction, Green : flux of hybridised reaction

best choice would be here to use the biological knowledge to choose the species actually involved in the reaction.

The problem with this step is that its success depends on the conservation of the transition positions between the simulations with static and dynamic mode control. Or, this property is valid only to some extent and the dynamic transitions can shift with respect to their static positions. As a consequence, solving all the inequations (8) may sometimes be impossible.

To cope with this issue, we introduced a variable ε so that the inequalities (8) are modified to :

$$(\sum_{j\in C_m} w_{mj}u_j^*(t_l) - h_j)s_k^m + \varepsilon > 0, \text{for all} \quad t_l \in I_k, \tag{9}$$

This modification enables us to solve all the inequations, and gives us a good metric to asses the

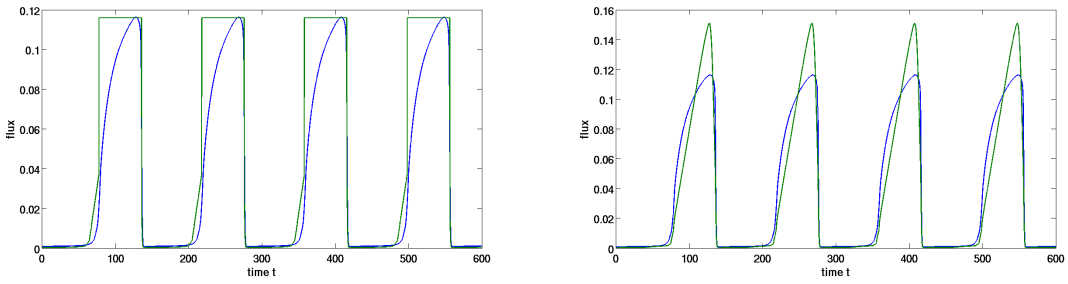


Figure 7: Comparison of reaction R6 results without and with the fitting of transition control parameters. Top : Fit without control parameters. Bottom : Fit with control parameters. Blue : flux of original reaction, Green : flux of hybridized reaction

quality of the resolution. Furthermore, the parameter ε can be minimized within the simplex algorithm. The ideal case is when ε is negative or zero. When simulating the hybrid model, we found out that with a positive epsilon, the model is most of the time unstable.

Periodicity is not the only difficulty for this step. In our formalism, the threshold to modify the boolean variables controlling a given reaction is the same for an activation or an inactivation. This could also be a problem, as we can not always enforce such a condition during the fitting. There are different solutions to this problem. The first one would be to have different thresholds for reaction activation and inactivation, but this choice misses the simplicity of the previous method of control. More precisely, even activation and inactivation thresholds correspond geometrically to control of the modes by manifold crossing (activation when crossing takes place in one direction, inactivation for crossing in the opposite direction), whereas different thresholds do not allow for such a simple picture.
The other solution would be not to limit ourselves to the biologically relevant variables to control these transitions. As we increase the number of variables, the probability to find a combination which satisfies the inequations increases. The problem with this choice is the large number of possible combinations. We used a genetic algorithm which selects the variables which had the lowest ε value and were able to find combinations which satisfy the inequations for some reactions. But for others reaction, especially Michaelis-Menten reactions, even with all variables, we were not able to obtain low enough ε. We were able to use this method to build a hybrid model which only hybridized the Goldbeter-Koshland reactions. The result can be seen in Fig. 8 and the corresponding model is given in Table 2.3.

4 Conclusion

We have presented a hybridization scheme, allowing to transform a biochemical network model, containing reactions with complex non-linear rate functions, into a hybrid model with piece-wise linear rate functions. This scheme can be applied to any model containing Goldbeter-Koshland or Michaelis-Menten kinetic laws. More generally, extensions of this method can be applied to biochemical network models whose kinetic laws are rational functions of the species concentrations. These include the Goldbeter-Koshland case, as this mechanism is obtained by model reduction from two coupled Michaelis-Menten reactions. The resulting hybrid model in the general case is piecewise smooth, but not necessarily linearly smooth. This generalization is based on tropicalization [15, 14] and consists in approximating rational rate functions by tropical polynomials, that are represented piecewisely by

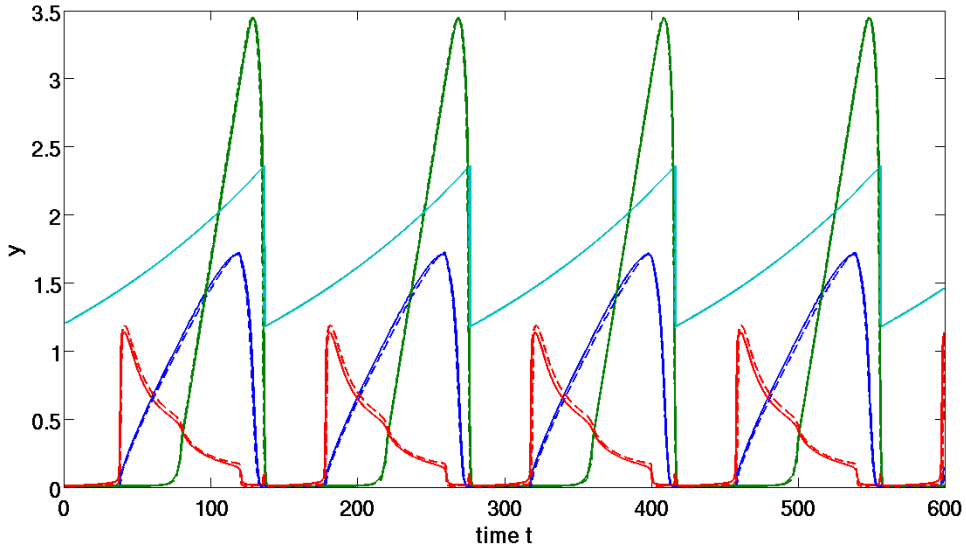


Figure 8: Comparison of the trajectories of the four main variables. (blue: Cyclin-A, green: Cyclin-B, red: Cyclin-C, aqua: cell size) (Plain lines) Original model (Dashed lines) Hybrid model with mode control parameters fitting.

multivariate monomials.

The identification algorithm proposed in the paper combines the static or dynamic location of the events, the identification of the mode parameters by simulated annealing, and the identification of the mode control parameters by linear programming. The hardest step of this algorithm is the simulated annealing. We have discussed three optimization strategies to reduce the number of the parameters to be determined by simulated annealing, while keeping the flexibility of the optimization scheme. In this paper, the hybrid cell cycle model has been obtained from artificial trajectories generated with a smooth model. That allowed us to include both concentration and rates trajectories in the cost function, which is a strong constraint. In the future, this constraint could be released and cost functions based on concentration trajectories only, could be used to learn hybrid cell cycle models directly from experimental data.

5 Acknowledgements

VN thanks University of Rennes 1 for supporting his research and CNRS PEPS MODREDBIO for sponsoring his participation to HSB2013.

Tables

Table 2.1 - Definition of reactions in the original and hybridized mammalian cell cycle model. The inequalities controlling the mode switching result directly from the definition of the reaction rates in the original model.

reaction smooth	variables	reaction hybrid	control
$R_1 = ksa_{pp}.[Mass].$ $.GK(v_1, v_2, Jatf, Jitf)$	$v_1 = katf_p + katfa_{pp}.[CycA] +$ $+katfd_{pp}.CycD0.[Mass]$ $v_2 = kitf_p + kitfa_{pp}.$ $.[CycA] + kitfb_{pp}.[CycB]$	$R_{1h} = ksa_{pp}.[Mass].s_1$	$s_1 = v_1 > v_2$
$R_2 = k25_{pp}.[pB]$ $.GK(v_1, v_2, Ja25, Ji25)$	$v_1 = ka25_p + ka25_{pp}.[CycE]$ $v_2 = ki25_p + ki25_{pp}.[Cdc20A]$	$R_{2h} = k25_{pp}.[pB].s_2$	$s_2 = v_1 > v_2$
$R_3 = kwee_{pp}.[CycB]$ $.GK(v_1, v_2, Jawee, Jiwee)$	$v_1 = kawee_p + kawee_{pp}.[Cdc20A]$ $v_2 = kiwee_p + kiwee_{pp}.[CycB]$	$R_{3h} = kwee_{pp}.[CycB].s_3$	$s_3 = v_1 > v_2$
$R_4 = ksb_{pp}.[Mass]$ $.GK(v_1, v_2, Jafb, Jifb)$	$v_1 = kafb.[CycB]$ $v_2 = kifb$	$R_{4h} = ksb_{pp}.[Mass].s_4$	$s_4 = v_1 > v_2$
$R_5 = kse_{pp}.[Mass]$ $.GK(v_1, v_2, Jatf, Jitf)$	$v_1 = katf_p + katfa_{pp}.[CycA] +$ $+katfd_{pp}.CycD0.[Mass]$ $v_2 = kitf_p + kitfa_{pp}.$ $.[CycA] + kitfb_{pp}.[CycB]$	$R_{5h} = kse_{pp}.[Mass].s_1$	$s_1 = v_1 > v_2$
$R_6 = ks20_{pp}$ $.X/(K_m + X)$	$X = [CycB]$ $K_m = J20$	$R_{6h} = ks20_{pp}.s_5$ $+ks20_{pp2}.\tilde{s}5.X$	$s_5 = X > K_m$
$R_7 = kaie.[CycB]$ $.X/(K_m + X)$	$X = (APCT - [APCP])$ $K_m = Jaie$	$R_{7h} = ks20_{pp}.[CycB]$	
$R_8 = kiie$ $.X/(K_m + X)$	$X = [APCP]$ $K_m = Jiie$	$R_{8h} = kiie.s_6$ $+kiie2 * \tilde{s}_6.X$	$s_6 = X > K_m$
$R_9 = ka20.[APCP]$ $.X/(K_m + X)$	$X = [Cdc20i]$ $K_m = Ja20$	$R_{9h} = ka20.[APCP]$	
$R_{10} = ki20$ $.X/(K_m + X)$	$X = [Cdc20A]$ $K_m = Ji20$	$R_{10h} = ki20.s_7$ $+ki202.\tilde{s}_7.X$	$s_7 = X > K_m$
$R_{11} = kah1_p$ $.X/(K_m + X)$	$X = (Cdh1T - [Cdh1])$ $K_m = Jah1$	$R_{11h} = kah1_p.s_8$ $+kah1_{p2}.\tilde{s}_8.X$	$s_8 = X > K_m$
$R_{12} = kah1_{pp}.[Cdc20A]$ $.X/(K_m + X)$	$X = (Cdh1T - [Cdh1])$ $K_m = Jah1$	$R_{12h} = kah1_{pp}.[Cdc20A].s_8$ $+kah1_{pp2}.[Cdc20A].\tilde{s}_8.X$	$s_8 = X > K_m$
$R_{13} = kih1a_{pp}.[CycA]$ $.X/(K_m + X)$	$X = [Cdh1]$ $K_m = Jih1$	$R_{13h} = kih1a_{pp}.[CycA].s_9$ $+kih1a_{pp2}.[CycA].\tilde{s}_9.X$	$s_9 = X > K_m$
$R_{14} = kih1b_{pp}.[CycB]$ $.X/(K_m + X)$	$X = [Cdh1]$ $K_m = Jih1$	$R_{14h} = kih1b_{pp}.[CycB].s_9$ $+kih1b_{pp}.[CycB].\tilde{s}_9.X$	$s_9 = X > K_m$
$R_{15} = kih1e_{pp}.[CycE]$ $.X/(K_m + X)$	$X = [Cdh1]$ $K_m = Jih1$	$R_{15h} = kih1e_{pp}.[CycE].s_9$ $+kih1e_{pp}.[CycE].\tilde{s}_9.X$	$s_9 = X > K_m$

Table 2.2.1 - Parameters of the original mammalian cell cycle model described in the table 2.1.

constant	value
kse_{pp}	0.18
$katfa_{pp}$	0.2
$katfd_{pp}$	3.0
$katfe_{pp}$	0.5
$kitf_p$	0.25
$kitfa_{pp}$	0.1

$kitfb_{pp}$	0.1
ksb_{pp}	0.03
$kwee_{pp}$	0.2
$k25_{pp}$	5
$kafb$	1.0
$kifb$	0.1
ksa_{pp}	0.025
$kaie$	0.07
$kiie$	0.18
$Jaie$	0.01
$Jiie$	0.01
$ks20_{pp}$	0.15
$J20$	1
$ka20$	0.5
$ki20$	0.25
$Ji20$	0.0050
$kah1_p$	0.18
$kah1_{pp}$	3.5
$kih1a_{pp}$	0.2
$kih1b_{pp}$	1.0
$kih1e_{pp}$	0.1
$Jah1$	0.01
$Jih1$	0.01
$kawee_p$	0.3
$kiwee_{pp}$	1.0
$ka25_{pp}$	1
$ki25_p$	0.3

Table 2.2.2 - Parameters of the hybridized mammalian cell cycle model described in the table 2.1.

constant	value
ksa_{pp}	0.024635
$katf_p$	0
$katfa_{pp}$	0.00090318
$katfd_{pp}$	2.6897
$katfe_{pp}$	2.1407
$kitf_p$	0.22282
$kitfa_{pp}$	0
$kitfb_{pp}$	0.14253
$k25_{pp}$	3.559
$ka25_p$	0
$ka25_{pp}$	21.93
$ki25_p$	5.425
$ki25_{pp}$	0
$kwee_{pp}$	0.096009
$kawee_p$	3.5714
$kawee_{pp}$	0

$kiwee_p$	0
$kiwee_{pp}$	9.003
ksb_{pp}	0.033299
$kafb$	0.15998
$kifb$	0.0056319
kse_{pp}	0.14842
$ks20_{pp}$	0
$ks20_{pp2}$	0.048074
$J20$	3
$kaie$	0.076693
$kiie$	0.1685
$kiie2$	17.568
$Jiie$	0.0096156
$ka20$	0.4815
$ki20$	0.24271
$ki202$	5.3118
$Ji20$	0.045084
$kah1_p$	0
$kah1_{p2}$	0.15387
$kah1_{pp}$	0
$kah1_{pp2}$	3.9768
$Jah1$	1
$kih1a_{pp}$	0.099851
$kih1a_{pp2}$	2.5689
$kih1b_{pp}$	0
$kih1b_{pp2}$	14.966
$kih1e_{pp}$	0.10502
$kih1e_{pp2}$	1.7755
$Jih1$	0.12035

Table 2.3 - Definition of reactions in the original and hybridized mammalian cell cycle model. The inequalities controlling the mode switching result from the computation of mode control parameters post-fitting.

reaction smooth	variables	reaction hybrid
$R_1 = ksa_{pp}.[Mass].$ $.GK(v_1, v_2, Jatf, Jitf)$	$v_1 = katf_p + katfa_{pp}.[CycA] +$ $+katfd_{pp}.CycD0.[Mass]$ $v_2 = kitf_p + kitfa_{pp}.$ $.[CycA] + kitfb_{pp}.[CycB]$	$R_{1h} = ksa_{pp}.[Mass].s_1$
$R_2 = k25_{pp}.[pB]$ $.GK(v_1, v_2, Ja25, Ji25)$	$v_1 = ka25_p + ka25_{pp}.[CycE]$ $v_2 = ki25_p + ki25_{pp}.[Cdc20A]$	$R_{2h} = k25_{pp}.[pB].s_2$
$R_3 = kwee_{pp}.[CycB]$ $.GK(v_1, v_2, Jawee, Jiwee)$	$v_1 = kawee_p + kawee_{pp}.[Cdc20A]$ $v_2 = kiwee_p + kiwee_{pp}.[CycB]$	$R_{3h} = kwee_{pp}.[CycB].s_3$
$R_4 = ksb_{pp}.[Mass]$ $.GK(v_1, v_2, Jafb, Jifb)$	$v_1 = kafb.[CycB]$ $v_2 = kifb$	$R_{4h} = ksb_{pp}.[Mass].s_4$
$R_5 = kse_{pp}.[Mass]$ $.GK(v_1, v_2, Jatf, Jitf)$	$v_1 = katf_p + katfa_{pp}.[CycA] +$ $+katfd_{pp}.CycD0.[Mass]$	$R_{5h} = kse_{pp}.[Mass].s_1$

	$v_2 = kitf_p + kitfa_{pp}.$ $.[CycA] + kitfb_{pp}.[CycB]$

control	value
s_1	$w_{1,1}.[CycA]+w_{1,2}.[CycB]+w_{1,4}.[APCP]+w_{1,6}.[Cdc20i]+w_{1,7}.[Cdh1]+w_{1,8}.[CKI]+$ $w_{1,9}.[Mass]-1>0$
s_2	$w_{2,3}.[CycE]+w_{2,9}.[Mass]+w_{2,12}.[TriE]-1>0$
s_3	$w_{3,3}.[CycE]+w_{3,9}.[Mass]+w_{3,12}.[TriE]-1>0$
s_4	$w_{2,2}.[CycB]+w_{2,9}.[Mass]+w_{2,10}.[pB]-1>0$

Table 2.4 - Parameters of the hybridized mammalian cell cycle model described in the table 2.3.

constant	value
ksa_{pp}	0.024064
kse_{pp}	0.18569
$kwee_pp$	0.17326
$k25_{pp}$	3.5168
ksb_{pp}	0.030148
$w_{1,1}$	1.e+9
$w_{1,2}$	0.4352e+9
$w_{1,4}$	-1.5677e+9
$w_{1,6}$	-4.0592e+9
$w_{1,7}$	1.e+9
$w_{1,8}$	-0.7937e+9
$w_{1,9}$	0.1138e+9
$w_{2,3}$	-2.218e+9
$w_{2,9}$	1.e+9
$w_{2,12}$	-10.027e+9
$w_{3,3}$	0.2278e+9
$w_{3,9}$	-0.1015e+9
$w_{3,12}$	1.e+9
$w_{4,2}$	0.2294e+9
$w_{4,9}$	-0.0294e+9
$w_{4,10}$	1e+9

References

[1] R. Alfieri, E. Bartocci, E. Merelli & L. Milanesi (2011): *Modeling the cell cycle: From deterministic models to hybrid systems*. Biosystems 105, pp. 34–40, doi:10.1016/j.biosystems.2011.03.002.

[2] V. Baldazzi, P. T. Monteiro, M. Page, D. Ropers, J. Geiselmann & H. Jong (2011): *Qualitative Analysis of Genetic Regulatory Networks in Bacteria*. Understanding the Dynamics of Biological Systems, pp. 111–130, doi:10.1007/978-1-4419-7964-3-6.

[3] G. Batt, H. De Jong, M. Page & J. Geiselmann (2008): *Symbolic reachability analysis of genetic regulatory networks using discrete abstractions*. Automatica 44(4), pp. 982–989, doi:10.1016/j.automatica.2007.08.004.

[4] K.W. Chu, Y. Deng & J. Reinitz (1999): *Parallel Simulated Annealing by Mixing of States 1*. Journal of Computational Physics 148(2), pp. 646–662, doi:10.1006/jcph.1998.6134.

[5] A. Csikász-Nagy, D. Battogtokh, K. C. Chen, B. Novák & J. J. Tyson (2006): *Analysis of a generic model of eukaryotic cell-cycle regulation*. Biophysical Journal 90(12), pp. 4361–4379, doi:10.1529/biophysj.106.081240.

[6] G. B. Dantzig, A. Orden & P. Wolfe (1955): *The generalized simplex method for minimizing a linear form under linear inequality restraints*. Pacific Journal of Mathematics 5(2), pp. 183–195, doi:10.2140/pjm.1955.5.183.

[7] M. Di Bernardo (2008): *Piecewise-smooth dynamical systems: theory and applications*. Springer Verlag, doi:10.1007/978-1-84628-708-4.

[8] S. Drulhe, G. Ferrari-Trecate & H. de Jong (2008): *The switching threshold reconstruction problem for piecewise-affine models of genetic regulatory networks*. IEEE Transactions on Automatic Control 53, pp. 153–165, doi:10.1109/TAC.2007.911326.

[9] A. N. Gorban & O. Radulescu (2008): *Dynamic and static limitation in reaction networks, revisited* . In David West Guy B. Marin & Gregory S. Yablonsky, editors: Advances in Chemical Engineering - Mathematics in Chemical Kinetics and Engineering, Advances in Chemical Engineering 34, Elsevier, pp. 103–173, doi:10.1016/S0065-2377(08)00003-3.

[10] Radu Grosu, S Mitra, Pei Ye, Emilia Entcheva, IV Ramakrishnan & Scott A Smolka (2007): *Learning cycle-linear hybrid automata for excitable cells*. In: Hybrid Systems: Computation and Control, Springer, pp. 245–258, doi:10.1007/978-3-540-71493-4_21.

[11] Jimmy Lam & Jean-Marc Delosme (1988): *An efficient simulated annealing schedule: derivation*. Yale University, New Haven, Connecticut, Technical Report 8816, doi:10.1.1.74.4461.

[12] P. Lincoln & A. Tiwari (2004): *Symbolic systems biology: Hybrid modeling and analysis of biological networks*. Hybrid Systems: Computation and Control, pp. 147–165, doi:10.1007/978-3-540-24743-2-44.

[13] B. Mishra (2009): *Intelligently deciphering unintelligible designs: algorithmic algebraic model checking in systems biology*. Journal of The Royal Society Interface 6(36), pp. 575–597, doi:10.1098/rsif.2008.0546.

[14] V. Noel, D. Grigoriev, S. Vakulenko & O. Radulescu: *Tropicalization and tropical equilibrations of chemical reactions*. Contemporary Mathematics, in press; arXiv : 1303.3963.

[15] V. Noel, D. Grigoriev, S. Vakulenko & O. Radulescu (2012): *Tropical geometries and dynamics of biochemical networks. Application to hybrid cell cycle models*. Electronic Notes in Theoretical Computer Science 284, pp. 75–91, doi:10.1016/j.entcs.2012.05.016.

[16] V. Noel, S. Vakulenko & O. Radulescu (2010): *Piecewise smooth hybrid systems as models for networks in molecular biology*. In: Proceedings of JOBIM 2010, Jobim, Montpellier.

[17] V. Noel, S. Vakulenko & O. Radulescu (2011): *Algorithm for identification of piecewise smooth hybrid systems: application to eukaryotic cell cycle regulation*. Lecture Notes in Computer Science 6833, pp. 225–236, doi:10.1007/978-3-642-23038-7-20.

[18] Vincent Noel, Dima Grigoriev, Sergei Vakulenko & Ovidiu Radulescu (2012): *Hybrid models of the cell cycle molecular machinery*. Electronic Proceedings in Theoretical Computer Science 92, pp. 88–105, doi:10.4204/EPTCS.92.7.

[19] R. Porreca, S. Drulhe, H. Jong & G. Ferrari-Trecate (2008): *Structural identification of piecewise-linear models of genetic regulatory networks*. Journal of Computational Biology 15(10), pp. 1365–1380, doi:10.1089/cmb.2008.0109.

[20] O. Radulescu, A. N. Gorban, A. Zinovyev & A. Lilienbaum (2008): *Robust simplifications of multiscale biochemical networks*. BMC systems biology 2(1), p. 86, doi:10.1186/1752-0509-2-86.

[21] D. Ropers, V. Baldazzi & H. de Jong (2011): *Model reduction using piecewise-linear approximations preserves dynamic properties of the carbon starvation response in Escherichia coli*. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8(1), pp. 166–181, doi:10.1109/TCBB.2009.49.

[22] Alexander Orden Savageau, M. A. anGeorge Bernard Dantzig & E. O. Philip Wolfed Voit (1987): *Recasting nonlinear differential equations as S-systems: a canonical nonlinear form*. Mathematical biosciences 87(1), pp. 83–115, doi:10.1016/0025-5564(87)90035-6.

[23] R. Singhania, R. M. Sramkoski, J. W. Jacobberger & J. J. Tyson (2011): *A hybrid model of mammalian cell cycle regulation*. PLoS computational biology 7(2), p. e1001077, doi:10.1371/journal.pcbi.1001077.

[24] P. Ye, E. Entcheva, SA Smolka & R. Grosu (2008): *Modelling excitable cells using cycle-linear hybrid automata*. Systems Biology, IET 2(1), pp. 24–32, doi:10.1049/iet-syb:20070001.

Exploring the Dynamics of Mass Action Systems

Oded Maler*
CNRS-VERIMAG
University of Grenoble
Oded.Maler@imag.fr

Ádám M. Halász†
Department of Mathematics
West Virginia University
halasz@math.wvu.edu

Olivier Lebeltel
CNRS-VERIMAG
University of Grenoble
Olivier.Lebeltel@imag.fr

Ouri Maler
Grenoble
ouri.maler@gmail.com

1 Introduction

Mass action is a fundamental notion in many situations in Chemistry, Biochemistry, Population Dynamics and Social Systems [1]. In this class of phenomena, one has a large population of individuals partitioned into several types of "species", whose dynamics is specified by a set of reaction rules. Each reaction indicates the transformation that is likely to take place when individuals of specific types come into contact. For example, a rule of the form $A+B \rightarrow A+C$ says that each time an instance of A meets an instance of B, the latter is transformed into a C. Denoting by n_A and n_B the number of instances of A and B existing at a certain moment, the likelihood of an (A,B)-encounter is proportional to $n_A \cdot n_B$. Hence the rate of change of n_B will have a *negative* contribution proportional to $n_A \cdot n_B$ and that of n_C will have the same magnitude of *positive* contribution. Combining for each of the species the negative contributions due to reactions in which it is transformed into something else with the positive contributions due to reactions that yield new instances of it, one typically obtains a system of *polynomial*[1] differential/difference equations.

The goal of the research program sketched in this paper (initially inspired by [3]) is to build a class of *synthetic* mathematical models for such systems, admitting some nice and clean properties which will reflect essential and fundamental aspects of mass action behavior while at the same time abstract away from accidental real-life details due to Chemistry, Physics and even some Geometry. Introducing such details at this preliminary stage would obscure the essence and render the analysis more complex. These models will then be subject to various investigations by analytical, simulation-based and other methods to explore their dynamics and discover the principles that govern their behavior. Such investigations may eventually lead to novel ways to control mass action systems with potential applications, among others, in drug design and social engineering. These issues have been studied, of course, for many years in various contexts and diverse disciplines, [7, 2] to mention a few, but we hope, nevertheless, to provide a fresh look on the subject.

The rest of this paper is organized as follows. In Section 2 we present the basic model of the individual agent (particle) as a *probabilistic automaton* capable of being in one out of several states, and where transition labels refer to the state of the agent it encounters at a given moment. We then discuss several ways to embed these individual agents in a model depicting the evolution of a large ensemble

*Supported by the ANR projects Syne2Arti, Eqinocs and Cadmidia.
†Supported by NIHGrant K25 CA131558. Work done while the author was visiting CNRS-VERIMAG.
[1] Actually *bilinear* if one assumes the probability of triple encounters to be zero, as is often done in Chemistry.

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 84–91, doi:10.4204/EPTCS.125.6

of their instances. In Section 3 we describe three such aggregate models. We start with a rather standard model where state variables correspond to the relative concentrations of agent types. Such models depict the dynamics of the *average* over all behaviors and they are traditionally ODEs but we prefer to work in discrete time to simplify the notation. The second model is based on stochastic simulation under the well-stirred assumption with no modeling of *space*, which is introduced in the third model where particles wander in space in some kind of random motion and a reaction takes place when the distance between two particles becomes sufficiently small. The model thus obtained is essentially a kind of a *reaction-diffusion* model. In Section 4 we briefly describe the **Populus** tool kit that we developed for exploring the dynamics of such models and illustrate its functionality by demonstrating some effects of the initial spatial distributions of some particles that lead to deviation from the predictions of a well-stirred model.

2 Individual Models and Aggregation Styles

We consider mass action systems where new individuals are not born and existing ones neither die nor aggregate into compound entities: they only change their *state*.

2.1 Individuals

A particle can be in one of finitely-many states and its (probabilistic) dynamics depicts what happens to it (every time instant) either spontaneously or upon encountering another particle. The object specifying a particle is a probabilistic automaton:[2]

Definition 1 (Probabilistic Automaton) *A probabilistic automaton is a triple $\mathscr{A} = (Q, \Sigma, \delta)$ where Q is a finite set of states, Σ is a finite input alphabet and $\delta : Q \times \Sigma \times Q \to \mathbb{R}$ is a probabilistic transition function such that for every $q \in Q$ and $a \in \Sigma$,*

$$\sum_{q' \in Q} \delta(q, a, q') = 1.$$

In our model $Q = \{q_1, \ldots, q_n\}$ is the set of particle types and each instance of the automaton is always in one of those. The input alphabet is $Q \cup \{\perp\}$ intended to denote the type of *another* particle encountered by the automaton and with the special symbol $\perp$ indicating a non-encounter. Intuitively, $\delta(q_1, q_2, q_3)$ represents the probability that an agent of type q_1 converts to type q_3 when it encounters an agent of type q_2. Likewise $\delta(q_1, \perp, q_3)$ is the probability of becoming q_3 spontaneously without meeting anybody. Table 1 depicts a 3-species probabilistic automaton. We use the notation $q_1 \xrightarrow{q_2} q_3$ for an actual invocation of the rule, that is, drawing an element of Q according to probability $\delta(q_1, q_2, .)$ and obtaining q_3 as an outcome.

In general our models are *synchronous* with respect to time: time evolves in fixed-size steps and at every step each particle detects whether it encounters another (and of what type) and takes the appropriate transition. The interpretation of when an agent meets another depends, as we shall see, on additional assumptions on the global aggregate model. It is worth noting that we restrict ourselves here to reaction rules which are "causal" in the following sense: when an (A, B)-encounter takes place, the influence of

[2] A probabilistic automaton [8] is a Markov chain with an input alphabet where each input symbol induces a different transition matrix. It is called sometimes a Markov Decision Process (MDP) but we prefer to reserve this term for a strategy synthesis problem in a game where the alphabet denotes the controller's action against a stochastic adversary. In the present model there is no "decision" associated with the input as it is an external (to the particle) influence on the dynamics.

A on B and the influence of B on A are *independent*. Hence not all types of probabilistic rewrite rules of the form $A+B \to A_1+B_1\ (p_1) \mid A_2+B_2\ (p_2) \mid \cdots \mid A_k+B_k\ (p_k)$ can be realized, only those that are products of simple rules. This restriction is not crucial for our approach but it simplifies some calculations.

δ	$\perp$			q_1			q_2			q_3		
q_1	0.9	0.1	0.0	1.0	0.0	0.0	0.7	0.2	0.1	0.7	0.0	0.3
q_2	0.1	0.8	0.1	0.0	0.6	0.4	0.0	1.0	0.0	0.1	0.9	0.0
q_3	0.0	0.0	1.0	0.7	0.0	0.3	0.3	0.4	0.3	0.0	0.0	1.0

$$\begin{aligned} x_1' &= x_1 - 0.09x_1 + 0.09x_2 - 0.06x_1x_2 + 0.08x_1x_3 + 0.08x_2x_3 \\ x_2' &= x_2 + 0.09x_1 - 0.18x_2 - 0.04x_1x_2 + 0.06x_2x_3 \\ x_3' &= x_3 + 0.09x_2 + 0.1x_1x_2 - 0.08x_1x_3 - 0.14x_2x_3 \end{aligned}$$

$$\begin{aligned} x_1' &= x_1 - 0.01x_1 + 0.01x_2 - 0.54x_1x_2 + 0.72x_1x_3 + 0.72x_2x_3 \\ x_2' &= x_2 + 0.01x_1 - 0.02x_2 - 0.36x_1x_2 + 0.54x_2x_3 \\ x_3' &= x_3 + 0.01x_2 + 0.9x_1x_2 - 0.72x_1x_3 - 1.26x_2x_3 \end{aligned}$$

Table 1: A 3-species probabilistic automaton, and the average dynamical system derived for the sparse situation $\alpha = 0.1$ and for the dense situation $\alpha = 0.9$. Starting from initial state $x = (0.4, 0.3, 0.3)$ the first system converges to the state $(0.366, 0.195, 0.437)$ while the second converges to $(0.939, 0.027, 0.033)$.

2.2 Aggregation Styles

Consider now a set S consisting of m individuals put together, each being modeled as an automaton. A global configuration of such a system should specify, at least, the state of each particle, resulting in the enormous state space Q^S consisting of n^m states (micro-states in Physpeak). A very useful and commonly-used abstraction is the *counting abstraction* obtained by considering two micro-state equivalent if they agree on the number of particles of each type, regardless of their particular identity. The equivalence classes of this relation form an abstract state-space of macro-states (also known as particle count representation) $P \subseteq S^Q$ consisting of n-dimensional vectors:

$$P = \{(X_1, \ldots, X_n) : \forall i\ 0 \leq X_i \leq m \wedge \sum_{i=1}^{n} X_i = m\}.$$

The formulation of a model that tracks the evolution of an ensemble of particles can be done in different styles. For our purposes we classify models according to two features: 1) Individual vs. average dynamics and 2) Spatially-extended vs. well-stirred dynamics. These two features are related but not identical.

For the first point, let us recall the trivial but important fact that we have a non-deterministic system where being in a given micro-state, each particle tosses one or more coins, properly biased according to the states of the other particles, so as to determine its next state. To illustrate, consider a rule which transforms a particle type A into B with probability p. Starting with m instances of A, there will be m coin tosses each with probability p leading to some number close to $m \cdot p$ indicating how many A's convert into

$B's$. Each individual run will yield a different number (and a different sequence of subsequent numbers) but on the average (over all runs) the number of A's will be reduced in the first step from m to $m \cdot (1-p)$.

Individualistic models, that is, stochastic simulation algorithms (SSA), generate such runs, one at the time. On the other hand, "deterministic" ODE models compute et every step the average number of particles for each type where this average is taken (in parallel) over all runs . For well-behaving systems, the relationship between this averaged trajectory and individual runs is of great similarity: the evolution in actual runs will appear as fluctuating around the evolution of the average. On the other hand, when we deal with more complex systems where, for example, trajectories can switch into two or more distinct and well-separated equilibria, the behavior of the average is not so informative. There is a whole research thread, starting with [4], that feeds on this important distinction (see [9, 6] for further discussions).

The other issue is whether and how one models the distribution of particles in *space*. Ignoring the spatial coordinates of particles, the probability of a particular transition being taken depends only on the total *numbers* of particles of each type, which is equivalent to the *well-stirred* assumption: all instances of each particle type are distributed uniformly in space and hence all particles will see the same proportion of other particles in their neighborhood. On the other hand, in spatially extended models each particle is endowed with a location which changes quasi-randomly and what it encounters in its moving neighborhood determines the interactions it is likely to participate in.

3 Implemented Aggregate Models

In the sequel we describe in some detail the derivation of three models: average dynamics, individual well-stirred dynamics and spatially-extended dynamics. All our models are in discrete time which will hopefully make them more accessible to those for whom the language of integrals is not native. For the others, note that our model corresponds to a fixed time-step simulation of ODEs.

3.1 Average Well-Stirred Dynamics

To develop the average dynamics under the well-stirred assumption.[3] we normalize the global macro-state of the system, a vector $X = (X_1, \ldots, X_n)$, into $x = (x_1, \ldots, x_n)$ with $x_i = X_i/m$ and hence $\sum x_i = 1$ (relative concentration). Let α, $0 \leq \alpha \leq 1$ be a density parameter which determines the probability of bumping into another particle in one time step. The evolution in this state space over time is the outcome of playing the following protocol at every time step. First, αS of the particles bump into others and hence follow a binary reaction rule while the remaining $(1-\alpha)S$ particles do not interact and hence follow the solitary transition function. We will derive the dynamics, which is of the general form[4] $x' = x + \Delta(x)$. For each variable, the additive change can be written as

$$\Delta(x_k) = (1-\alpha)\Delta_1(x_k) + \alpha\Delta_2(x_k)$$
$$\Delta_1(x_k) = \sum_{i=1}^{n} (x_i \cdot \delta(x_i, \perp, x_k) - x_k \cdot \delta(x_k, \perp, x_i))$$
$$\Delta_2(x_k) = \sum_{i=1}^{n}\sum_{j=1}^{n} (x_i x_j \cdot \delta(x_i, x_j, x_k) - x_k x_i \cdot \delta(x_k, x_i, x_j))$$

[3] Using PDEs one can sometimes derive average models under *non-uniform* distributions of particles in space but most chemical reaction models employ the well-stirred assumption.

[4] We use the primed notation where x stands for $x[t]$ and x' denotes $x[t+1]$.

Here, Δ_1 and Δ_2 are the expected contributions to x_k by the solitary (resp. binary) reactions, each summing up the transformations of other agents into type k minus the transformation of type k into other types. Thus, we obtain a discrete-time bilinear dynamical system, which is linear when $\alpha = 0$, see example in Table 1. As already mentioned, this deterministic dynamics tracks the evolution of the *average* concentration of particles over all individual runs.

3.2 Individual Well-Stirred Dynamics

The second model, whose average behavior is captured by the previous one, generates individual behaviors without spatial information. A micro-state of the systems is represented as a set L of particles, each denoted as (g,q) where g is the particle identifier and q is its current state.

Algorithm 1 (Individual Well-Stirred Dynamics)
Input*: A list L of particles and states*
Output *A list L' representing the next micro-state*

$L' := \emptyset$
repeat
  draw *a random particle* $(g,q) \in L$; $L := L - \{(g,q)\}$
  draw *binary/solitary with probability* α
  if *solitary* **then**
    apply solitary rule $q \xrightarrow{\perp} q'$
    $L' := L' \cup \{(g,q')\}$
  else
    draw *a random particle* $(g',q') \in L$; $L := L - \{(g',q')\}$
    apply binary rules $q \xrightarrow{q'} q''$ *and* $q' \xrightarrow{q} q'''$
    $L' := L' \cup \{(g,q''),(g',q''')\}$
  endif
until $L = \emptyset$

After each update round, particle types are counted to create macro-states. The algorithm can most likely be made more efficient by working directly on macro-states and drawing the increments of each particle type using a kind of binomial distribution that sums up the multiple coin tosses. Similar ideas underlie the τ-leaping algorithm of [5].

3.3 Individual Spatial Dynamics

Our third model does take space into account by representing each particle as (g,q,y) with y being it spatial coordinates, currently ranging over a bounded rectangle. The next state is computed in two phases that correspond to diffusion and reaction. First, each particle is displaced by a vector of random direction and magnitude (bounded by a constant s). For mathematical convenience reasons we use *periodic boundary conditions* so that when a particle crosses the boundary of the rectangle it reappears on the other side as if it was a torus. Then for each particle we compute its set of neighbors N, those residing in a ball of a pre-specified *interaction radius* r, typically in the same order of magnitude as s. If the particle has several neighbors[5] we compute the outcome of all those possible interactions and choose among them

[5]Which turns out not to be negligible with the parameters we have chosen so far which are unlike commonly-used models where the average distance between particles is orders of magnitude larger than the interaction radius r.

randomly.

Algorithm 2 (Individual Spatial Dynamics)
Input: *A list L of particles and states including planar coordinates*
Output *A list L' representing the next micro-state*

$L' := \emptyset$
foreach *particle* $(g,q,y) \in L$
  draw *randomly* $h \in [0,s]$ *and* $\theta \in [0,2\pi]$
  $y := y + (h,\theta)$
endfor
foreach $(g,q,y) \in L$
  $N := \{(g',q',y') : d(y,y') < r\}$
  if $N = \emptyset$ **then**
    apply solitary rule $q \xrightarrow{\perp} q'$
    $L' := L' \cup \{(g,q',y)\}$
  else
    $M := \emptyset$
    foreach $(g',q',y') \in N$
      apply binary rule $q \xrightarrow{q'} q''$
      $M := M \cup \{q''\}$
    endfor
    draw $q' \in M$
    $L' := L' \cup \{(g,q',y)\}$
  endif
endfor

The connection between this model, embedded in a rectangle of area W, and the non-spatial ones is made via the computation of the density factor α. The probability of a particle *not* interacting with another particle is its probability to be outside its interaction ball, that is, $1 - \frac{\pi r^2}{W}$, and the odds of not interacting with any of the other $m-1$ particles is $(1 - \frac{\pi r^2}{W})^{m-1}$ and hence $\alpha = 1 - (1 - \frac{\pi r^2}{W})^{m-1}$.

4 The Populus Toolkit: Preliminary Experiments

We developed a prototype tool called **Populus**, written in Java and Swing, for exploring such dynamics. The input to the tool is a particle automaton along with additional parameters such as the dimensions of the rectangle where particles live, the geometric step size s, the interaction radius r and the initial number of each particle type, possibly restricted to some sub rectangles. The tool simulates the three models, plotting the evolution of particle counts over time as well as animating the spatial evolution.

To demonstrate the difference between spatial and non-spatial models we simulated a system with 5 species, A, B, C, D and E to E. A and B are initially present in small quantities, 50 each, while D has 1000 instances. When A and B meet, A is transformed into an active and stable agent C which converts D's to E's . Since E is also rather stable, the emergence of C will eventually convert a large number of D's to E's. However, B is very unstable and each step it may change with probability 0.5. Hence the initial spatial distribution of A and B may influence the evolution significantly. We simulated the corresponding spatial model on a 20×20 square, starting from three different initial micro-states in all

(a) (b) (c)

Figure 1: The evolution of the 5-species system where A's and B's are initially (a) distributed uniformly in space; (b) close to each other and (c) remote from each other. The plot depict the spatial and non-spatial models with the black curve indicating the growth of E.

of which D is distributed uniformly over all the square: (a) A and B are distributed uniformly all over space; (b) Both A and B are concentrated in a unit square in the middle; (c) A and B are concentrated inside distinct unit squares far apart from each other. The results are plotted in Figure 1. As a first observation, in scenario (a) the spatial model converts D to E slower than the well-stirred one, despite the well-stirred initial condition. It is too early to speculate about the reasons but it might be that with our parameters where reaction is not slower than diffusion, a C particle converts the D's in its neighborhood and hence meets less of them than what their global concentration would imply. In scenario (b), due to the proximity of A and B there is a burst of C's at the beginning and the spatial model progresses faster than the well-stirred. Finally, when A and B are initially far apart, no C and hence no E are produced, unlike the prediction of the well-stirred model. In all those experiments the behavior of the well-stirred model was close to that of the average model.

References

[1] Philip Ball (2004): *Critical mass: How one thing leads to another*. Macmillan.

[2] Luca Bortolussi & Jane Hillston (2013): *Checking Individual Agent Behaviours in Markov Population Models by Fluid Approximation*. In: *Formal Methods for Dynamical Systems*, Springer, pp. 113–149, doi:10.1007/978-3-642-38874-3_4.

[3] Luca Cardelli (2009): *Artificial biochemistry*. In: *Algorithmic Bioprocesses*, Springer, pp. 429–462, doi:10.1007/978-3-540-88869-7_22.

[4] Daniel T Gillespie (1977): *Exact stochastic simulation of coupled chemical reactions*. *The Journal of Physical Chemistry* 81(25), pp. 2340–2361, doi:10.1021/j100540a008.

[5] Daniel T Gillespie (2001): *Approximate accelerated stochastic simulation of chemically reacting systems*. *The Journal of Chemical Physics* 115, p. 1716, doi:10.1063/1.1378322.

[6] A Agung Julius, Ádám Halász, Mahmut Selman Sakar, Harvey Rubin, Vijay Kumar & George J Pappas (2008): *Stochastic Modeling and Control of Biological Systems: The Lactose Regulation System of Escherichia Coli*. *IEEE Transactions on Automatic Control* 53, pp. 51–65, doi:10.1109/TAC.2007.911346.

[7] J-Y Le Boudec, David McDonald & Jochen Mundinger (2007): *A generic mean field convergence result for systems of interacting objects*. In: *QEST*, IEEE, pp. 3–18, doi:10.1109/QEST.2007.3.

[8] Azaria Paz (1971): *Introduction to Probabilistic Automata*. Academic Press.

[9] Michael S Samoilov & Adam P Arkin (2006): *Deviant effects in molecular reaction pathways*. Nature biotechnology 24(10), pp. 1235–1240, doi:10.1038/nbt1253.

Producing a Set of Models for the Iron Homeostasis Network

Nicolas Mobilia
UJF-Grenoble 1 / CNRS
TIMC-IMAG UMR 5525,
Grenoble, F-38041, France
nicolas.mobilia@imag.fr

Alexandre Donzé
EECS Department
University of California Berkeley
Berkeley, CA 94720 USA
donze@eecs.berkeley.edu

Jean Marc Moulis
Univ. Grenoble Alpes, Laboratory of Fundamental and Applied Bioenergetics (LBFA),
and Environmental and Systems Biology (BEeSy), Grenoble, France
Inserm, U1055, Grenoble, France
CEA-iRTSV, Grenoble, France
jean-marc.moulis@cea.fr

Éric Fanchon
UJF-Grenoble 1 / CNRS
TIMC-IMAG UMR 5525,
Grenoble, F-38041, France
eric.fanchon@imag.fr

1 Introduction

The continuous parts of complex biological systems are often modeled by use of Ordinary Differential Equations (ODE). When experimental data are available, they usually have large variability, for example due to variability in cell cultures. Since data on a given system are scarce, one generally uses data from different cell types, different organisms, or different conditions. All this translates into large parameter uncertainty. To cope with this situation and try to integrate all available data in a consistent model, we represent such data by intervals rather than single numerical values. The ranges of the intervals vary depending on the type of experiment and the nature of the experimental system. Some parameters or concentrations are not known at all and are initially defined to belong to the physiological domain. This set of intervals define the search space. Other experimental data are expressed in terms of inequalities involving derived quantities. Our goal is to build a set of models that satisfy all the constraints deduced from experiments, and to analyze the salient features of the dynamics of this set of models.

We present a method for modeling biological systems which combines formal techniques on intervals, numerical simulations and formal verification of STL (Signal Temporal Logic) formula. This allows us to consider intervals for each parameter and to describe the expected behavior of the model. We apply this method to the modeling of the cellular iron homeostasis network in precursors of erythroid cells. A core model [5] has been presented previously. Herein, we describe a more evolved model in which the regulation mechanism acting at the translational level is explicitly considered. This leads to a larger model with more parameters and the integration of newly obtained experimental data. This new model provides a more detailed description of the regulatory mechanism, including quantitative considerations pertaining to the involved species, and it should allow us to more precisely address pending biological questions. The higher level of complexity of this model, compared to the core model, required the development of a method to characterize efficiently steady states.

In Section 2, we describe the iron homeostasis network, and, in Section 3, the corresponding model. Then, in the Section 4, we describe the method used. We finally explain the work that remains to be done and conclude.

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 92–98, doi:10.4204/EPTCS.125.7

2 Biological system

Iron is an essential element for mammalian cells (eg. hemoglobin contains iron), but if present in too high quantity iron has a deleterious effect. The level of available iron is thus finely tuned in mammalian cells. Our goal is to describe and understand this regulatory mechanism. The regulatory network, described in Figure 1, is composed of fifteen species. The species *Fe* (pool of available iron), *IRP* (Iron Regulatory Protein), *Ft* (ferritin), *FPN1a* (ferroportin) and *TfR*1 (transferrin receptor) were present in the previous model [5]. The other ones are the mRNA of these proteins either in a free form or in a form complexed with an IRP. The central regulatory mechanism, based on the IRP is described in Mobilia & al [5].

In a nutshell, the available data belong to several categories. A qualitative description of the dynamics, obtained from a large body of biological experiments is the following: if the amount of iron is sufficient, the cell is in a steady state. From this state, if an iron input cut-off occurs, the amount of iron in cells decreases and the IRP are activated, leading to increased binding of IRP to IRE-containing mRNAs. New kinetic parameters have been measured, as well as absolute mRNA concentrations in the iron-replete regime. Our aim is to build models which simulate the behavior of the biological system. From the iron-replete steady state, the evolution of the concentrations of the different species must be qualitatively reproduced upon cutting off the iron supply.

3 Model

We model this system with fifteen differential equations. These equations contain 28 parameters. In the following sub-section, we exhibit the equations related to the ferritin, the transferrin receptor and the IRP. The conventions for the parameter names are the following: a parameter named p_X represents the transcription speed of the mRNA of X (units: mol/(L·s)); t_X represents the reaction rate constant for X mRNA translation (units: s^{-1}) ; dp_X represents the degradation rate of the protein X (units: s^{-1}); dr_X represents the degradation rate of the mRNA of X (units: s^{-1}); drs_X represents the degradation rate of the mRNA of X, when this mRNA contains an IRE in the 3'-UTR region and is bound to an IRP (units: s^{-1}). The species ending with the subscript $\_p$ represent proteins, while the ones ending with the subscript $\_f$ (resp. $\_b$) represent free (resp. bound) mRNA concentration (units: mol/L).

3.1 Ferritin equations

The ferritin mRNAs contain an IRE in the 5'-UTR region, so the translation speed is proportional to the free mRNA concentration. This is described by the first term in Equation (1). Moreover, as a ferritin is constituted by 24 sub-units, a factor 1/24 appears in this term. The second term represents the spontaneous degradation of the ferritin.

$$\frac{\mathrm{d}Ft\_p}{\mathrm{d}t} = (t\_Ft/24)\cdot Ft\_f - dp\_Ft\cdot Ft\_p \tag{1}$$

The free ferritin mRNA concentration, described in Equation (2), depends on four terms. The first one is the transcription speed p_Ft. The second one represents the complexation of this mRNA with an IRP (the parameter *ka* is the complexation second-order reaction rate parameter), while the third one represents the decomplexation of mRNA:IRP complex. The parameter K_d is equal to the ratio kd/ka, where *kd* is the decomplexation first-order reaction rate parameter. The last term represents the spontaneous

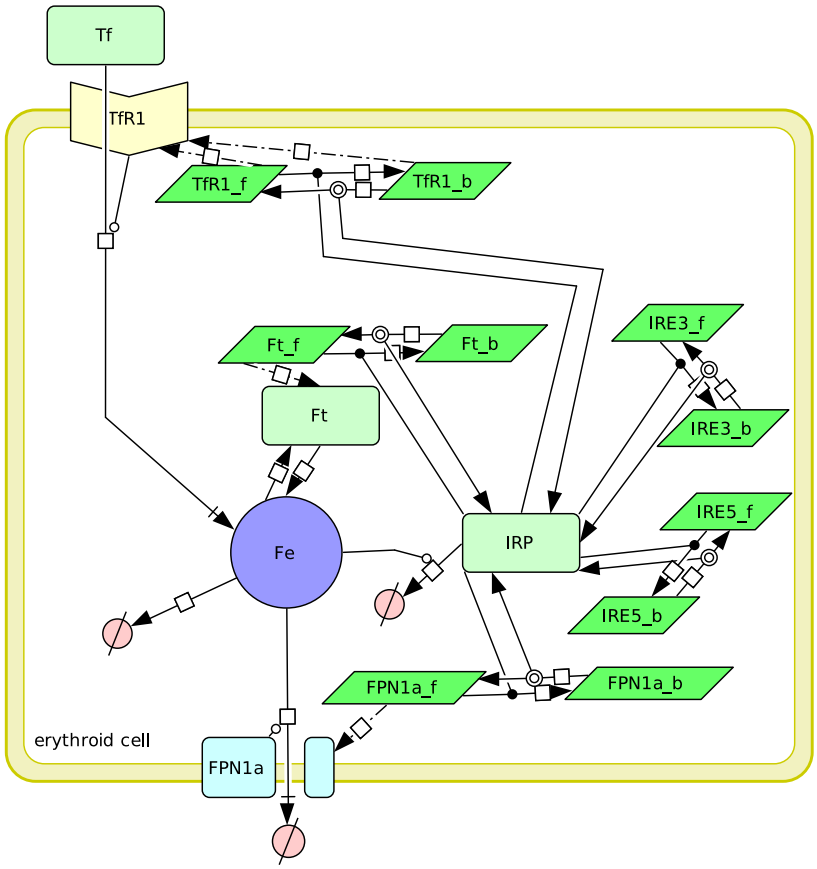


Figure 1: Schematic representation of the main biological processes involved in the cellular control of iron concentration. This diagram was drawn with the software CellDesigner [2]. The dashed arrows represent translation of mRNA into proteins. The lines ending with a combined perpendicular stroke and arrow represent iron transport through membranes. The regular arrows leading to an empty set symbol which indicate either degradation (for *IRP*) or internal consumption (for iron). Moreover, the multi-arrows containing a black dot represent complexation while the ones with two empty circles mean decomplexation. Finally, the two regular arrows represent the loading/unloading of iron into/from the ferritins. The rounded rectangles represent proteins, the parallelograms represent mRNA, and the circle labeled *Fe* represents the pool of available iron. The concave hexagon represents the transferrin receptor. The species $IRE5\_f$ (resp. $IRE3\_f$) and $IRE5\_b$ (resp. $IRE3\_b$) represent all the mRNA having an IRP binding site (called IRE) in the 5'-UTR (resp. 3'-UTR) excepted the ones explicitly drawn.

degradation of the mRNA.

$$\frac{\mathrm{d}Ft\_f}{\mathrm{d}t} = p\_Ft - ka \cdot Ft\_f \cdot IRP + ka \cdot K_d \cdot Ft\_b - dr\_Ft \cdot Ft\_f \tag{2}$$

Finally, the bound ferritin mRNA concentration is described in Equation (3). This equation is composed of three terms. The first two, describing the complexation and the decomplexation, have the same meaning as in the equation of the free ferritin mRNA. The last term describes the spontaneous degrada-

tion of the mRNA.

$$\frac{\mathrm{d}Ft\_b}{\mathrm{d}t} = ka \cdot Ft\_f \cdot IRP - ka \cdot K_d \cdot Ft\_b - dr\_Ft \cdot Ft\_b \tag{3}$$

3.2 Transferrin receptor equation

The transferrin receptor mRNA contains five IREs in its 3'-UTR region. Here, we make a simplification and consider that TfR1 mRNA contains only one IRE. Equation (4) describes the transferrin receptor concentration. It contains a translation term and a spontaneous degradation term. As the IRE is located in the 3'-UTR region, both free and bound mRNA are translated into proteins. Moreover, because the receptor is a dimer, a factor 1/2 appears in the first term.

$$\frac{\mathrm{d}TfR1\_p}{\mathrm{d}t} = (t\_TfR1/2) \cdot (TfR1\_f + TfR1\_b) - dp\_TfR1 \cdot TfR1\_p \tag{4}$$

The equation of the free transferrin receptor mRNA is very similar to the free ferritin mRNA. This equation is shown in Equation (5) and is composed of a translation term, a complexation term, a decomplexation term and a spontaneous degradation term.

$$\frac{\mathrm{d}TfR1\_f}{\mathrm{d}t} = p\_TfR1 - ka \cdot TfR1\_f \cdot IRP + ka \cdot K_d \cdot TfR1\_b - dr\_TfR1 \cdot TfR1\_f \tag{5}$$

Equation (6) describes the concentration of the transferrin receptor mRNA complexed with an IRP. This equation is similar to the bound ferritin mRNA, except that the binding of an IRP on this mRNA leads to the mRNA stabilization. To model this mechanism, a specific degradation rate parameter (drs_TfR1) is considered. This parameter is lower than the free mRNA degradation rate parameter (dr_TfR1).

$$\frac{\mathrm{d}TfR1\_b}{\mathrm{d}t} = ka \cdot TfR1\_f \cdot IRP - ka \cdot K_d \cdot TfR1\_b - drs\_TfR1 \cdot TfR1\_b \tag{6}$$

3.3 IRP equation

The IRP equation is described in Equation (7).

$$\begin{aligned} \frac{\mathrm{d}IRP}{\mathrm{d}t} \quad = \quad & -\left(Ft\_f + FPN1a\_f + TfR1\_f + IRE3\_f + IRE5\_f\right) \cdot ka \cdot IRP \\ & + \left(Ft\_b + FPN1a\_b + TfR1\_b + IRE3\_b + IRE5\_b\right) \cdot ka \cdot K_d \\ & - k_{Fe \to IRP} \cdot sig^{+}(Fe, \theta_{Fe \to IRP}) \cdot IRP - dp\_IRP \cdot IRP \end{aligned} \tag{7}$$

The first line of this equation describes the complexation of free mRNAs and IRP for all mRNAs, while the second line describes the decomplexation of the bound mRNAs. The last line describes IRP inactivation. This inactivation is described by a constant basal term for degradation ($dp\_IRP \cdot IRP$) and the iron-triggered regulation ($k_{Fe \to IRP} \cdot sig^{+}(Fe, \theta_{Fe \to IRP}) \cdot IRP$). Then, if the iron level is significantly below the threshold $\theta_{Fe \to IRP}$, the degradation rate is $dp\_IRP \cdot IRP$. Otherwise, if the iron concentration is significantly above this threshold, the degradation rate is $(k_{Fe \to IRP} + dp\_IRP) \cdot IRP$, where $k_{Fe \to IRP}$ is the parameter describing the inhibition of IRP by iron.

3.4 Other equations

The iron equation is the same than in the previous model [5]. The eight remaining equations are very similar to those shown above. The equations for ferroportin and $IRE5$ are similar to that of ferritin, and the equations for the $IRE3$ species are similar to that of the transferrin receptor.

3.5 Data

For each parameter, we consider an interval deduced from biological data or incorporating meaningful values. For example, in K562 cells, the ferritin half-life is 11 hours [4]. We deduce that the parameter $dp_{\_Ft}$ is included in the interval [3.8e-6, 3.8e-5] s^{-1}.

Moreover, some data are expressed as relations between parameters. To give an example, our recent data indicate the ferritin mRNA concentration largely exceed that of the other IRP targets in proliferating cells. The total ferritin (resp. IRE5) mRNA concentration at steady state being equal to $p_{\_Ft}/dr_{\_Ft}$ (resp. $p_{\_IRE5}/dr_{\_IRE5}$), it follows the relation described in Equation (8).

$$\frac{p_{\_Ft}}{dr_{\_Ft}} > \frac{p_{\_IRE5}}{dr_{\_IRE5}} \tag{8}$$

We can also note that the lower degradation rate due to the binding of IRP on IRE in 3'-UTR mRNA region translates into Equation (9) and Equation (10).

$$drs_{\_TfR1} < dr_{\_TfR1} \tag{9}$$

$$drs_{\_IRE3} < dr_{\_IRE3} \tag{10}$$

In addition, some relations describe data related to the stationary state. For example, the degradation rate of total TfR1 mRNA belong to the interval $[7.0\times10^{-6}, 7.0\times10^{-5}]$ s^{-1} [7][6]. This describe the sum of the degradation of both free and complexed mRNA and translates into the equation (11). The superscript ^{eq} indicates that we consider the steady state concentration.

$$\frac{dr_{\_TfR1}\cdot TfR1^{eq}_{\_f} + drs_{\_TfR1}\cdot TfR1^{eq}_{\_b}}{TfR1^{eq}_{\_b} + TfR1^{eq}_{\_f}} \in [7.0\times10^{-6},\ 7.0\times10^{-5}]\ s^{-1} \tag{11}$$

The last kind of data is related to the dynamic of the system when an iron cut-off happens. The modeling of these data using STL formula is described in Mobilia & al [5].

4 Method

The set of intervals and constraints can be divided in two: those pertaining to the iron-replete steady state, and those pertaining to the cell response to iron shortage. In our previous work [5], we first reduced the search space by using the interval solver Realpaver [3]. Then, we represented formally the whole set of constraints as an STL formula and devised a search algorithm to satisfy it, based on the tool Breach [1]. Basically, a point is randomly drawn in the search space, a simulation is performed and the STL formula is evaluated. In the present more complete model, no iron-replete steady state was initially found following the same procedure. In addition, this failure did not instruct us on the origin of the problem.

To cope with this limitation, we improved the method in two ways. (i) In the first step, the interval solver Realpaver allowed to reduce the intervals by propagating the constraints. Instead of being solely a hyper-rectangle as previously, the search space was allowed to be a union of hyper-rectangles thus reducing it more efficiently. In case the interval solver found an inconsistency, we improved the method by looking for the smallest sets of constraints that have to be lifted to release the inconsistency. This information gives insight when one wants to revisit the model and the data used. (ii) We decomposed the

search algorithm in two parts. We had developed an algorithm to generate efficiently a large number of explicit solutions (steady states concentrations and model parameters) satisfying the constraints of a stable steady state (the set of constraint contains algebraic equations and inequalities involving polynomial expressions). These explicit solutions, that were prerequisite to perform simulation of the dynamics of the system, were then fed into our Breach-based procedure in order to search models satisfying the STL formula specifying the cell response to iron deprivation.

The unknowns of the problem are the model parameters and the concentrations in the iron-replete steady state. The methodology proceeds basically as follows:

1. perform interval reduction with Realpaver;
2. select a subset of unknowns to be sampled (we start with unknowns within a narrow interval, then other criteria are used to decouple the equations and to optimize the following step);
3. for each sample of this subset of unknowns:
4. replace the instantiated unknowns in the algebraic equations and perform deductions (new unknowns can get instantiated);
5. check domain of newly instantiated unknowns;
6. check the validity of inequality constraints as soon as possible;
7. for each sub-problem (i.e.: set of decoupled constraints), apply this algorithm;
8. loop to step 3. until all samples are tried;
9. loop to step 2.

The basic principles underlying this search are to decouple the constraints in order to solve subproblems, and to identify the hardest sub-problems (most constrained) and try to solve them in priority. The aim is of course to trim the branches of the search tree as soon as possible.

The interval solver Realpaver, used during the first step, allows to reduce the intervals by propagating the constraints. The result is an hyper-rectangle (or an union of hyper-rectangles) containing the solutions, if they exist. The existence of solutions is not guaranteed, but it is certain that there are no solutions outside of the volume given by Realpaver. Consequently, this step is important to reduce the search space. Nevertheless, the remaining space may be very large with regard to the solution space. A simple example to illustrate this aspect is the following: consider two unknowns x_1 and x_2, within the [0, 1] interval, and the constraint $abs(x_1 - x_2) < eps$, with eps small compared to x_1 and x_2. If the solution space is defined by one box, Realpaver cannot reduce the search space. Considering a union of boxes allows a reduction of the search space. As it is hard to find explicit solutions, we say informally that the constraint is hard to satisfy (the smaller eps, the harder it is).

The constraint system is constituted of algebraic equations and inequalities. For the majority of them, the algebraic equations are used to deduce the values of unknowns, and are thus automatically satisfied. Inequalities are checked *a posteriori*. To be efficient, it is important to check an inequality as soon as all the unknowns in it have been instantiated. For efficiency reasons, redundant constraints are also added. The aim is to add constraints simpler than the initial ones , which can be checked early in the search process. Typically, from the constraints described by Equation (11), we straightforwardly deduce that $t\_TfR1 \cdot TfR1^{eq}_{\_f} < 5.5 \times 10^{-13}$ and that $t\_TfR1 \cdot TfR1^{eq}_{\_b} < 5.5 \times 10^{-13}$. Even if $TfR1\_f$ or $TfR1\_b$ is not instantiated, one of these two constraints can be checked and may invalidate this partial instantiation.

When applying this algorithm, we store the result of each verification of domain and check of constraints (whether the constraint or domain is verified or not). When no solution is found, this may be due

to different constraints. This information thus provide the level of difficulty to satisfy each constraint. This may help to manually found inconsistencies between constraints that were not automatically found by Realpaver.

This methodology either provides us valid sets of values, or indicates the hardest subset of constraints to satisfy. We applied it on the iron homeostasis network. No solutions could be found, which is not a proof of nonexistence, but the subset of hard constraints identified allowed us to prove that there was indeed a contradiction. After revision of this part of the model (namely: removing one non-reliable constraint and extending some intervals), the procedure then generated thousands of valid steady states in a short execution time.

5 Conclusion

This evolved model describes in a more realistic way the action of IRP and takes into account the fact that their effect depends on the location of the binding site on mRNA. Moreover, it easily incorporates new information obtained on the system.

Nevertheless, some work still remains to be done in order to completely automate this search, and to interface the steady state search with the part dealing with dynamical behavior (specified with an STL formula).

Here we described our approach to nicely integrate different kinds of biological data, combining an interval solver, simulations, and temporal STL formula verification. We are applying it on a new model of iron homeostasis in mammalian cells.

Acknowledgements

This work was supported by Microsoft Research through its PhD Scholarship Programme and by the following grants: Région Rhône-Alpes Cible 2010, IXXI-Spring 2012, and Univ. J. Fourier-Agir 2013.

References

[1] A. Donzé (2010): *Breach, A Toolbox for Verification and Parameter Synthesis of Hybrid Systems*. In: *CAV*, pp. 167–170, doi:10.1007/978-3-642-14295-6_17.

[2] A. Funahashi, M. Morohashi, H. Kitano & N. Tanimura (2003): *CellDesigner: a process diagram editor for gene-regulatory and biochemical networks*. *BIOSILICO* 1(5), pp. 159–162, doi:10.1016/S1478-5382(03)02370-9.

[3] L. Granvilliers & F. Benhamou (2006): *Algorithm 852: RealPaver: an interval solver using constraint satisfaction techniques*. *ACM Transactions on Mathematical Software* 32(1), pp. 138–156, doi:10.1145/1132973.1132980.

[4] T. Z. Kidane, E. Sauble & M. C. Linder (2006): *Release of iron from ferritin requires lysosomal activity*. *Am. J. Physiol., Cell Physiol.* 291(3), pp. C445–455, doi:10.1152/ajpcell.00505.2005.

[5] N. Mobilia, A. Donzé, J. M. Moulis & É. Fanchon (2012): *A Model of the Cellular Iron Homeostasis Network Using Semi-Formal Methods for Parameter Space Exploration*. *EPTCS* 92, pp. 42–57, doi:10.4204/EPTCS.92.4.

[6] C. Seiser, M. Posch, N. Thompson & L. C. Kuhn (1995): *Effect of transcription inhibitors on the iron-dependent degradation of transferrin receptor mRNA*. *J. Biol. Chem.* 270(49), pp. 29400–29406, doi:10.1074/jbc.270.49.29400.

[7] L. V. Sharova, A. A. Sharov, T. Nedorezov, Y. Piao, N. Shaik & M. S. Ko (2009): *Database for mRNA half-life of 19 977 genes obtained by DNA microarray analysis of pluripotent and differentiating mouse embryonic stem cells*. *DNA Res.* 16(1), pp. 45–58, doi:10.1093/dnares/dsn030.

A Hybrid Model of a Genetic Regulatory Network in Mammalian Sclera

Qin Shu
Department of Aerospace and Mechanical Engineering
University of Arizona, Tucson, Arizona, USA
shuq@email.arizona.edu

Diana Catalina Ardila
Department of Aerospace and Mechanical Engineering
University of Arizona, Tucson, Arizona, USA
cardila@email.arizona.edu

Ricardo G. Sanfelice
Department of Aerospace and Mechanical Engineering
University of Arizona, Tucson, Arizona, USA
sricardo@u.arizona.edu

Jonathan P. Vande Geest
Department of Aerospace and Mechanical Engineering
Biomedical Engineering, BIO5 Institute
University of Arizona, Tucson, Arizona, USA
jpv1@email.arizona.edu

Myopia in human and animals is caused by the axial elongation of the eye and is closely linked to the thinning of the sclera which supports the eye tissue. This thinning has been correlated with the overproduction of matrix metalloproteinase (MMP-2), an enzyme that degrades the collagen structure of the sclera. In this short paper, we propose a descriptive model of a regulatory network with hysteresis, which seems necessary for creating oscillatory behavior in the hybrid model between MMP-2, MT1-MMP and TIMP-2. Numerical results provide insight on the type of equilibria present in the system.

1 Introduction

This short paper presents a descriptive model of a genetic regulatory network in the mammalian sclera using the formalism of hybrid dynamical systems. This model is deduced from experimental observations of enzyme interactions that govern the remodeling of the collagen tissue in the sclera. A number of research publications indicate that myopia is closely related to an unbalanced remodeling in sclera [12, 22]. Myopia is an optical condition in which the eye grows abnormally in the axial direction, causing images to form in front of the retina compared to on the retina, as it normally occurs [9, 12, 13, 18]. The excessive length of the eye is driven by the remodeling of the scleral extra cellular matrix (ECM) (e.g., loss of Type I collagen, COL1A1), leading the progressive thinning of this tissue [1, 11, 12]. Scleral remodeling is regulated by a large number of growth factors, membrane receptors, proteases, and protease inhibitors, which work in concert to optimize the dynamic synthesis and degradation of COL1A1[3, 11, 27]. One of the most studied actors in sclera remodeling is the Type II matrix metalloproteinase (MMP-2), because of its role in the degradation of COL1A1 [6, 12, 16]. MMP-2 is regulated by the Type II tissue inhibitor of the matrix metalloproteinases (TIMP-2), and when the two enzymes are properly balanced, the sclera develops normally. MMP-2 regulation by TIMP-2 shows a particular mechanism in which TIMP-2 not only inhibits the proteolytic activity of MMP-2, but is also necessary for the production of this metalloproteinase in its active form [12, 24, 26]. Such a mechanism is very important for the balance between COL1A1 production and degradation in sclera, and hence, should play a key role in a model of a genetic regulatory network in this tissue.

The remainder of the paper is organized as follows. Section 2 introduces the mechanisms governing the regulatory network of interest and proposes a hybrid system model. Section 3 presents results from simulations of the proposed model, which, for a particular set of parameters, identify both isolated equilibria and limit cycles. Final remarks and a discussion of the current efforts appear in Section 4.

T. Dang and C. Piazza (Eds.): HSB 2013
EPTCS 125, 2013, pp. 99–105, doi:10.4204/EPTCS.125.8

2 Modeling

We develop a model of a regulatory network in mammalian sclera from the following experimental observations. Sufficient high levels of MMP-2 protein cause the expression of TIMP-2 [12, 21] (considering expression as the result of transcription, translation and activation of the protein latent form). When the concentration of TIMP-2 exceeds a minimum threshold, this protein indirectly modulates the increment of MMP-2: TIMP-2 triggers the expression of active membrane-type I matrix metalloproteinase (MT1-MMP) [12, 21, 24], which is necessary for the activation of latent MMP-2 [12, 24, 26]. When the concentration of TIMP-2 protein is sufficiently high, TIMP-2 inhibits the proteolytic activity of MMP-2 and MT1-MMP [12, 19, 21, 24, 26]. As we mentioned above, MT1-MMP triggers the activation of latent MMP-2 when sufficiently high [5, 15]; therefore, by blocking MT1-MMP, TIMP-2 is also inhibiting the activation of latent MMP-2 [12, 24, 26]. In fact, [12, 26] argue that the increased TIMP-2 mRNA and protein levels are significant as TIMP-2 is not only a protein inhibitor of both the active and latent form of MMP-2 but also paradoxically essential for the MT1-MMP dependent activation of MMP-2. The genetic network capturing these mechanisms is depicted in Figure 1.

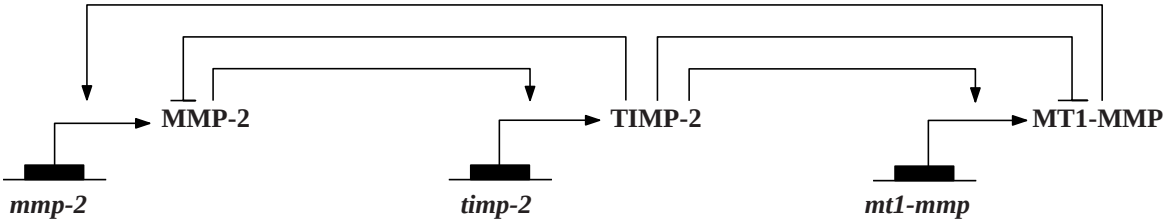


Figure 1: *Proposed genetic regulatory network for sclera. Lowercase names refer to genes, uppercase names refer to proteins. Lines ending in arrows represent expression triggers and lines ending in flatheads refer to inhibition triggers.*

The mechanisms described above can be encoded in a piecewise-linear differential equation following the modeling technique in [6, 14]. However, the resulting model of the genetic network in sclera would not incorporate hysteresis, which is a key player in genetic regulatory networks [2, 7, 10, 20]. To incorporate hysteresis, we follow the approach in [23] and propose a hybrid system model in the framework of [4]. To this end, we define the state of the hybrid system as

$$z = [x_1, x_2, x_3, q_1, q_2, q_3, q_4]^\top \tag{1}$$

where $z \in Z := \mathbb{R}^3_{\geq 0} \times \{0,1\}^4$. The continuous states x_1, x_2, x_3 represent the protein concentrations, where x_1 represents the protein concentration of TIMP-2, x_2 the concentration of MT1-MMP, and x_3 the concentration of MMP-2. Positive constants $\gamma_1, \gamma_2, \gamma_3$ define the decay rates and k_1, k_2, k_3 define the growth rates, respectively, for each of the concentrations. The discrete states (logic variables) q_1, q_2, q_3, q_4 define the boolean value (1 or 0) of the hysteresis functions associated with each of the thresholds $\theta_1, \theta_2, \theta_3, \theta_4$ and the hysteresis half-width constants h_1, h_2, h_3, h_4 associated with each of the thresholds, respectively.

Threshold	Definition
θ_1	TIMP-2 level for MT1-MMP expression
θ_2	MT1-MMP level for MMP-2 expression
θ_3	TIMP-2 level for MT1-MMP/MMP-2 inhibition
θ_4	MMP-2 level for TIMP-2 expression

Table 1: Definition of protein thresholds in the genetic regulatory network for sclera.

Following the definitions in Table 1, the thresholds and hysteresis half-width constants are used to determine when, for current values of the protein concentrations and of the logic variables, changes of the logic variables should occur. For instance, according to the mechanisms described above, if $q_4 = 0$ and x_3 is small, then x_1 should decay according to its decay rate γ_1. However, if $q_4 = 0$ and x_3 becomes large (i.e., the concentration of MMP-2 is large) then q_4 should change to 1 and x_1 should be expressed according to its own growth rate k_1. The continuous evolution of x_3 can be captured mathematically by the differential equation

$$\dot{x}_1 = k_1 q_4 - \gamma_1 x_1$$

while the discrete change of q_4 can be captured by the difference equation

$$q_4^+ = 1 - q_4 \qquad \text{when} \qquad q_4 = 0 \text{ and } x_3 \geq \theta_4 + h_4, \qquad \text{or} \quad q_4 = 1 \text{ and } x_3 \leq \theta_4 - h_4$$

In this way, the *flow map* of the hybrid system defining the continuous dynamics of z is given by

$$F(z) := \begin{bmatrix} k_1 q_4 - \gamma_1 x_1 \\ k_2 q_1 (1 - q_3) - \gamma_2 x_2 \\ k_3 q_2 (1 - q_3) - \gamma_3 x_3 \\ 0_{4\times 1} \end{bmatrix} \tag{2}$$

Changes of the variables occur when z is in the *jump set*, which is conveniently written as

$$D := \bigcup_{i=1}^{4} D_i$$

where

$$\begin{aligned} D_1 &:= \{z : q_1 = 1, x_1 \leq \theta_1 - h_1\} \cup \{z : q_1 = 0, x_1 \geq \theta_1 + h_1\} \\ D_2 &:= \{z : q_2 = 1, x_2 \leq \theta_2 - h_2\} \cup \{z : q_2 = 0, x_2 \geq \theta_2 + h_2\} \\ D_3 &:= \{z : q_3 = 1, x_1 \leq \theta_3 - h_3\} \cup \{z : q_3 = 0, x_1 \geq \theta_3 + h_3\} \\ D_4 &:= \{z : q_4 = 1, x_3 \leq \theta_4 - h_4\} \cup \{z : q_4 = 0, x_3 \geq \theta_4 + h_4\} \end{aligned}$$

The right-hand side of the difference equation discretely updating the logic variables is given by the *jump map*

$$G(z) = \begin{cases} g_1(z) & z \in D_1 \setminus (D_2 \cup D_3 \cup D_4) \\ g_2(z) & z \in D_2 \setminus (D_1 \cup D_3 \cup D_4) \\ g_3(z) & z \in D_3 \setminus (D_1 \cup D_2 \cup D_4) \\ g_4(z) & z \in D_4 \setminus (D_1 \cup D_2 \cup D_3) \\ \hat{g}(z) & z \in D_1 \cap D_2 \cap D_3 \cap D_4 \end{cases} \tag{3}$$

where

$$g_1(z) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 - q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}, \quad g_2(z) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ q_1 \\ 1 - q_2 \\ q_3 \\ q_4 \end{bmatrix}, \quad g_3(z) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ q_1 \\ q_2 \\ 1 - q_3 \\ q_4 \end{bmatrix}, \quad g_4(z) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ q_1 \\ q_2 \\ q_3 \\ 1 - q_4 \end{bmatrix}$$

and $\hat{g}(z) = \{g_1(z), g_2(z), g_3(z), g_4(z)\}$. Note that x_1 and its associated logic variables q_1 and q_3 are the only "inputs" to the dynamics of x_2, which suggests that $\theta_1 + h_1 < \theta_3 - h_3$ should hold for x_2 to ever grow. Moreover, x_1 and its associated logic variable q_3 are "inputs" to the dynamics of x_3, while x_3 and q_4 are "inputs" to the dynamics of x_1, in what resembles to a feedback interconnection.

With the definitions mentioned before, a hybrid system $\mathscr{H} = (F, C, G, D)$ in the framework of [4] capturing the mechanism in the genetic network of sclera with hysteresis is given as

$$\mathscr{H}: \quad z \in Z \quad \begin{cases} \dot{z} = F(z) & z \in C := \overline{Z \setminus D} \\ z^+ \in G(z) & z \in D \end{cases} \tag{4}$$

3 Simulation Results

We simulate the hybrid model of the scleral genetic network within a Matlab/Simulink toolbox [20]. Unless otherwise stated, the growth rate k_i and decay rates γ_i, $i = 1,2,3$ for the three proteins are identically set to 1 and the hysteresis h_i, $i = 1,2,3,4$ are set to 0.01.

3.1 Isolated Equilibrium Points

Figure 2(a) and Figure 2(b) present simulation results in which the hybrid system evolves to the equilibrium point at $x^* = (0,0,0)$. Under these initial conditions and protein thresholds, the concentration of TIMP-2 (x_1) is not sufficiently high to permit continued expression of the MT1-MMP (x_2) and MMP-2 (x_3) genes. The protein concentration associated with the MMP-2 gene continues to grow, but when the MT1-MMP gene is inhibited, MMP-2 will become inhibited with time.

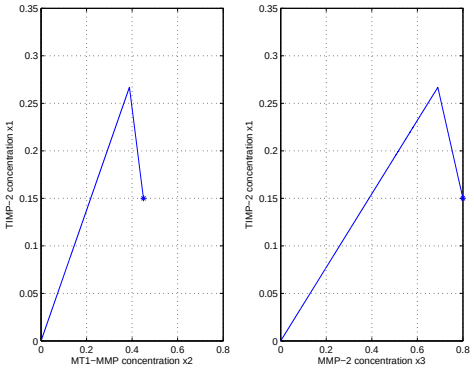
Figure 2(c) and Figure 2(d) show that the solution of the hybrid system goes toward the equilibrium point at $x^* = \left(\frac{k_1}{\gamma_1}, \frac{k_2}{\gamma_2}, \frac{k_3}{\gamma_3}\right)$. With the given initial conditions and parameters, the concentration of TIMP-2 (x_1) is not high enough to inhibit the expression of the MT1-MMP (x_2) and MMP-2 (x_3) genes. This situation can be a cause of high myopia [12, 17].

3.2 Limit cycles

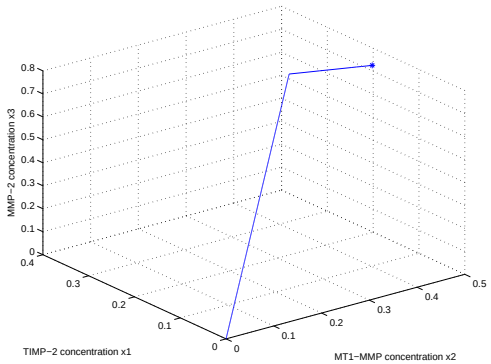
Figure 3(a) and Figure 3(b) illustrates the oscillatory behavior in the hybrid system when the concentration of TIMP-2 exceeds θ_1 and the concentration of MMP-2 exceeds θ_4 recurrently. In this scenario, the discrete state behavior stabilizes to a periodic orbit. It is apparent that the TIMP-2 protein as modeled here has a stabilizing effect on the other two protein concentrations when it is at a sufficiently high level. In this scenario, the sclera develops normally. To illustrate that such normal development of the sclera is only possible when hysteresis is present, the previous simulation is repeated for half-width hysteresis constants equal to zero. Figure 3(c) and Figure 3(d) show the corresponding system response. The solution to the hybrid system converges to an isolated equilibrium point.

4 Conclusion

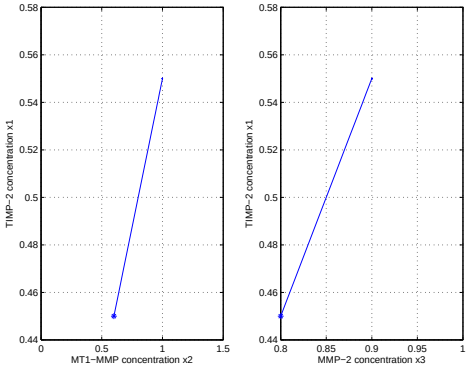
A mathematical model of a regulatory network with hysteresis to describe the mechanisms in the mammalian sclera was introduced. The model captures the interaction between MMP-2, MT1-MMP, and TIMP-2. Numerical results indicate that the system can have both isolated equilibria and limit cycles in the 3-dimensional space of protein concentrations. For the arbitrarily chosen parameters, numerical results seem to suggest that hysteresis is needed for normal development of sclera. Current efforts include characterizing the type of equilibria in terms of the values of the systems constants using the hybrid



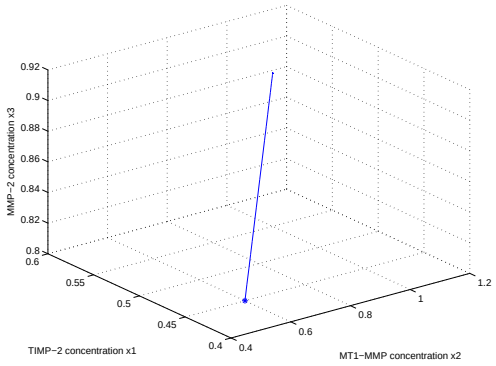
(a) $\theta_1 = 0.4, \theta_2 = 0.5, \theta_3 = 0.6, \theta_4 = 0.7, x_1(0) = 0.15, x_2(0) = 0.45, x_3(0) = 0.8, q_1(0) = 1, q_2(0) = 1, q_3(0) = 0, q_4(0) = 1$, $*$ is the initial point.



(b) $\theta_1 = 0.4, \theta_2 = 0.5, \theta_3 = 0.6, \theta_4 = 0.7, x_1(0) = 0.15, x_2(0) = 0.45, x_3(0) = 0.8, q_1(0) = 1, q_2(0) = 1, q_3(0) = 0, q_4(0) = 1$, $*$ is the initial point.



(c) $k_1 = 0.55, k_3 = 0.9, \theta_1 = 0.4, \theta_2 = 0.5, \theta_3 = 0.6, \theta_4 = 0.7, x_1(0) = 0.45, x_2(0) = 0.6, x_3(0) = 0.8, q_1(0) = 1, q_2(0) = 1, q_3(0) = 0, q_4(0) = 1$, $*$ is the initial point.



(d) $k_1 = 0.55, k_3 = 0.9, \theta_1 = 0.4, \theta_2 = 0.5, \theta_3 = 0.6, \theta_4 = 0.7, x_1(0) = 0.45, x_2(0) = 0.6, x_3(0) = 0.8, q_1(0) = 1, q_2(0) = 1, q_3(0) = 0, q_4(0) = 1$, $*$ is the initial point.

Figure 2: Solutions to $\mathscr{H}$ for different parameters and initial conditions. For the chosen values, solutions converge to isolated equilibrium points.

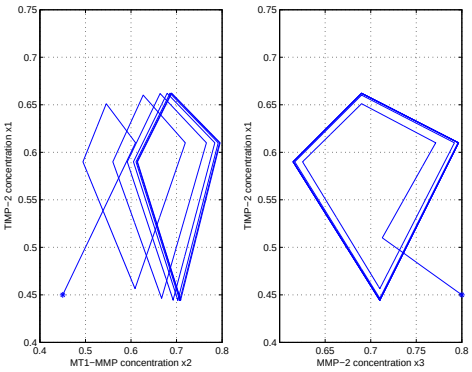
systems techniques employed in [23] and the design of in-vivo experiments to identify the parameters of the genetic model.

5 Acknowledgments

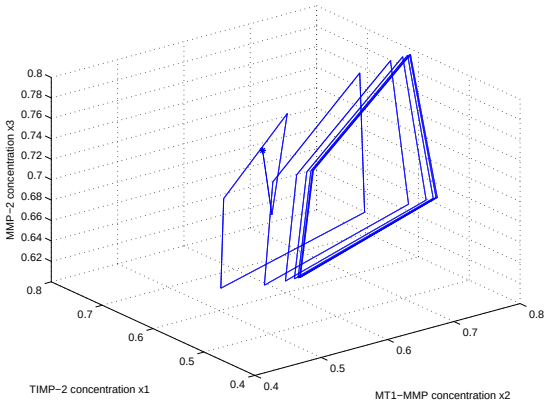
Research by D. C. Ardila and J. P. Vande Geest has been partially supported by the NIH research grant 1RO11EY020890-02A1. Research by R. G. Sanfelice has been partially supported by the National Science Foundation under CAREER Grant no. ECS-1150306 and by the Air Force Office of Scientific Research under YIP Grant no. FA9550-12-1-0366.

References

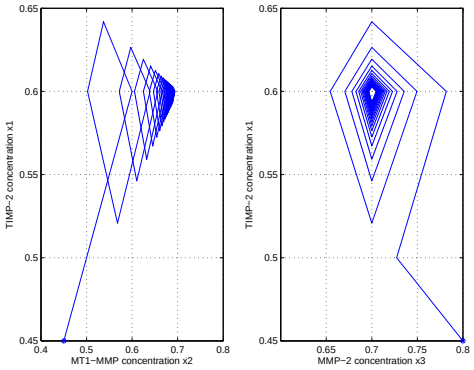
[1] S. Backhouse & J. R. Phillips (2010): *Effect of induced myopia on scleral myofibroblasts and in vivo ocular biomechanical compliance in the Guinea pig*. *Investigative Ophthalmology & Visual Science* 51(12), pp.



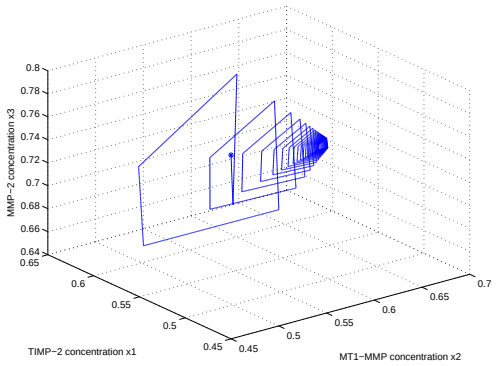
(a) $h_i = 0.01$ for each $i = 1,2,3,4$, $*$ is the initial point.



(b) $h_i = 0.01$ for each $i = 1,2,3,4$, $*$ is the initial point.



(c) $h_i = 0$ for each $i = 1,2,3,4$, $*$ is the initial point.



(d) $h_i = 0$ for each $i = 1,2,3,4$, $*$ is the initial point.

Figure 3: Solutions to $\mathscr{H}$ for $\theta_1 = 0.4, \theta_2 = 0.5, \theta_3 = 0.6, \theta_4 = 0.7, x_1(0) = 0.45, x_2(0) = 0.45, x_3(0) = 0.8, q_1(0) = 1, q_2(0) = 1, q_3(0) = 0, q_4(0) = 1$ with nonzero and zero h_i for each $i = 1,2,3,4$.

6162–6171, doi:10.1167/iovs.10-5387.

[2] J. Das, M. Ho, J. Zikherman, C. Govern, M. Yang, A. Weiss, A. K. Chakraborty & J. P. Roose (2009): *Digital Signaling and Hysteresis Characterize Ras Activation in Lymphoid Cells*. *Cells* 136, pp. 337–351, doi:10.1016/j.cell.2008.11.051.

[3] A. Gentle, Y. Liu, J. E. Martin, G. L. Conti & N. A. McBrien (2003): *Collagen gene expression and the altered accumulation of scleral collagen during the development of high myopia*. *Journal of Biological Chemistry* 278(19), pp. 16587–16594, doi:10.1074/jbc.M300970200.

[4] R. Goebel, R. G. Sanfelice & A. R. Teel (2012): *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press.

[5] C. Guo, J. Jiang, J. Martin Elliott & L. Piacentini (2005): *Paradigmatic identification of MMP-2 and MT1-MMP activation systems in cardiac fibroblasts cultured as a monolayer*. *Journal of cellular biochemistry* 94(3), pp. 446–459, doi:10.1002/jcb.20272.

[6] J. Hu, D. Cui, X. Yang, S. Wang, S. Hu, C. Li & J. Zeng (2008): *Bone morphogenetic protein-2: a potential regulator in scleral remodeling*. *Molecular Vision* 14, pp. 2370–2380.

[7] J. Hu, K. R. Qin, C. Xiang & T. H. Lee (2010): *Modeling of Hysteresis in a Mammalian Gene Regulatory Network*. In: *9th Annual International Conference on Computational Systems Bioinformatics*, 9, Life

Sciences Society, pp. 50–55.

[8] M. D. Jacobs (2009): *Multiscale systems integration in the eye*. *WIREs Systems Biology and Medicine* 1, pp. 15–27, doi:10.1002/wsbm.29.

[9] F. T. Ji, Q. Li, Y. L. Zhu, L. Q. Jiang, X. T. Zhou, M. Z. Pan & J. Qu (2009): *Form Deprivation Myopia in C57BL/6 Mice*. *[Zhonghua yan ke za zhi] Chinese journal of ophthalmology* 45(11), p. 1020.

[10] B. P. Kramer & M. Fussenegger (2005): *Hysteresis in a synthetic mammalian gene network*. *Proceedings of the National Academy of Sciences (USA)* 102, pp. 9517–9522, doi:10.1073/pnas.0500345102.

[11] N. A. McBrien (2013): *Regulation of Scleral Metabolism in Myopia and the Role of Transforming Growth Factor-beta*. *Experimental eye research*, doi:10.1016/j.exer.2013.01.014.

[12] N. A. McBrien & A. Gentile (2003): *Role of the sclera in the development and pathological complications of myopia*. *Progress in Retinal and Eye Research* 22, pp. 307–338, doi:10.1016/S1350-9462(02)00063-0.

[13] N. A. McBrien, P. Lawlor & A. Gentle (2000): *Scleral Remodeling During the Development of and Recovery from Axial Myopia in the Tree Shrew*. *Investigative ophthalmology & visual science* 41(12), pp. 3713–3719.

[14] T. Mestl, E. Plahte & S. W. Omholt (1995): *A mathematical framework for describing and analysing gene regulatory networks*. *Journal of Theoretical Biology* 176, pp. 291–300, doi:10.1006/jtbi.1995.0199.

[15] S. Monea, K. Lehti, J. Keski-Oja & P. Mignatti (2002): *Plasmin activates pro-matrix metalloproteinase-2 with a membrane-type 1 matrix metalloproteinase-dependent mechanism*. *Journal of cellular physiology* 192(2), pp. 160–170, doi:10.1002/jcp.10126.

[16] E. Morgunova, A. Tuuttila, U. Bergmann & K. Tryggvason (2002): *Structural insight into the complex formation of latent matrix metalloproteinase 2 with tissue inhibitor of metalloproteinase 2*. *Proceedings of the National Academy of Sciences of the USA* 99(11), pp. 7414–7419, doi:10.1073/pnas.102185399.

[17] J. A. Rada, C. A. Perry, M. L. Slover & V. R. Achen (1999): *Gelatinase A and TIMP-2 expression in the fibrous sclera of myopic and recovering chick eyes*. *Investigative ophthalmology & visual science* 40(13), pp. 3091–3099.

[18] J. A. Rada, S. Shelton & T. T. Norton (2006): *The Sclera and Myopia*. *Experimental eye research* 82(2), pp. 185–200, doi:10.1016/j.exer.2005.08.009.

[19] N. Sakalihasan, P. Delvenne, B. V. Nusgens, R. Limet & C. M. Lapière (1996): *Activated forms of MMP2 and MMP9 in abdominal aortic aneurysms*. *Journal of vascular surgery* 24(1), pp. 127–133, doi:10.1016/S0741-5214(96)70153-2.

[20] R. G. Sanfelice, D. A. Copp & P. Nanez (2013): *A Toolbox for Simulation of Hybrid Systems in Matlab/Simulink: Hybrid Equations (HyEQ) Toolbox*. In: *Proceedings of Hybrid Systems: Computation and Control Conference*, pp. 101–106, doi:10.1145/2461328.2461346.

[21] L. Shelton & J. S. Rada (2007): *Effects of cyclic mechanical stretch on extracellular matrix synthesis by human sclera fibroblasts*. *Experimental Eye Research* 84, pp. 314–322, doi:10.1016/j.exer.2006.10.004.

[22] S. L. Shelton (July 2009): *Characterization of Mechanisms Regulating Scleral Extracellular Matrix Remodeling to Promote Myopia Development*. *PhD thesis, University of Oklahoma Health Science Center, Oklahoma City*.

[23] Q. Shu & R. G. Sanfelice (2013): *Dynamical Properties of a Two-gene Network with Hysteresis*. Submitted to Special Issue on Hybrid Systems and Biology, Elsevier Information and Computation.

[24] J. T. Siegwart & T. T. Norton (2005): *Selective regulation of MMP and TIMP mRNA levels in tree shrew sclera during minus lens compensation and recovery*. *Investigative ophthalmology & visual science* 46(10), pp. 3484–3492, doi:10.1167/iovs.05-0194.

[25] J. C. H. Tan, F. B. Kalapesi & M. T. Coroneo (2006): *Mechanosensitivity and the eye: cells coping with the pressure*. *British Journal of Ophthalmology* 90, pp. 383–388, doi:10.1136/bjo.2005.079905.

[26] W. Xiong, R. Knispel, J. Mactaggart & B. T. Baxter (2006): *Effects of tissue inhibitor of metalloproteinase 2 deficiency on aneurysm formation*. *Journal of vascular surgery* 44(5), pp. 1061–1066, doi:10.1016/j.jvs.2006.06.036.

[27] Y. Yang, X. Li, N. Yan, S. Cai & X. Liu (2009): *Myopia: A collagen disease? Medical hypotheses* 73(4), pp. 485–487, doi:10.1016/j.mehy.2009.06.020.