An Algorithm for Qualitative Simulation of Gene Regulatory Networks with Steep Sigmoidal Response Functions

Liliana Ironi[1], Luigi Panzeri[1], and Erik Plahte[2]

[1] IMATI - CNR, via Ferrata 1, 27100 Pavia, Italy
[2] CIGENE (Centre for Integrative Genetics) and Department of Mathematical Sciences and Technology, Norwegian University of Life Sciences, P.O. Box 5003, N-1432 As, Norway

Abstract. A specific class of ODEs has been shown to be adequate to describe the essential features of the complex dynamics of Gene-Regulatory Networks (GRN). But, the effective exploitation of such models to predict the dynamics of specific GRNs by classical numerical schemes is greatly hampered by the current lack of precise and quantitative information on regulation mechanisms and kinetic parameters. Due to the size and complexity of large GRNs, classical qualitative analysis could be very hard, or even impracticable, to be carried out by hand, and conventional qualitative simulation approaches rapidly lead to an exponential growth of the generated behavior tree that, besides all possible sound behaviors, may also contain spurious ones. This paper discusses the work-in-progress of a research effort aiming at the design and implementation of a computational framework for qualitative simulation of the dynamics of a class of ODE models of GRNs. The algorithm we propose results from a set of symbolic computation algorithms that carry out the integration of qualitative reasoning techniques with singular perturbation analysis methods. The former techniques allow us to cope with uncertain and incomplete knowledge whereas the latter ones lay the mathematical groundwork for a sound and complete algorithm capable to deal with regulation processes that occur at different time scales.

Key words: Gene regulatory network, qualitative simulation, singular perturbation analysis.

1 Introduction

A variety of modeling formalisms, ranging from directed graphs over Boolean networks to differential models, ordinary and partial differential equations, along with related simulation algorithms have been applied to study gene regulatory systems as demonstrated by a rich literature, and discussed in several monographs and survey papers. The mathematical aspects of such approaches, and the evaluation of their relative strengths and weaknesses are discussed in [1].

A growing number of theoretical as well as experimental papers show that gene regulation is threshold-dependent, i.e. only effective above or below a certain threshold. ODE models with switch-like interaction terms allow us to provide detailed descriptions of gene regulatory mechanisms at the molecular level [2], and, in theory, they could be

used to make numerical predictions of the network behavior and its response to environmental stimuli. But, in practice, even when the system at hand is very well studied, their exploitation meets several obstacles: (i) models are often large, complex and above all, nonlinear, and traditional pen and paper analysis could be inadequate or too time-consuming; (ii) numerical simulations are seriously hampered by the current lack of precise and quantitative information on the biochemical reaction mechanisms underlying regulatory interactions, kinetic parameters and threshold concentrations.

The qualitative analysis and simulation of the dynamics of GRNs is a rather appropriate solution as, in the current state of knowledge, the key issue is to understand how specific activity patterns derive from given network structures, and what different types of dynamical behaviors are possible. However, due to the complexity of GRN models, to carry out analytically a complete qualitative study of their dynamics might be very hard, or even impracticable. Nor is the recourse to conventional qualitative simulation approaches [3], developed within the Artificial Intelligence research framework to cope with the need to represent and predict the dynamics of systems characterized by incomplete knowledge, an appropriate solution. Such approaches can deal with generic classes of dynamical systems, and generate, starting from an initial system state, the whole range of system dynamics. Each qualitative behavior is generated by applying transition rules that are grounded on mathematical tools actually too simple to compensate for the lack of complete knowledge. This results in a number of drawbacks, e.g. their inability to upscalability, the exponential growth of the generated behaviors, and the generation of spurious behaviors, that reveal to be particularly serious in predicting nonlinear dynamics of regulatory networks even in the case of networks with a small number of interacting genes. Thus, the need for the development of *ad hoc* computational frameworks for qualitative analysis and simulation of GRN models .

A first effort in this direction is given by a qualitative simulator, called GNA [4]. It is based on the integration of Qualitative Reasoning (QR) concepts [3] and control theory methods to cope with both incomplete knowledge and threshold-dependent regulation mechanisms. GNA assumes that threshold-regulated response functions are step functions, discontinuous in the threshold hyperplanes. On the one hand, such an assumption considerably simplifies the analysis as the model results in piecewise-linear equations. On the other hand, it raises the problem to find a proper continuous solution across the threshold hyperplanes, or, in other word, to seek for generalized solutions of ODEs with discontinuous right-side terms. Among the possible definitions of generalized solutions, GNA adopts the Filippov one [5], that is quite popular and convenient in a control context, but may fail when applied to approximate the limit solutions of a continuous model. This together with a further approximation introduced in the GNA algorithm for computational purposes might not always guarantee its soundness and completeness [6].

These problems are avoided if the response functions are sigmoidal and vary continuously from zero to one with a steep rise around the threshold. The ensuing dynamics is both linear and nonlinear with different time scales, but has been analysed using singular perturbation theory [7]. Here we present a qualitative simulation algorithm for models with steep sigmoid response functions based on this work. Despite an incomplete knowledge of parameter values, we construct all possible trajectories and their associated parameter space domains, starting from an initial state and parameter space

domain, by using QR key concepts suitably revised, and by iteratively exploiting symbolic computation procedures.

2 A Modeling Framework for the Study of GRN Dynamics

Recent experimental findings as well as theoretical justifications (see [8] for references) seem to support the following generic and phenomenological dynamic model for a GRN:

$$\dot{x_i} = f_i(\mathbf{Z}) - \gamma_i x_i \ , \tag{1}$$

where the dot denotes time derivative, x_i is the concentration of gene product number i, $i = 1, \ldots, n$, γ_i is the relative decay rate of x_i, $\mathbf{Z}$ is a vector with Z_{jk} as components, and $Z_{jk} = S(x_j, \theta_{jk}, q)$ is a sigmoid or binary (i.e. Heaviside or step) response function with threshold θ_{jk} and steepness parameter q. The thresholds associated with each x_i are ordered according to $\theta_{ij} < \theta_{ik}$ if $j < k$. Finally, $\mathbf{x}(t, \mathbf{x^0}, q)$ is the solution satisfying the initial condition $\mathbf{x}(0, \mathbf{x}^0, q) = \mathbf{x}^0$.

The differentiable functions f_i are regulatory production functions, frequently composed by algebraic equivalents of Boolean functions [9]. Eq. (1) is assumed to catch the essential features of a wide range of regulatory systems, where the regulatory control may be at the level of transcription, mRNA stability, translation, or post-translation. The state variables may be concentrations of proteins, hormones, mRNA, and intracellular ions [10]. The framework has been applied to model real world networks: the initiation of sporulation in *B. subtilis* [11], and the response to nutritional stress and carbon starvation in *E. coli* [12, 13], using Heaviside response functions. We assume the sigmoid functions are very steep Hill functions, viz. $S(x, \theta, q) = x^{1/q}/(x^{1/q} + \theta^{1/q})$, where $0 < q \ll 1$. Under a number of reasonable assumptions a solution $\mathbf{x}(t, \mathbf{x^0}, q)$ of Eq. (1) starting in an arbitrary initial point $\mathbf{x}^0$ at $t = 0$ can be uniformly approximated by the zero order solution $\mathbf{x}(t, \mathbf{x^0}, 0)$. The derivation is based on singular perturbation theory.

Eq. (1) could be applied to GRNs of any size and complexity, and the generic analysis developed in [7, 14] is fully applicable to networks of any size. However, as the network becomes large, there is a combinatorial explosion of phase-space domains and parameter combinations that need to be investigated, and a computerized, algorithmic approach becomes a necessity. That is the first motivation for the present work. Our second motivation is that sigmoidal response functions in many cases are more realistic than step functions, and could be preferred for modeling real systems.

Regular and Singular Stationary Points. The threshold hyperplanes $x_i = \theta_{ij}$ divide phase-space into regular and switching domains. A *regular domain* D_R (also called a *box*) is an open rectangular domain between threshold planes in which the values of all sigmoids are close to 0 or 1. In a box we put $Z_{ij} = B_{ij}$. In a *switching domain* D_S the x_i are divided into two disjoint sets: the switching variables x_s and the regular variables x_r, $s \in \mathcal{S}$, $r \in \mathcal{R}$, where $\mathcal{S} \cup \mathcal{R} = \mathcal{N} = \{1, 2, \ldots, n\}$. A x_s is very close to one of its thresholds, while a x_r lies in the open domain between two adjacent thresholds. Thus, a switching domain is a narrow boundary layer surrounding a section of a threshold plane or an intersection of threshold planes. The union of all the regular domains is denoted Δ_R, while Δ_S is the union of all the switching domains, and $\Delta = \Delta_R \cup \Delta_S$.

A stationary point $\mathbf{P}(q)$ is called a *regular stationary point* (RSP) if it is located in a box, and a *singular stationary point* (SSP) if it is located in a switching domain. The zero order approximation $\mathbf{P}^0$ of a RSP $\mathbf{P}(q)$ is a solution of $f_i(\mathbf{B}) - \gamma_i x_i = 0$, $i \in \mathcal{N}$, where $\mathbf{B}$ represent the Boolean values of the Z-variables. This is trivial to solve for each $\mathbf{B}$. A RSP is always asymptotically stable.

If a SSP exists in a D_S, it is in lowest order found as the solution $\mathbf{P}^* = (x_r^*, \theta_s)$ of

$$\begin{aligned} f_r(B_r, Z_s) - \gamma_r x_r = 0, \quad r \in \mathcal{R}\ , \\ f_s(B_r, Z_s) - \gamma_s \theta_s = 0, \quad s \in \mathcal{S}\ . \end{aligned} \tag{2}$$

In the following we make the apparently realistic and simplifying

Assumption $\mathcal{A}$. Every x_i only regulates one gene at each of its thresholds.

Because each Z_{ij} then only occurs in a single rate function, the second of Eqs. (2), which would otherwise represent a set of polynomial equations that could be very hard to solve, is reduced to a set of linear equations. Then, there is at most a single stationary point $\mathbf{P}^*$ in each D_S. A necessary condition for a solution of the second equation is that there is precisely one Z-term in each equation and that these terms produce a non-singular Jacobian matrix $\mathbf{J}_s = \partial f_s / \partial Z_s$. These Jacobian elements form a loop $\mathcal{L}$ with loop product L. A suitable renumbering leads to a block-diagonal $\mathbf{J}_s$, where each block is a permutation matrix associated with a sub-loop $\mathcal{L}_j$ of $\mathcal{L}$. Then the characteristic equation $|\mathbf{J}_s - \lambda I| = 0$ is

$$\prod_{j=1}^{m} \Big((-\lambda)^{l(j)} + L_j \Big) = 0 \tag{3}$$

where m is the number of sub-loops of $\mathbf{J}_s$, $l(j)$ is the length of $\mathcal{L}_j$ and L_j is the loop product of $\mathcal{L}_j$. It follows that $\mathbf{P}^*$ has no eigenvalues with positive real part and is stable iff (i) $l(j) = 1$ or $l(j) = 2$ for all j, (ii) $L_j < 0$ for $j = 1$, (iii) $L_j > 0$ for $j = 2$, the latter case occuring if there is a negative loop among the two variables, giving a pair of imaginary eigenvalues. All other cases will give an eigenvalue with a positive real part.

Thus, when the SSP is stable, we only encounter all eigenvalues with negative real part in the very special case when $\mathbf{J}_s$ is diagonal with only negative diagonal elements. If some $l(j) = 2$, we get a pair of purely imaginary eigenvalues. This causes a problem, because the standard proof of the validity of the singular perturbation approximation requires eigenvalues with negative real parts. This remains to be investigated. However, numerical simulations indicate that singular perturbation should work also in this case.

Dynamical Behavior. In a box all x_i are regular, and in the step function limit the solutions approach the solutions of

$$\dot{x_r} = f_r(\mathbf{B}) - \gamma_r x_r\ . \tag{4}$$

In a switching domain D_S where $x_s = \theta_{s,s_j}$, the rapid motion of the switching variables is described by

$$Z'_{s,s_j} = D_{s,s_j}[f_s(B_r, Z_{s,s_j}) - \gamma_s \theta_{s,s_j}]\ , \tag{5}$$

where $D_{s,s_j} = Z_{s,s_j}(1 - Z_{s,s_j})/\theta_{s,s_j}$ stems from the derivative of the Hill function, and the prime denotes differentiation with respect to $\tau = t/q$.

If Eq. (5) has no stationary and asymptotically stable solution $Z^*_{s,s_j} \in (0,1)^\sigma$, where $\sigma = |\mathcal{S}|$, the system just passes through D_S and leaves $(0,1)^\sigma$ in an *exit point* at the boundary, with no change in the regular variables. Otherwise, the Z-variables come to rest in the exit point Z^*_{s,s_j}, and the slow motion of the regular variables x_r in this D_S is described by the linear and independent equations

$$\begin{aligned} \dot{x_r} &= f_r(B_r, Z^*_{s,s_j}) - \gamma_r x_r \ , \\ Z_{s,s_j} &= Z^*_{s,s_j} \ , \end{aligned} \tag{6}$$

until a stable state is reached or the solution leaves D_S and passes into an adjacent switching domain. In the singular perturbation language, Eq. (5) is called the *boundary layer equation* and Eq. (6) the *reduced equation*.

The phase space of Eq. (5) is confined to the so-called Z-cube $\mathcal{Z}(D_S) = [0,1]^\sigma$ associated with D_S. In the limit $q \to 0$, the interior of the Z-cube describes the motion of the singular Z-variables, while the faces describe the switching variables in the adjacent switching domains, and the vertices represent the adjacent boxes [14]. Each Z-cube has a set of *entrance* and *exit* points where trajectories can enter, respectively exit from, the interior of the Z-cube. An entrance point is also an exit point of an adjacent domain, so we will only need exit points. An exit point can be located on the boundary of the cube or be an internal point.

According to singular perturbation theory [15], the solutions of Eqs. (5,6) taken together approximate the exact solution in D_S for q close to zero, i.e. for steep sigmoids [7]. One just has to express x_s by the solution $Z_{s,s_j}(\tau)$ and replace τ by t/q. For each switching domain D_S there is essentially only one problem: to determine the exit point Z^*_s where the trajectory leaves D_S or the system comes to rest. The details of the motion of the Z-variables are not important as they only occur in a narrow part of phase space and in a negligible time span. Using this approach, we can construct a solution starting in an arbitrary initial point $\mathbf{x}^0$ at $t = 0$ through any finite sequence of regular and switching domains by supplying Eqs. (4-6), solved for each domain encountered, with initial conditions which ensure a continuous trajectory. This zero order solution $\mathbf{x}(t, \mathbf{x^0}, 0)$ is then a uniform approximation to $\mathbf{x}(t, \mathbf{x^0}, q)$ for $0 < q \ll 1$. This is the theoretical basis for the algorithm to be described below.

3 Qualitative Simulation of GRN Models

Our work aims at the development of a qualitative simulation algorithm based on sophisticated mathematical tools, and specifically tailored to capture network dynamical properties that are invariant for ranges of values of kinetic parameters. To this end, in the following we revise and *ad hoc* tailor the key concepts underlying qualitative simulation algorithms to our specific class of models.

Qualitative Value.The partition of the whole system domain, induced by the ordered sets, Θ_i, of the n_i symbolic threshold values θ_{ij} associated with each x_i, identifies

qualitatively distinct *n*-dimensional hyper-rectangles D that define *the system qualitative values*. Let us observe that, to characterize switching domains, instead of the sharp value θ_{ij}, we consider a range of values around it, whose width, $\delta > 0$, is a monotonic function of the steepness parameter q with $\delta(q) \to 0$ for $q \to 0$. Let us denote by $\underline{\theta}_{ij}$ and $\overline{\theta}_{ij}$ the values $\theta_{ij} - \delta/2$ and $\theta_{ij} + \delta/2$, respectively. Then, each D results from the product of intervals, either all open, $(\overline{\theta}_{ij}, \underline{\theta}_{i(j+1)})$, or at least one closed, $[\underline{\theta}_{i(j+1)}, \overline{\theta}_{i(j+1)}]$.

Qualitative State. Let $A(D)$ be the set of domains adjacent to $D \in \Delta$. The *qualitative state* of D, $QS(D) = \{D_k \mid D_k \in A(D),\ D \to D_k\}$, is defined by all of its adjacent domains D_k towards which a transition from it is possible. Each transition from D identifies a domain next traversed by a system trajectory. More precisely, if we number by i the domain D traversed at time t_i, each of its successors $D_k \in QS(D)$ will be traversed by different trajectories at time t_{i+1}.

State Transitions. The possible transitions from D are determined by different strategies according to whether $D \in \Delta_R$ or $D \in \Delta_S$. In the former case, like in traditional QR methods and in GNA [4], transitions are determined by the signs of $\dot{x}_i$. As $\dot{x}_i$ are defined by linear expressions, such signs are easily determined by exploiting the inequalities that define the parameter space domain, and constrain the RSPs to belong to specific domains. In the case, $D \in \Delta_S$, a sign-based strategy is not practicable as the expressions for $\dot{x}_i$ are nonlinear. A convenient way to proceed is given by singular perturbation analysis: transitions from D towards adjacent D_k are determined by the locations of its exit points that can be either on (i) the boundary of D or in (ii) its interior. Except in the case (ii), the number of exit points may be greater than one. Then, in general, the successors of D are not uniquely determined. But, through symbolic computation procedures, it is possible to calculate the set of inequalities, I^i_j, on parameters that hold when a transition from D_i to D_j occurs. Then, each path from D_i to D_j is clearly identified by the 3-tuple $\langle D_i, D_j, I^i_j \rangle$.

Qualitative Behavior. A finite sequence of paths, where each path is clearly both linked and consistent with its predecessor and successor, defines a *qualitative behavior*:
$QB = \langle D_0, I_0 \rangle, \langle D_0, D_1, I^0_1 \rangle, \ldots, \langle D_k, D_i, I^k_i \rangle, \ldots, \langle D_F, I_F \rangle$, where D_0 is the initial domain, and D_F either contains a stable fixed point or identifies a cycle, i.e it is an already visited domain. I_0 is the initial set of inequalities that defines the parameter space domain, and I_F the set of inequalities on parameter values associated with D_F.

3.1 The Simulation Algorithm

Given as input, (i) n symbolic state equations of the form (1); (ii) n ordered sets $\Theta_i = \{\theta_{ij}\}$; (iii) an initial domain $D_0 \in \Delta$; (iv) a set of symbolic inequalities I_0 on parameter values defining a parameter space domain PSD_0, the simulation algorithm generates all possible state transitions, and represents them by a directed tree rooted in D_0, $\mathrm{BT}(D_0)$. In such a tree the vertices correspond to D_i, and the arcs, labeled by the inequalities I^i_j, to the transitions from D_i to D_j. Each branch in $\mathrm{BT}(D_0)$ defines a qualitative trajectory from D_0, that occurs when the values of parameters satisfy its related inequalities. Such a trajectory is characterized by the traverse of specific domains, and abstracts all those numeric solutions of the ODE, obtained with different values of

either the initial condition $\mathbf{x}^0 \in D_0$ or parameters, that cross the same domains. The main steps of the algorithm, sketched in Fig. 1, are summarized below:

1. *Partition* the phase space into regular and switching domains.
2. *Calculate* the qualitative state $QS(D_i)$ of the current domain D_i.
3. *Determine constraints* I_k^i on parameters for each path $e_{ik} = D_i \rightarrow D_k$, where $D_k \in QS(D_i)$.
4. *Append* $\langle D_i, D_k, I_k^i \rangle$ to BT(D_0) if I_k^i are consistent with the initial constraints I_0, and *mark* D_i as visited domain.
5. *Repeat* from step 2 for each D_k.

In the following we detail step 2, that is the core of the algorithm.

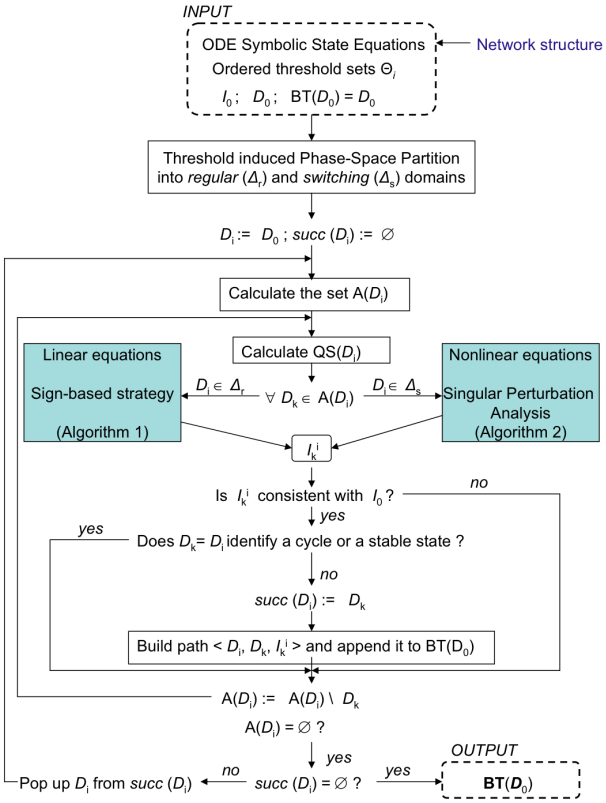


Fig. 1. Main steps of the simulation algorithm.

3.2 Calculation of the Qualitative State

The calculation of the qualitative state requires two separate algorithms to implement the different strategies adopted according to whether $D_i \in \Delta_R$ or $D_i \in \Delta_S$. Both

algorithms calculate the conditions on parameters I_k^i that are consistent with $I_0, \forall D_k \in QS(D_i)$. Let us define I_k^i *consistent* with I_0 when it defines a not empty parameter space domain PSD_k^i such that $PSD_k^i \subseteq PSD_0$. Furthermore, we define the relative position of D_1 with respect to D_2, indicated by $V(D_1, D_2) = \{v_j\}_{j=1}^n$ where $v_j \in \{-1, 0, 1\}$, by the comparison of the intervals defining D_1 and D_2.

Transition from a Regular Domain (Algorithm 1). The algorithm that constructs the possible paths from regular domains is, in principle, similar to that one proposed by GNA, but it is more informative as it calculates the I_k^is. From now on, for the sake of simplicity, we indicate the two consecutive thresholds $\theta_{ji}, \theta_{j(i+1)}$ by θ_j, θ'_j. Then, let D_i be defined by $D_i = \prod_{j=1}^n (\overline{\theta}_j, \underline{\theta}'_j)$. In outline, the algorithm performs the following steps:

1- *Calculate $A(D_i)$ and state equations in D_i.* The algorithm calculates the set $A(D_i)$, the symbolic state equations in D_i, and the focal point $\mathbf{x}^*$ towards which all trajectories head when $t \to \infty$.

2- *Calculate I_k^i and possible transitions.* $\forall D_k \in A(D_i)$, the algorithm calculates the set of inequalities on parameters I_k^i that need to be fulfilled to have a transition from D_i to D_k. As all the equations are linear in D_i, such inequalities are calculated by imposing that the signs of state variable rates match the relative position of D_k with D_i. Let $V(D_k, D_i) = \{v_j\}_{j=1}^n$ be the relative position of D_k with respect to D_i. I_k^i is given, $\forall j \in \{1, \ldots, n\}$, by either the inequality $x_j^* > \overline{\theta}'_j$ if $v_j = 1$ or $x_j^* < \underline{\theta}_j$ if $v_j = -1$. Thus, if the calculated inequality set defines a not empty parameter space domain $PSD_k^i \subseteq PSD_0$, then a transition towards D_k is possible and the qualitative state $QS(D_i)$ is updated accordingly.

3- *Check the existence of a* RSP *in D_i.* A stable point RSP exists in D_i, i.e. $D_i \in QS(D_i)$, if $\widetilde{PSD} \subseteq PSD_0$ and $\widetilde{PSD} \neq \emptyset$, where $\widetilde{PSD}$ is a parameter space domain defined by the set of inequalities $\overline{\theta}_j < x_j^* < \underline{\theta}'_j, \forall j \in \{1, \ldots, n\}$.

Algorithm 1 Calculate $QS(D_i)$ for Regular Domain

1: Set $I_k^i \leftarrow I_0$
2: **for all** $D_k \in A(D_i)$ **do**
3: *Calculate* $V(D_k, D_i) = \{v_j\}_{j=1}^n$
4: **for** $j = 1$ to n **do**
5: *Calculate* the symbolic state equation in D_i and its stationary solution x_j^*;
6: *Update*:

$$I_k^i \leftarrow I_k^i \wedge \begin{cases} (x_j^* > \overline{\theta}'_j) & \text{if } v_j = 1 \\ (x_j^* < \underline{\theta}_j) & \text{if } v_j = -1 \end{cases}$$

7: **if** $PSD_k^i \neq \emptyset$ **then**
8: *Append* D_k to $QS(D_i)$ and *label* the path $D_i \to D_k$ by I_k^i.
9: *Append* D_i to $QS(D_i)$ if $\widetilde{PSD} \subseteq PSD_0$ and $\widetilde{PSD} \neq \emptyset$, where $\widetilde{PSD}$ is a parameter space domain defined by the set of inequalities $\{(\overline{\theta}_j < x_j^* < \underline{\theta}'_j)\ \forall j \in \{1, \ldots, n\}$

Transition from a Switching Domain (Algorithm 2). The nonlinear dynamics in a switching domain D_i is characterized by fast and slow motions, respectively associated with x_s and x_r that are independently calculated.

The study of the fast dynamics is performed in $\mathcal{Z}(D_i)$ in the scaled time τ, and aims at localizing the set of exit points in $\mathcal{Z}(D_i)$ rather than at detailing the dynamics within it. Such points clearly identify the next domains the trajectories are moving towards from D_i along the x_s directions. To this end, the algorithm proceeds as follows:
1- *Calculate the boundary layer equations in* D_i. The algorithm symbolically calculates the boundary layer equations in the Z variables, and defines the mapping $\Sigma_{D_i} : \mathcal{D}_i \rightarrow \mathcal{Z}(D_i)$, where $\mathcal{D}_i = D_i \cup A(D_i)$, that states a correspondence between D_i and its adjacent domains D_k with the interior and the elements on the boundary of $\mathcal{Z}(D_i)$. Let $\mathcal{F}$ be the set of both the faces and the interior of $\mathcal{Z}(D_i)$: its generic element $F = \Sigma_{D_i}(D)$, $D \in \Delta_s$ is either a face of $\mathcal{Z}(D_i)$ when $D \in A(D_i)$ or its interior when $D = D_i$.
2- *Search for stationary points.* Let us denote by EP the set of stationary points, initially made up of the vertices of $\mathcal{Z}(D_i)$. The set of the candidate exit points EP is updated by the possible stationary point on each element of $\mathcal{F}$. To this end, the algorithm symbolically calculates, $\forall F \in \mathcal{F}$, the Jacobian matrix $\mathbf{J}_F$. As the presence of a non-zero loop is a necessary condition for the existence of a stationary point, the algorithm first searches for a non-zero loop involving all variables in $\mathbf{J}_F$: in case, it symbolically calculates the stationary point on F, and updates accordingly the set of candidate exit points EP.
3- *Calculate* I_k^i *and possible transitions by checking stability of stationary points.* The inequality set I_k^i is calculated for each candidate exit point $\tilde{\mathbf{Z}}^{\mathbf{k}} = \{\tilde{Z}_s^k\} \in EP$ by requiring that each point fulfills stability conditions. In addition, for those $\tilde{\mathbf{Z}}^{\mathbf{k}}$ located on elements of $\mathcal{F}$, I_k^i is further constrained by the inequalities on parameters that impose $0 < \tilde{Z}_s^k < 1$ for each $\tilde{Z}_s^k \notin \{0, 1\}$. The stable points located on $\mathcal{Z}(D_i)$ clearly identify the set of all possible exit domains, i.e. those domains towards which a transition from D_i is possible. Such domains are easily calculated by applying the map Σ^{-1} to each element of $\mathcal{Z}(D_i)$ that contains an exit point.

The slow dynamics of regular variables x_r is studied in the normal time in the usual frame of reference, and it is reconstructed from the reduced system through the same symbolic procedure given for regular domains.

3.3 Remarks about Symbolic Computations

Most of the calculations are performed symbolically. In additon to plain symbolic algebraic manipulation like arithmetic and derivative operations, the algorithms are required to tackle more complex tasks, such as: (i) refine an inequality set with an another one; (ii) check the consistency of two sets of inequalities I_1 and I_2; (iii) solve systems of equations; (iv) find loops in the Jacobian matrix.

Assumption $\mathcal{A}$ leads to major simplifications. As for (iii), Eq. (1) are generally multilinear in $\mathbf{Z}$, but now assume a linear form in each D_S, and can be straightforwardly solved and analyzed for stability. Also, the solution of problems (i) and (ii) are simplified as the inequalities are always linear. Thanks to algorithms proposed both by the literature and common symbolic computation package, such as Mathematica [16],

Algorithm 2 Calculate $QS(D_i)$ for a Switching Domain

Initialize $EP \leftarrow \emptyset$
Calculate symbolically the boundary layer system in $\mathcal{Z}(D_i)$
Update EP by adding all the vertices of $\mathcal{Z}(D_i)$
Calculate symbolically the Jacobian matrix $\mathbf{J}$
for all $F \in \mathcal{F}$ **do**
  Calculate $\mathbf{J_F}$
  if there is a complete loop in $\mathbf{J_F}$ **then**
    Calculate the stationary point in F by solving symbolically $\mathbf{Z'_s} = 0$
    Update EP by adding the point calculated at the previous step
for all $\tilde{\mathbf{Z}}^{\mathbf{k}} = \{\tilde{Z}_s^k\} \in EP$ **do**
  Initialize $I_k^i \leftarrow I_0$
  for $s = 1$ to $\sigma(D_i)$ **do**
    if $\tilde{\mathbf{Z}}^{\mathbf{k}} \in F, \quad F \in \mathcal{F}$ **then** *Build* $I_k^i \leftarrow I_k^i \wedge (0 < \tilde{Z}_s^k < 1)$
  for $j = 1$ to m **do**
    if $lenght(l_j) > 2$ **then** *Remove* $\tilde{\mathbf{Z}}^{\mathbf{k}}$ from EP
    if $length(l_j) = 1$ **then**
      if $L_j < 0$ **then** *Build* $I_k^i \leftarrow I_k^i \wedge (L_j < 0)$ **else** *Remove* $\tilde{\mathbf{Z}}^{\mathbf{k}}$ from EP
    else
      if $length(l_j) = 2$ **then**
        if $L_j > 0$ **then** *Build* $I_k^i \leftarrow I_k^i \wedge (L_j > 0)$ **else** *Remove* $\tilde{\mathbf{Z}}^{\mathbf{k}}$ from EP
  for all $l \in \mathcal{L}_F = \{l : l \in \{1, \ldots, \sigma(D_i)\}, \quad \tilde{Z}_l^k \in \{0, 1\}\}$ **do**
    Build

$$I_k^i \leftarrow I_k^i \wedge \begin{cases} (Z'_l(\tilde{\mathbf{Z}}^{\mathbf{k}}) > 0) & \text{if } \tilde{Z}_l^k = 1 \\ (Z'_l(\tilde{\mathbf{Z}}^{\mathbf{k}}) < 0) & \text{if } \tilde{Z}_l^k = 0 \end{cases}$$

  Calculate the Exit Domain Set: $ED = \{D_k : D_k = \Sigma_{D_i}^{-1}(\tilde{\mathbf{Z}}^{\mathbf{k}}), \tilde{\mathbf{Z}}^{\mathbf{k}} \in EP\}$
  Calculate symbolically in $\tilde{\mathbf{Z}}^{\mathbf{k}}$ the reduced system and its stationary solution $\mathbf{x}^*$
  for all $D_k \in ED$ **do**
    Calculate $V(D_k, D_i) = \{v_j\}_{j=1}^n$
    for $r = \sigma(D_i) + 1$ to n **do**
      Build

$$I_k^i \leftarrow I_k^i \wedge \begin{cases} (x_r^* > \theta'_r) & \text{if } v_r = 1 \\ (x_r^* < \theta_r) & \text{if } v_r = -1 \end{cases}$$

    if $PSD_k^i \neq \emptyset$ **then** *Append* D_k to $QS(D_i)$ and *label* the path $D_i \rightarrow D_k$ with I_k^i
Append D_i to $QS(D_i)$ if $\exists EP \in int(\mathcal{Z}(D_i))$ and $\widetilde{PSD} \subseteq PSD_0$ and $\widetilde{PSD} \neq \emptyset$ where $\widetilde{PSD}$ is defined by the set of inequalities $\overline{\theta}_j < x_j^* < \underline{\theta}'_j \; \forall j \in \{\sigma(D_i) + 1, \ldots, n\}$

the tasks (i)-(iii) are simplified and feasible. As for the task (iv), it is performed by using cycle–detection algorithms and tools of matrix graph theory [17].

In a more general modeling framework where Assumption $\mathcal{A}$ is removed, it could be really very hard to solve symbolically both the inequalities and equations as they might result in polynomials with very high order even for low dimensional systems.

4 An Example of the Algorithm at Work

To illustrate the algorithm at work, let us consider as an example the ODE system:

$$\begin{aligned} \dot{x_1} &= \kappa_1(1 - Z_{11})(1 - Z_{22}) + \kappa_2(1 - Z_{21}) - \gamma_1 x_1 \ , \\ \dot{x_2} &= \kappa_3(1 - Z_{12}) - \gamma_2 x_2 \ , \end{aligned} \tag{7}$$

where the Z_{jk} are expressed by Hill functions, parameters are all strictly positive and the quantity spaces $\Theta_1 = \{0 < \theta_{11} < \theta_{12} < \overline{x}_1\}$, $\Theta_2 = \{0 < \theta_{21} < \theta_{22} < \overline{x}_2\}$ partition the phase space into domains as showed in Fig. 3(a).

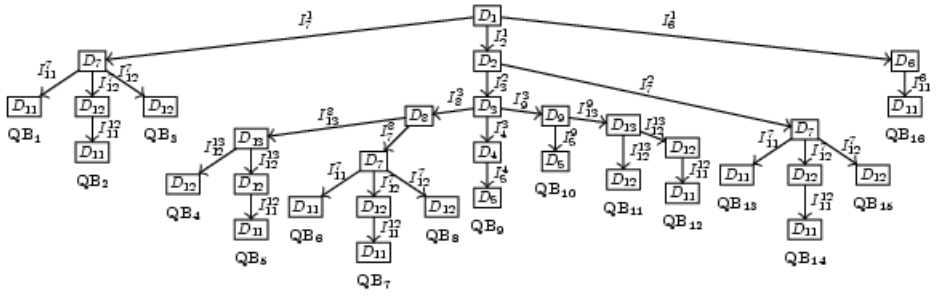


Fig. 2. Behavior tree rooted in D_1.

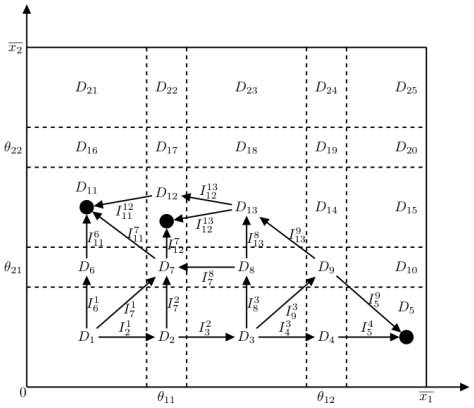
The simulation starts from D_1 with I_0 defined as follows:

$$I_0 : (\frac{\kappa_1 + \kappa_2}{\gamma_1} > \overline{\theta}_{11}) \quad \wedge \quad (\overline{\theta}_{21} < \frac{\kappa_3}{\gamma_2} < \underline{\theta}_{22}) \ . \tag{8}$$

The algorithm builds the behavior tree showed in Fig. 2, and calculates the inequalities on parameters, listed in Fig. 3(b), that are associated with each path in BT(D_1). Three reachable stable states, located in D_{11}, D_{12} and D_5, are identified by the final leaf of each branch in BT. As $D_{12} \in \Delta_s$, one of them is a SSP whereas the others are RSPs. These stable states are reached by different predicted qualitative behaviors, each of them occurring under specific constraints on parameters. For example, the trajectory QB_{16} starting from D_1, crossing D_6, and reaching a RSP in D_{11} is allowed when the inequalities I_6^1, I_{11}^6 and $I_{11} = (0 < \kappa_1/\gamma_1 < \underline{\theta}_{11}) \wedge (\overline{\theta}_{21} < \kappa_3/\gamma_2 < \underline{\theta}_{22})$ hold.

Let us observe, that at present, as we have not yet tackled the problem of identifying the admissible connections between entrance points and exit points, the algorithm may generate spurious behaviors, that is trajectories that can never occur for any set of numerical values of parameters. The behavior QB_2, e.g., is spurious as I_{12}^7 is not consistent with I_{11}^{12}. Similarly, QB_7 and QB_{14} are spurious. However, by checking the

consistency of all inequalities that belong to a branch in a BT, we can identify and filter out possible spurious behaviors. As $n = 2$, a representation in the phase plane of the trajectories described by the possibly filtered tree is also given (Fig. 3(a)).



(a)

I_3^2	$I_0 \wedge (\frac{\kappa_2}{\gamma_1} > \theta_{11})$
I_7^2	$I_0 \wedge (\frac{\kappa_2}{\gamma_1} < \theta_{11})$
I_5^4, I_5^9	$I_0 \wedge (\frac{\kappa_2}{\gamma_1} > \theta_{12})$
I_9^3, I_4^3	$I_0 \wedge (\frac{\kappa_2}{\gamma_1} > \overline{\theta}_{12})$
I_5	$I_0 \wedge (\overline{\theta}_{12} < \frac{\kappa_2}{\gamma_1} < \overline{x}_1)$
I_{11}^7, I_{11}^{12}	$I_0 \wedge (\frac{\kappa_1}{\gamma_1} < \theta_{11})$
I_{12}^7, I_{12}	$I_0 \wedge (\frac{\kappa_1}{\gamma_1} > \theta_{11})$
I_{11}	$I_0 \wedge (\frac{\kappa_1}{\gamma_1} < \underline{\theta}_{11})$

(b)

Fig. 3. (a) Phase space representation of trajectories described by BT after filtering; • denotes a stable state. (b) Inequalities calculated by the algorithm. I_2^1, I_6^1, I_7^1, I_8^3, I_{11}^6, I_7^8, I_{13}^8, I_{12}^{13}, I_{13}^9 are equal to I_0.

The simulation outcomes are numerically confirmed, and in Fig. 4 we report some of the numerical simulations performed under different conditions. In the following, we give a sketch of the algorithms at work when it calculates $QS(D_1)$, $QS(D_7)$.

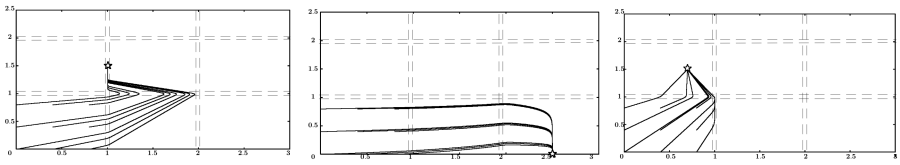
Calculation of $QS(D_1)$. First, the algorithm calculates the set $A(D_1) = \{D_6, D_7, D_2\}$ and the relative positions $V(D_6, D_1) = (0, 1)$, $V(D_7, D_1) = (1, 1)$, $V(D_2, D_1) = (1, 0)$. In D_1 the model (7) reduces to the linear ODEs:

$$\begin{aligned} \dot{x_1} &= \mu_1 - \gamma_1 x_1 \ , \\ \dot{x_2} &= \mu_2 - \gamma_2 x_2 \ , \end{aligned} \tag{9}$$

where $\mu_1 = \kappa_1 + \kappa_2$ and $\mu_2 = \kappa_3$. Transitions from D_1 are possible under the following conditions on parameters:

$$\begin{aligned} &\hat{I}_2^1 : \dot{x_1} > 0 \Rightarrow (\tfrac{\kappa_1+\kappa_2}{\gamma_1} > \overline{\theta}_{11}) \ \text{ to go to } D_2 \ , \\ &\hat{I}_6^1 : \dot{x_2} > 0 \Rightarrow (\tfrac{\kappa_3}{\gamma_2} > \overline{\theta}_{21}) \qquad \text{to go to } D_6 \ , \\ &\hat{I}_7^1 : \dot{x_1} > 0, \dot{x_2} > 0 \Rightarrow \hat{I}_2^1 \wedge \hat{I}_6^1 \ \text{ to go to } D_7 \ . \end{aligned} \tag{10}$$

As the parameter space domains defined by $I_2^1 : I_0 \wedge \hat{I}_2^1$, $I_6^1 : I_0 \wedge \hat{I}_6^1$ and $I_7^1 : I_0 \wedge \hat{I}_7^1$ belong to PSD_0, $QS(D_1) = \{D_2, D_6, D_7\}$, and the simulation spawns through three different paths. Let us follow the path related to condition I_7^1 and calculate $QS(D_7)$.



(a) (b) (c)

Fig. 4. Phase space plots of the numerical simulations performed with different parameter sets and initial conditions taken on an uniform grid of points in D_1. Common parameter values are: $\theta_{11} = \theta_{21} = 1$, $\theta_{12} = \theta_{22} = 2$, $q = 0.01$, $\kappa_3 = 1.5$, $\gamma_2 = 1$. Other parameters are: (a) $\kappa_1 = 2.5, \kappa_2 = 2.5, \gamma_1 = 1$; (b) $\kappa_1 = 25, \kappa_2 = 2.5, \gamma_1 = 10$; (c) $\kappa_1 = 0.7, \kappa_2 = 0.7, \gamma_1 = 1$. QB_4 and QB_{11} abstract the trajectories in (a), QB_9 in (b), and QB_1 and QB_{16} in (c).

Calculation of $QS(D_7)$. In the switching domain D_7 the Boundary Layer System is:

$$\begin{aligned} Z'_{11} &= \frac{Z_{11}(1-Z_{11})}{\theta_{11}}(\kappa_1(1-Z_{11}) + \kappa_2(1-Z_{21}) - \gamma_1\theta_{11}) \,, \\ Z'_{21} &= \frac{Z_{21}(1-Z_{21})}{\theta_{21}}(\kappa_3 - \gamma_2\theta_{21}) \,. \end{aligned} \tag{11}$$

The set $\mathcal{F}(\mathcal{Z}(D_7))$ has five elements, the four faces F_k corresponding to D_2, D_6, D_8, D_{12}, and the interior of $\mathcal{Z}(D_7)$. The associated Jacobian matrices are:

$$\mathbf{J_{int(\mathcal{Z}(D_7))}} = \begin{pmatrix} -\kappa_1 & -\kappa_2 \\ 0 & 0 \end{pmatrix}, \ \mathbf{J_2} = (-\kappa_1) \,, \ \mathbf{J_6} = (0) \,, \ \mathbf{J_8} = (0) \,, \ \mathbf{J_{12}} = (-\kappa_2) \,.$$

Only $\mathbf{J_2}$ and $\mathbf{J_{12}}$ have a complete loop. Then, the algorithm looks for the stationary state on F_2 and F_{12}: $\tilde{\mathbf{Z}}^{\mathbf{2}} = (1 + \kappa_2/\kappa_1 - \gamma_1\theta_{11}/\kappa_1, 0)$ and $\tilde{\mathbf{Z}}^{\mathbf{12}} = (1 - \gamma_1\theta_{11}/\kappa_1, 1)$. The exit point candidate set is built by adding the vertices to these points. Then, the algorithm imposes that both $\tilde{Z}_1^2$ and $\tilde{Z}_1^{12}$ are in (0,1) (line 13 of algorithm 2):

$$0 < \tilde{Z}_1^2 < 1 \Rightarrow (\kappa_1 + \kappa_2 > \gamma_1\theta_{11}) \wedge (\kappa_2 < \gamma_1\theta_{11}) \,, \tag{12}$$

$$0 < \tilde{Z}_1^{12} < 1 \Rightarrow (\kappa_1 > \gamma_1\theta_{11}) \wedge (\gamma_1\theta_{11} > 0) \,. \tag{13}$$

Stability conditions for $\tilde{\mathbf{Z}}^{\mathbf{2}}$ and $\tilde{\mathbf{Z}}^{\mathbf{12}}$ are both fulfilled as $-\kappa_1 < 0$ and $-\kappa_2 < 0$, whereas stability conditions on variable $Z_l, l = 2$ requires that:

$$Z'_2(\tilde{\mathbf{Z}}^{\mathbf{2}}) < 0 \Rightarrow \kappa_3 - \gamma_2\theta_{21} < 0 \,, \tag{14}$$

$$Z'_2(\tilde{\mathbf{Z}}^{\mathbf{12}}) > 0 \Rightarrow \kappa_3 - \gamma_2\theta_{21} > 0 \,. \tag{15}$$

Condition I defined by (15) is compatible with (8), but condition (14) is not. Then, $\tilde{\mathbf{Z}}^{\mathbf{2}}$ is removed from the exit point set, whereas $\tilde{\mathbf{Z}}^{\mathbf{12}}$ is an exit point if $I_{12}^7 : I_0 \wedge I$ holds. Stability conditions on vertices are fulfilled only on vertex $\tilde{\mathbf{Z}}^{\mathbf{11}} = (0, 1)$. Then, the *exit domains* are D_{12}, D_{11}, and $QS(D_7) = \{D_{12}, D_{11}\}$.

5 Discussion and Future Work

The work described above is the first step towards the realization of a qualitative simulation algorithm that aims at (i) generating, from a given initial state, all and none

but the trajectories of a class of nonlinear ODE models of GRNs, and (ii) providing the constraints on parameters that should be satisfied so that a specific behavior occurs. At the current stage, the algorithm guarantees that, for $0 < q < \overline{q} \ll 1$, the behavior tree captures all of the sound behaviors. However, as we have not yet performed a thorough analysis with respect to entrance-exit transition, or in other words, we have not yet solved the problem (i) of identifying the only admissible connections between entrance and exit points, the behavior tree may also contains spurious behaviors. Moreover, singular perturbation analysis is a "local" procedure that works quite well in a quantitative context but that needs, in a qualitative context, to be supported by a "global" criterion when local paths are combined to produce a specific trajectory. We are quite confident that when the solution of (i) together with (ii) has been automatized, the consistency of the whole sequence of inequalities that characterizes a behavior will allow us to filter out all spurious solutions, and to prove the soundness and completeness of our algorithm. The tasks (i) and (ii) raise feasible but non-trivial algorithmic and computational issues and require proper symbolic calculation procedures.

Another important methodological issue that needs to be further studied deals with the stability of stationary points. More precisely, this poses two different problems: one related to the validity of the singular perturbation analysis in the presence of zero real part eigenvalues, and the other one to the determination of an upper bound, $\overline{q}$, of q that guarantees the Jacobian matrix stability. Although not a matter of discussion in this paper, we have already solved the latter problem, but a formal proof that stability but not asymptotic stability in the step function limit does not affect the main conclusions from the singular perturbation analysis is still lacking.

Acknowledgements. This work is carried out within the Interdepartmental CNR-BIOINFORMATICS Project. It was supported in part by the National Programme for Research in Functional Genomics in Norway (FUGE) in the Research Council of Norway (grant no. NFR153302).

References

1. de Jong, H.: Modeling and simulation of genetic regulatory systems: A literature review. J. Computational Biology 9, 67–103 (2002)
2. Glass, L., Kauffman, S.A.: The logical analysis of continuous, nonlinear biochemical control networks. J. Theoretical Biology 39, 103–129 (1973)
3. Kuipers, B.: Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge. MIT Press, Cambridge, MA (1994)
4. de Jong, H., Gouzé, J.L., Hernandez, C., Page, M., Sari, T., Geiselmann, J.: Qualitative simulations of genetic regulatory networks using piecewise linear models. Bulletin of Mathematical Biology 66, 301–340 (2004)
5. Filippov, A.F.: Differential equations with discontinuous right hand sides. Kluwer Academic Publishers Group, Dordrecht (1988)
6. Dordan, O., Ironi, L., Panzeri, L.: Some critical remarks on GNA. (in preparation)
7. Plahte, E., Kjøglum, S.: Analysis and generic properties of gene regulatory networks with graded response functions. Physica D: Nonlinear Phenomena 201, 150–176 (2005)

8. Gjuvsland, A., Plahte, E., Omholt, S.W.: Threshold-dominated regulation hides genetic variation in gene expression networks. BMC Systems Biology 1 (2007)
9. Plahte, E., Mestl, T., Omholt, S.W.: A methodological basis for description and analysis of systems with complex switch-like interactions. Journal of Mathematical Biology 36, 321–348 (1998)
10. Brazhnik, P., de la Fuente, A., Mendes, P.: Gene networks: how to put the function in genomics. Trends in Biotechnology 20, 467–472 (2002)
11. de Jong, H., Geiselmann, J., Batt, G., Hernandez, C., Page, M.: Qualitative simulation of the initiation of sporulation in *Bacillus subtilis*. Bulletin of Mathematical Biology 66, 261–300 (2004)
12. Ropers, D., de Jong, H., Geiselmann, J.: Mathematical modeling of genetic regulatory networks: Stress response in *Esterichia coli*. In: Fu, P., Latterich, M., Panke, S. (eds.) Systems and Synthetic Biology. Wiley & Sons (in press)
13. Ropers, D., de Jong, H., Page, M., Schneider, D., Geiselmann, J.: Qualitative simulation of the carbon starvation response in *Esterichia coli*. BioSystems 84, 124–152 (2006)
14. Veflingstad, S.R., Plahte, E.: Analysis of gene regulatory network models with graded and binary transcriptional responses. Biosystems 90, 323–339 (2007)
15. Holmes, M.: Introduction to Perturbation Methods. Springer, Berlin (1995)
16. Wolfram, S.: The Mathematica Book. Wolfram Media (2003)
17. Gross, J., Yellen, J.: Graph Theory and its Applications. Chapman & Hall/CRC Press, New York (2006)