

Modeling and analysis of qualitative behavior of gene regulatory networks

Alvis Brazma¹, Karlis Cerans², Dace Ruklisa³, Thomas Schlitt⁴, and Juris Viksna^{2*}

¹ European Bioinformatics Institute, Cambridge, UK brazma@ebi.ac.uk

² Institute of Mathematics and Computer Science, Riga, Latvia
karlis.cerans@lumii.lv, juris.viksna@lumii.lv

³ UCL Genetics Institute, London, UK d.ruklisa@ucl.ac.uk

⁴ Kings College London, London, UK thomas.schlitt@genetics.kcl.ac.uk

Abstract. We describe a hybrid system based framework for modeling gene regulation and other biomolecular networks and a method for analysis of the dynamic behavior of such models. A particular feature of the proposed framework is the focus on qualitative experimentally testable properties of the system. With this goal in mind we introduce the notion of the frame of a hybrid system, which allows for the discretisation of the state space of the network. We propose two different methods for the analysis of this state space. The result of the analysis is a set of attractors (generalizations of attractors used in Boolean models) that characterise the underlying biological system.

Whilst in the general case the problem of finding attractors in the state space is algorithmically undecidable, we demonstrate that our methods work for comparatively complex gene regulatory network model of λ -phage. For this model we are able to identify attractors corresponding to two known biological behaviors of λ -phage: *lysis* and *lysogeny* and also to show that there are no other stable behavior regions in state space of λ -phage model.

Introduction

Hybrid systems (HS) are a natural choice for modeling of biomolecular networks for at least two reasons: 1) HS are designed to represent and model processes that are relevant to behavior of biomolecular networks – they can describe both discrete aspects (e.g. states of activity of specific promoters) and continuous aspects (e.g. concentrations of biological substances in a cell); 2) there already exist well established mathematical techniques and supporting software tools for analysis of such hybrid system models. One of the first explicit applications of a HS based approach to the modeling of biomolecular networks has been described by Alur et al in [2], where the authors discuss a rather general class of HS models and show that such models are adequate for description and simulation of biological networks.

* The authors are listed in alphabetical order and have equally contributed to the paper.

There is a significant number of other studies discussing applications of HS to biomolecular network modeling, often proposing somewhat more restricted HS formalisms than the one used in [2] and providing examples of applications of these models describing specific biological systems (see for example [7, 8], [4], [10], [3], [15] and [1], which is by no means a comprehensive list). One of the most recent such studies [9] describes an HS based Temporal Evolution Model and applies it to modeling of *Drosophila* circadian cycle. Model checking software is then used to identify the constraints that guarantee the cyclic behavior of circadian cycle.

Our work presented here is generally in line with previous studies of application of hybrid systems to biomolecular network modeling, however it is motivated by two observations. First, it can be experimentally difficult to measure the quantitative parameters of biological systems accurately and experimental results often can be more of qualitative assay rather than a quantitative measurement (e.g., it may be possible to detect if the concentration of a particular substance is increasing or decreasing, while measuring the exact rate is much more difficult). Second, in some cases it is possible to separate the structure of the underlying regulatory network from its quantitative parameters. In this case it is natural to try to study to what extent the qualitative behavior of the system depends on the structure of the network alone, and to what extent on the exact quantitative values (relative or absolute) of the parameters.

Driven by these assumptions we propose a tailored Hybrid System Model (HSM) for description of biomolecular networks and gene regulatory networks in particular. HSM can be viewed as a restricted version of hybrid system that still provides sufficient power for modeling of biological systems, but the restrictions imposed upon HSM facilitate the analysis of the models. HSM is a continuation of the authors' previous work on Finite State Linear Model (FSLM) ([5, 13, 14]). Depending on a view point, HSM can be regarded as both simplification as well as an extension of FSLM. A variation of HSM model and applications of it to analysis of behavior of gene network of λ -phage ([12]) has been previously described by the authors in [6]. In comparison with this previous work here we present a new algorithm for analysis of the *universal state space* of the *frame* of HSM. This allows us to derive constraints that affect the behavior of the system in the process of such analysis, instead of exhaustive separate analysis of state spaces for all the possible sets of constraints that has been done previously. The mathematical formalism used here has been modified and is presented mathematically in more rigorous terms than in the previous work, nevertheless some definitions and descriptions are identical.

For the assessment of the merits of the modeling and analysis methods described here, we have applied the model to a well-studied gene network of λ -phage [12]. λ -phage is a well studied biological system (a bacterial virus), which when invading its host can exhibit two different stable behaviors *lysis* and *lysogeny*. Lysis behavior is characterized with the proliferation of the virus and death of the host, while in lysogenic behavior the genome of the virus integrates itself in the host and is replicated along side the host's genome.

It should be noted that due to the undecidability of HMS state reachability problem we can not guarantee in principle that HSM state space analysis will provide any results. Therefore it is noteworthy that for a comparatively complicated gene regulatory network model of λ -phage our method was able to identify two regions of 'stable behavior' in model's state space that correspond to the two biologically known behaviours – lysis and lysogeny . Moreover these are the only regions of stable behavior, containing correspondingly 12 and 2 states, in the state space consisting of 1024 states. Also the existence and behavior of these regions does not depend on the exact quantitative parameters of the model, but only from the structure of the network and the experimentally known qualitative information about the system's parameters (the order of the promoter affinities).

Hybrid systems for modeling gene regulatory networks

Like in most previous approaches using HS for modeling gene regulatory network (GRN), we use *modes* to represent the different possible combinations of transcription binding site states (a binding sites may be either in a vacant or an occupied state) and continuous variables for representing the concentrations of various biological substances (e.g. proteins) in a cell. In this context we assume that in each mode (i.e., in each of the possible binding site state combinations), the substance concentrations change according to continuous functions from a given set of possible such functions. The change of mode is defined by a condition in transition diagram when a substance concentration reach a certain threshold, i.e., one of the predicates "guarding" transitions from the given mode is satisfied. In a gene network these thresholds correspond to association or dissociation concentrations required for a protein to bind or dissociate from a particular binding site.

We try to keep our particular HS model HMS as simple as possible to disallow behaviors that do not correspond to known biological behaviors of biological systems, but at the same time still providing the modeling power of HS that is needed for description of biological processes that can occur in practice. Most notably we disallow instantaneous resetting of continuous values to 0 – whilst natural to other application areas of hybrid models, such resetting does not seem to have any valid biological interpretation.

Definition 1. *A Hybrid System Model (HSM) is a 6-tuple $\mathcal{H} = \langle M, X, C, T, F, MF \rangle$, where:*

1. $M = \{\mu_1, \dots, \mu_k\}$ is a finite set of modes.
2. $X = \{x_1, \dots, x_m\}$ is a finite set of continuous variables that can take real non-negative values.
3. $C = \{c_1, \dots, c_r\}$ is a finite set of transition constants that can take real non-negative values.
4. T is a set of mode transitions, where each transition $\tau \in T$ has the form $\tau = \alpha \rightarrow_p \beta$, where $\alpha, \beta \in M$ and $p = p(\tau)$ is a predicate that has a form $x \leq c$ or $x \geq c$ for some variable $x = x(\tau) \in X$ and a real non-negative constant $c = c(\tau) \in C$. Predicate p , called a guard, is a function

- $p : \mathbf{R}_+ \rightarrow \{true, false\}$ the value of which depends from the value of variable x .
5. $F = \{f_1, \dots, f_n\}$ is a set of real non-negative two argument growth/degradation functions $f_i : \mathbf{R}_+ \times \mathbf{R}_+ \rightarrow \mathbf{R}_+$ that are continuous and monotonous in both arguments and for which $f_i(z, 0) = z$ for every $z \in \mathbf{R}_+$.
 6. $MF : M \times X \rightarrow F$ is a mapping providing mode-function assignments assigning to each mode $\alpha \in M$ and each variable $x \in X$ a function $g \in F$.

An hybrid system model of 'toy example' of GRN with two genes and three protein binding sites is shown in Figure 1.

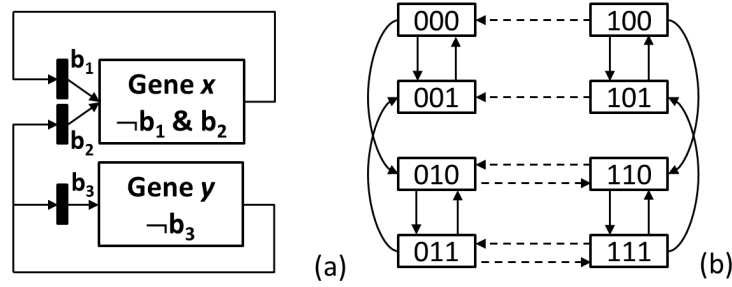


Fig. 1. A 'toy' example of GRN with two genes and three protein binding sites (a), and the corresponding HSM with 8 modes (b). Each binding site b becomes occupied if the concentration of gene product binding to it reaches an association constant a_b and becomes vacant if the concentration drops to dissociation constant d_b (where $a_b > d_b$). HSM modes are denoted by binding site states (e.g. 100 represents the situation when site b_1 is occupied and b_2 and b_3 are vacant); no concrete growth functions are shown, but functions are increasing if the values of Boolean expressions from the states of binding sites are *true* and decreasing if *false*. Transitions corresponding to variable x are shown with dashed and transitions corresponding to variable y with solid lines, constants in transition guards are not shown.

Intuitively a mode-function assignment assigns to each mode $\alpha \in M$ and each variable $x \in X$ a growth/degradation function that describes the change of this variable in time. Technically these growth/degradation functions could be viewed also simply as constituents of the corresponding modes, however if we are concentrating on qualitative aspects of such functions (e.g. are they increasing or decreasing) and not their precise form, it may be useful to distinguish between pairs of modes with identical and pairs of modes with different function assignments for some specific variable.

Each growth/degradation function f has two real valued arguments. The first of them corresponds to the value of a variable x at the time of switching to a particular mode for which f is assigned to be the function regulating x , while the second is the time Δt elapsed since the switch. For instance, in a special case when the growth rates are constant, the functions are linear and have a form

$g(x, \Delta t) = x + c \cdot \Delta t$, where c is a constant. For these particular functions the condition $g(z, 0) = z$ is also satisfied. This condition ensures the continuity of trajectories when the system switches from one mode to another.

In comparison with more general definitions of hybrid systems we have placed several restrictions on the set of functions according to which the substance concentrations can change. One such restriction is that only the mode of the system and the value of a particular substance determines the rate of concentration change until a switch to a different mode happens. In addition the monotonicity of these functions is required, meaning that in a given mode the concentration of a particular substance can only increase, decrease or stay constant. Also the form of guard predicates is restricted. The formalism is strong enough to describe biological systems, but does not provide more freedom for the behavior of the system than is necessary.

Although the role of growth/degradation functions is to describe the change of values of variables with time, this change may be non-deterministic. To describe the behavior of a HSM starting at a given initial mode with a given set of initial variable values we use a concept of *run*. A run describes an evolution of HSM with time by specifying a concrete sequence of modes through which the system is evolving and assigning a well-defined values $x(t)$ to each of the variables $x \in X$ at time moments t .

Let $\mathcal{H} = \langle M, X, C, T, F, MF \rangle$ be a HSM with given initial mode α_0 , initial time moment t_0 , and initial values of variables $X(t_0) = (x_1(t_0), \dots, x_m(t_0))$. We define a *run* of the system $\mathcal{H}$ as a (finite or infinite) sequence of modes $\alpha_i \in M$ and times t_i : $\mathcal{R}(\alpha_0, t_0, X(t_0)) = (\alpha_0, t_0) \rightarrow (\alpha_1, t_1) \rightarrow (\alpha_2, t_2) \rightarrow \dots$.

For each (α_i, t_i) t_i is the time point where $\mathcal{H}$ switches to the mode α_i . While the system is in the mode α_i , its variables change as defined by the mode function assignment in that mode (i.e. for all j : $x_j(t) = g_j(x_j(t_i), t - t_i)$, where $g_j = MF(\alpha_i, x_j)$ – a growth/degradation function assigned to x_j in mode α_i). Such an evolution continues until one of the guards is satisfied (if several are satisfied simultaneously, we can assume that one is selected by some deterministic procedure).

A particular run $\mathcal{R}(\alpha_0, t_0, X(t_0))$ assigns well-defined functions describing changes of the variables in time. For each $x_j \in X$ and $t \in [t_i, t_{i+1}]$ it defines $x_j(t) = g_j(x_j(t_i), t - t_i)$ and thus also defines a vector function $X = (x_1, \dots, x_m)$ in the whole interval $[t_0, \infty[$.

Thus a specific run of $\mathcal{H}$ describes a precise and (in principle) experimentally measurable behavior of the system, and given an appropriate HSM for some biological system, run can be regarded as simulation of the behavior of this system starting from some known initial conditions. Still, if we want to describe all the possible behaviors of the system in such a way, we (normally) need a continuum of different runs.

A natural and well explored alternative is to disregard the exact values of function X describing changes of variables, but consider only sequences of modes $\alpha_0, \alpha_1, \dots$ that can occur in runs. Provided that modes of HSM have well defined biological interpretation (e.g. modes represent activity states of genes) modes can also be more easily determined in experiments than exact values of

concentrations of substances. Still experimental measurement of modes may not be a simple task, and runs of HSM may easily have also some instantaneous transitions with $t_i = t_{i+1}$, which makes mode α_i in principle experimentally non-detectable.

We define *path* of HMS as a useful notation to describe a finite sequence of modes that can occur in some run: a finite sequence $\alpha_0, \dots, \alpha_n$ is called a *path* if there is a run $\mathcal{R}(\alpha_0, t_0, X(t_0)) = (\alpha_0, t_0) \rightarrow \dots \rightarrow (\alpha_n, t_n) \rightarrow \dots$.

Qualitative behavior of networks and frames of HSM

An appropriate HSM $\mathcal{H}$ can provide a good approximation of biological system. However such a model also involves a large number of quantitative parameters: the set of growth/degradation functions and the set of the guards governing transitions, the knowledge of which is only rarely a realistic assumption.

Nevertheless usually we can assume that set of modes M is known – a separate mode can be assigned to each state of gene activity (active or not) and/or binding site state (occupied or not)). In addition it is often possible to define the set of variables X (e.g. the set of substances in whose concentrations we are interested), and gather the information about the transitions in T (the modes and variables involved) and a partial information about the growth and degradation functions (e.g. whether a particular concentration is growing or decreasing). Guards in our model have a very simple form: either $x \leq c$ or $x \geq c$ and often we know the type of the inequality, but not the exact constant c .

To specify a HSM using only such limited information (more of qualitative than quantitative nature) about the system we introduce the notion of *frame*. A single frame is intended to represent a whole set of HSMs that are consistent with the existing knowledge about the system.

A *frame* of HSM is defined as a 5-tuple $\mathcal{F} = \langle M, X, C', T, MF' \rangle$, where the requirements on M, X, T are the same as in the definition of HSM, but we don't have a set of concrete functions F and use an assignment $MF' : M \times X \rightarrow \{\nearrow, \searrow, \rightarrow\}$ instead*. In addition for frames set C' is considered not as set of constants, but as set of *variables* taking real non-negative values. For two different transitions τ_1 and τ_2 the assignment $c(\tau_1) = c(\tau_2)$ indicates that 'constants' involved in the guards of these transitions are the same, otherwise it is assumed that these 'constants' are distinct. Normally we will use $c(\tau_1) = c(\tau_2)$ only for transitions with $x(\tau_1) = x(\tau_2)$.

Thus essentially a frame is a 'reduction' of HSM, for which only a qualitative information about growth/degradation functions and about guards is specified.

For a given HSM $\mathcal{H}$ we can easily construct the frame by using the same sets of modes M , variables X and transitions T and simply replacing C and MF with C' and MF' providing less restricted information about the transition

* For biomolecular networks the value " $\rightarrow$ " describing the situation where concentration of some substance does not change is generally reserved for the cases in which concentration is either 0 or fluctuates around the maximal biologically feasible saturation value.

guards and growth functions. We call such frame an *induced* frame of $\mathcal{H}$ and denote it by $\mathcal{F}(\mathcal{H})$.

The concept of *run* can be extended to frames. Since for frames we lack information about substance concentrations, concentration change rates and relation of concentrations to transition guards, for frames each run is specified only by an initial mode α_0 and is just a sequence of modes $\mathcal{R}(\alpha_0) = \alpha_0 \rightarrow \alpha_1 \rightarrow \alpha_2 \rightarrow \dots$, where $\alpha_i \rightarrow \alpha_{i+1}$ is allowed if and only if there is a transition $\tau = \alpha_i \rightarrow_p \alpha_{i+1} \in T$.

Runs for frames are generally non-deterministic (i.e. not uniquely specified by initial α_0); usually runs will be infinite sequences of modes, unless they terminate with some mode β that does not have outgoing transitions. Consistently with terminology used for HSM a finite sequence $\alpha_0, \dots, \alpha_n$ is called a *path* if there is a run $\mathcal{R}(\alpha_0)$ containing this sequence as an initial fragment.

In general frame $\mathcal{F} = \mathcal{F}(\mathcal{H})$ of HSM $\mathcal{H}$ typically will have a multitude of paths $\alpha, \dots, \beta$ in $\mathcal{F}$ for which there are no corresponding paths (i.e. with identical mode sequences) in $\mathcal{H}$. Moreover, for some paths $\alpha, \dots, \beta$ in $\mathcal{F}$ there may not exist any HSM $\mathcal{H}$ with $\mathcal{F} = \mathcal{F}(\mathcal{H})$ in which $\alpha, \dots, \beta$ is a path (the reason is that for frames we do not have the means to enforce consistent behavior of growth/degradation functions each time a particular mode is encountered in the run).

Thus generally frame runs may describe system's 'behaviors' that are not consistent with qualitative information about the system that has been included in specification of its frame. However it is evident that for each run $\mathcal{R}(\alpha_0, t_0, X(t_0))$ of $\mathcal{H}$ there always will be a run $\mathcal{R}(\alpha_0)$ of $\mathcal{F}(\mathcal{H})$ with the same sequence of modes (so clearly the same will hold also for paths). Therefore we do not lose any possible systems' behaviors when considering frames instead of HSM.

Whilst for a given HSM $\mathcal{H}$ its behavior can be represented by its induced frame, for analysis purposes it may be useful to characterize $\mathcal{H}$ with other frames that have more fine grained structure. To do this we introduce the notion of *frame refinement*.

Definition 2. A frame $\mathcal{F}_1 = \langle M_1, X, C', T_1, MF'_1 \rangle$ is a refinement of frame $\mathcal{F}_2 = \langle M_2, X, C', T_2, MF'_2 \rangle$ (denoted $\mathcal{F}_1 < \mathcal{F}_2$), if there are surjective mappings $m : M_1 \rightarrow M_2$ and $t : T_1 \rightarrow T_2$ such that:

1. for all $\alpha \rightarrow_p \beta \in T_1$: $t(\alpha \rightarrow_p \beta) = m(\alpha) \rightarrow_p m(\beta) \in T_2$;
2. for all $\alpha \in M_1$ and all $x \in X$: $MF'_1(\alpha, x) = MF'_2(m(\alpha), x)$;
3. the sequence $m(\alpha_0), \dots, m(\alpha_n)$ is a path in frame $\mathcal{F}_2$ for any path $\alpha_0, \dots, \alpha_n$ in frame $\mathcal{F}_1$.

Thus generally we allow partitioning of modes and (possibly) removing some transitions from partitioned modes in frame refinement as long as all the paths from the original frame are still preserved in refined frame. Therefore by the definition of refinement all the 'behaviors' of the initial frame will also be exhibited by its refinement.

In a similar manner we say that frame $\mathcal{F}$ *supports* HSM $\mathcal{H}$ (denoted by $\mathcal{F} \triangleleft \mathcal{H}$), if for all paths in $\mathcal{H}$ there are corresponding paths in supporting frame $\mathcal{F}$. It is evident that frame refinement is transitive and that any HSM $\mathcal{H}$ will always be supported by the induced frame $\mathcal{F}(\mathcal{H})$.

In HSM models of biological networks guards of transitions $x \geq c$ ($x \leq c$) can correspond to an event when the concentration of a protein described by a variable x reaches an association constant of some site and binds to it (or drops below a dissociation constant and vacates the site). The exact values of binding site affinities usually are unknown, however in cases where there are several binding sites for the same protein at least partial ordering of binding affinities may be known. To characterize such affinity orderings we use the notion of *constraints*. For a given frame $\mathcal{F} = \langle M, X, C', T, MF' \rangle$ we define a *constraint* $O(C')$ as a transitive directed acyclic graph with a set of vertices C' . A constraint effectively specifies a *strict* (i.e. non-reflexive) partial order of variables of C' : for $c_1, c_2 \in C'$ and edge (c_1, c_2) in constraint $O(C')$ is interpreted as inequality $c_1 < c_2$. We denote by $Cons(\mathcal{F})$ the set of all constraints of frame (i.e. set of all partial orderings of C'). By $O_\emptyset(C')$ we denote constraint with no edges.

Given two constraints $O_1(C')$ and $O_2(C')$ with the same set of vertices C' and sets of edges $E(O_1)$ and $E(O_2)$ the union $O_1 \cup O_2$ denotes graph with the vertex set C' and set of edges being transitive closure of $E(O_1) \cup E(O_2)$. If $O_1 \cup O_2$ contains a cycle, it is not a constraint and we say that constraints O_1 and O_2 are *incompatible*. Otherwise we say that O_1 and O_2 are *consistent*. Similarly the intersection of constraints $O_1 \cap O_2$ denotes graph with the set of edges $E(O_1) \cap E(O_2)$. $O_1 \cap O_2$ is always a constraint.

Definition 3. A constrained frame is a pair $(\mathcal{F}, CA)$, where $\mathcal{F} = \langle M, X, C', T, MF' \rangle$ is a frame and $CA : T \rightarrow Cons(\mathcal{F})$.

We say that two constraint assignments CA_1 and CA_2 are *incompatible* if for some transition τ constraints $CA_1(\tau)$ and $CA_2(\tau)$ are incompatible. Otherwise we say that CA_1 and CA_2 are *consistent*. For consistent CA_1 and CA_2 it is convenient to denote by $CA_1 \cup CA_2$ the assignment of constraint $CA_1(\tau) \cup CA_2(\tau)$ to each of transitions τ . In analogous way we can also define $CA_1 \cap CA_2$. For a constraint $O \in Cons(\mathcal{F})$ by CA_O we denote the assignment of single constraint O to all transitions.

In constrained frame each transition is annotated with a partial ordering of C' and we consider transition as 'available' only if its own transition constant has the highest priority among all the transitions from the same mode and involving the same variable in this ordering. Moreover we require that in constrained frames a sequence of modes forming a run should be 'passable' by sequence of transitions that does not include any transition pairs with incompatible constraint assignments.

To make this more precise we define constraint $O(\alpha, \beta)$ as intersection of all constraints $CA(\tau)$ for transitions $\tau = \alpha \rightarrow_p \beta \in T$ for each pair of modes α, β in constrained frame $(\mathcal{F}, CA)$. Furthermore, for each sequence of modes $P = \alpha_0, \dots, \alpha_n$ we define a graph $O(P)$ as union of all constraints $O(\alpha_i, \alpha_{i+1})$, $i = 0 \dots n - 1$. Informally, if $O(P)$ is a constraint (i.e. without cycles) it can be regarded as the least restrictive *single* constraint under which the sequence of modes P is 'passable'.

For constrained frames it is convenient to reverse definitions of path and run – a finite sequence $P = \alpha_0, \dots, \alpha_n$ is a *path* if $O(P)$ is a constraint. Then

$\mathcal{R}(\alpha_0) = \alpha_0 \rightarrow \alpha_1 \rightarrow \alpha_2 \rightarrow \dots$ is a *run* if each initial fragment of sequence of its modes $\alpha_0, \dots, \alpha_n$ is a path. For finite runs $\mathcal{R}(\alpha_0) = \alpha_0 \rightarrow \dots \rightarrow \alpha_n$ we also require that $\mathcal{R}(\alpha_0)$ is not a proper prefix of some other run $\hat{\mathcal{R}}(\alpha_0)$.

The notion of frame refinement can be extended to constrained frames. The definition of constrained frame $(\mathcal{F}_1, CA_1)$ being refinement of $(\mathcal{F}_2, CA_2)$ (denoted by $(\mathcal{F}_1, CA_1) < (\mathcal{F}_2, CA_2)$) is analogous to Definition 2 with an additional requirement that mapping t of transitions should be consistent with constraint assignments (i.e. $CA_1(t(\tau)) = CA_2(\tau)$) and the requirement for path conservation referring to paths in constrained frames.

Let us consider HSM $\mathcal{H}$ describing some gene regulatory network. If we have complete knowledge about the ordering of affinities C of binding sites in $\mathcal{H}$, this complete ordering being represented by constraint O , it would be adequate to describe this GRN with induced frame $\mathcal{F}(\mathcal{H})$ and constraint assignment CA_O . We call such constraint assignment CA_O *maximal* and for given $\mathcal{H}$ denote it by $Max(\mathcal{H})$. The induced constrained frame is correspondingly denoted by $(\mathcal{F}(\mathcal{H}), Max(\mathcal{H}))$. This notation allows to extend definition of frame support for HSM to constrained frames: constrained frame $(\mathcal{F}, CA)$ supports $\mathcal{H}$ (denoted by $(\mathcal{F}, CA) \triangleleft \mathcal{H}$) if $(\mathcal{F}, CA) < (\mathcal{F}(\mathcal{H}), Max(\mathcal{H}))$.

If however we do not possess any knowledge about the orderings of affinities we can consider the least constrained frame that will support $\mathcal{H}$ regardless of what is affinity ordering in $\mathcal{H}$. In such frame we assign to all transitions in $\mathcal{F}$ constraints that give them the highest priority – i.e. to each transition $\alpha \rightarrow_p \beta \in T$ with guard p of the form $x(\tau) \leq c(\tau)$ we assign a constraint with edge set $\{(c(\tau), c(\hat{\tau})) | \hat{\tau} = \alpha \rightarrow_p \beta \in T, \tau \neq \hat{\tau}, x(\tau) = x(\hat{\tau})\}$ (or edge set $\{(c(\hat{\tau}), c(\tau)) | \hat{\tau} = \alpha \rightarrow_p \beta \in T, \tau \neq \hat{\tau}, x(\tau) = x(\hat{\tau})\}$ if p is of form $x(\tau) \geq c(\tau)$). We call the resulting constraint assignment *minimal* and for frame $\mathcal{F}$ denote it by $Min(\mathcal{F})$. (In contrast to maximal assignment, which is based on underlying HSM $\mathcal{H}$, minimal assignment is defined solely on basis of properties of $\mathcal{F}$).

Analysis of dynamic behavior of hybrid system models

Given a constrained frame $(\mathcal{F} = \langle M, X, C', T, MF' \rangle, CA)$ describing some system, all its possible behaviors are represented by runs that are allowed under the specified constraints. We can conveniently characterize all such runs simply by graph with vertices corresponding to modes of $\mathcal{F}$ and edges corresponding to these transitions in $\mathcal{F}$ that are allowed to appear in runs by constraint assignment CA . We call such graph a *state space graph* of $\mathcal{F}$ and denote it by $G = G(\mathcal{F}, CA)$. The vertex set of $V(G)$ is simply M , the set of edges $E(G)$ is subset of $\{(\alpha, \beta) | \alpha \rightarrow_p \beta \in T\}$.

There are notable similarities between frame state space graphs and state space graphs in Boolean gene network models. For Boolean models the analysis of their space graphs is also simple and gives unambiguous characterization of the system's behavior – the graphs decompose into cyclic *attractor* subgraphs, each of which can be regarded as a descriptor of one of the possible behaviors of the system ([11]). The information that frame state space provide about the system's behavior is very similar to that given by state space graphs of Boolean models.

Unfortunately however frame state space graphs can be much more complex and do not allow simple partitioning into attractor basins.

Still we are interested to identify regions in frame state space graphs $G = G(\mathcal{F}, CA)$, which will characterize 'stable behaviors' of the system. In order to achieve this task we propose to partition G into *strongly connected components* (SCC) and to relate these components to stable behaviors. In the worst case however the whole G can consist just of a single SCC and such a partitioning will tell us nothing useful about the system's behavior. (The proof of Turing completeness of FSLM [13] can be also applied to frame state space graphs, implying that such cases are in principle unavoidable.)

We can perform a more detailed analysis of the dynamics than just compute a partition of G into SCCs. Let us consider a SCC $S \subseteq V(G)$ and a variable x , such that for all modes $\alpha \in S$: 1) we have $MF'(\alpha, x) = \nearrow$ (or $MF'(\alpha, x) = \searrow$), and 2) there is an edge $(\alpha, \beta) \in E(G)$ derived from transition $\tau = \alpha \rightarrow_p \beta$ with the guard of the form $x \leq c(\tau)$ (correspondingly $x \geq c(\tau)$). In such situation we can conclude that the system can stay in S only for a limited time, since eventually one of guards for these transitions will get satisfied. In such cases we say that SCC S is *transitional*.

Definition 4. A *strongly connected non-transitional component of constrained state space graph* $G(\mathcal{F}, CA)$ is called an *attractor*.

Finding of attractors require splitting state space graphs in SCCs and checking whether each SCC is or is not transitional. The latter task can be achieved in linear time with respect to the size of graph.

Suppose that we have HSM model $\mathcal{H}$ for some gene regulatory network. We assume that we have only qualitative knowledge about parameters of $\mathcal{H}$, i.e., we have complete knowledge whether growth/degradation functions are increasing or decreasing and possibly a partial knowledge about the ordering of transition constants. Such amount of available information seems to be typical for many biological networks. In terms of constrained frames our knowledge about the system is represented by induced constrained frame $(\mathcal{F}(\mathcal{H}), CA)$ with a set of constraints CA ranging somewhere between $Min(\mathcal{F}(\mathcal{H}))$ (no knowledge about transition constants) and $Max(\mathcal{H})$ (complete knowledge about transition constants). Usually however some information about constant ordering is available – e.g. if constants correspond to association and dissociation affinities of binding sites, then for the same binding site association affinity must be the largest of these two. Additional constraints derived from the known biological facts also might be introduced. Let us denote these *known* constraint assignments by $External(\mathcal{H})$.

In order to make some judgments about possible $\mathcal{H}$ behaviors we are interested to find sets of all attractors of all frames $(\mathcal{F}(\mathcal{H}), CA)$ for which CA is consistent with $External(\mathcal{H})$. There are two natural ways to do this.

Firstly, we could consider all *complete orderings* $Ord \in Cons(\mathcal{F}(\mathcal{H}))$ of transition constants that are consistent with $External(\mathcal{H})$ and analyze attractor structure of all the corresponding graphs $G(\mathcal{F}(\mathcal{H}), CA_{Ord})$ (thus essentially we check for attractor structure of graphs corresponding to all the possible choices

of $Max(\mathcal{H})$). Such approach has been used by the authors in [6] for analysis of λ -phage model. Despite comparatively large size of this model (11664 modes and 32 transition constants) the number of different orderings that have to be considered is comparatively small – only 42. The main reason for this is (easy provable) fact that attractor structure is influenced only by orderings of subsets of constants $\{c(\tau)|x(\tau) = x\}$ for each variable $x \in X$. Therefore for HSM models where transitions are defined by binding site affinities, in the special case when for each binding factor there is only single binding site it affects, there will be only a single ordering Ord to consider. For λ -phage model there are multiple binding sites for most of the binding factors and 'biologically known' constraints are used to reduce the number of possible orderings from few thousands to 42. One of the results we have presented in [6] is the fact that the attractor structure remains the same for all these 42 orderings and contains only 2 attractors that correspond to two known biological behaviors of λ -phage: *lysis* and *lysogeny*.

An alternative approach on which we focus in this paper is try to analyze directly attractor structure $G(\mathcal{F}, CA)$ using only known limited knowledge of CA that is given by $External(\mathcal{H})$ and $Min(\mathcal{F}(\mathcal{H}))$ without explicit consideration of all the possible choices of CA . Such an approach has several advantages: first, if two different assignments CA_1 and CA_2 have either identical state space graphs, or at least graphs with identical attractor structure, we probably can save some work by noticing shared parts in these graphs; second this could help to decide whether the behaviors described by the attractors yielded by different choices of CA are essentially the same or different; third such an approach could help to derive automatically the conditions (i.e. constraint assignments) that separate different qualitative behaviors.

To perform such analysis we can start with initially given state space graph $G_0 = G(\mathcal{F}(\mathcal{H}), Min(\mathcal{F}(\mathcal{H})) \cup External(\mathcal{H}))$, check for parts of G_0 that will 'behave' differently if different additional restrictions will be imposed on existing constraints, and try to partition these parts in such a way that different behaviors are represented by different parts of these partitions. Let G_1 be a graph obtained by such a process from G_0 . By repeating this process we will obtain (hopefully finite) sequence of graphs $G_0, G_1, \dots, G_n$, where G_n can not be further partitioned. By analyzing attractor structure of G_n we can then expect to find attractors for all different choices of constraints compatible with initial constraints $Min(\mathcal{F}(\mathcal{H})) \cup External(\mathcal{H})$ and, moreover, hope, although that is not guaranteed, that attractors completely defined by some partial ordering will be present only in a single copy.

It turns out that the process of computing space graph sequence $G_0, G_1, \dots, G_n$ essentially can be regarded as a process of constructing appropriate constrained frame refinements $(\mathcal{F}_0, CA_0), \dots, (\mathcal{F}_n, CA_n)$, where $(\mathcal{F}_0, CA_0) = (\mathcal{F}(\mathcal{H}), Min(\mathcal{F}(\mathcal{H})) \cup External(\mathcal{H}))$, $(\mathcal{F}_n, CA_n) < \dots < (\mathcal{F}_0, CA_0)$ and $G_n = G(\mathcal{F}_n, CA_n)$. An example in Figure 2 shows that such refinement process indeed can reduce number of states in attractors of state space graph. For computing the sequence of graphs we propose the following *RefineStateSpace* algorithm.

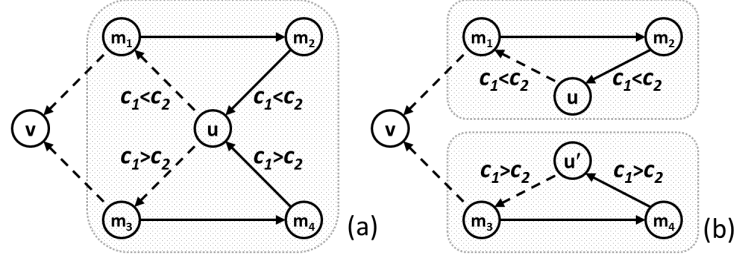


Fig. 2. A simple example showing that refinement of constrained frame (b) can have attractors with fewer number of states than in original frame (a). In original frame there is SCC containing states m_1, m_2, m_3, m_4, u , which have incompatible constraints for two different paths between states u and v . In refinement this SCC has been split in two SCCs with states m_1, m_2, u and m_3, m_4, u' . Solid and dashed lines represent correspondingly transitions and paths (each path segment can consist also of a single transition, however the constraints shown refer to the whole paths).

Algorithm *RefineStateSpace*

Input: State space graph $G(\mathcal{F}_0, CA_0)$

Output: Refined state space graph $G(\mathcal{F}, CA)$

1. $\mathcal{F} = \langle M, X, C', T, MF' \rangle \leftarrow \mathcal{F}_0, CA \leftarrow CA_0$
2. for each $\alpha \in M$ assign $m^*(\alpha) \leftarrow \alpha$ and $trans(\alpha) \leftarrow O_\emptyset(C')$
3. **while** there exist $\alpha, \beta \in M$ and different paths $P_1 = \alpha, \gamma_1, \dots, \beta, P_2 = \alpha, \gamma_2, \dots, \beta$ such that:
 4. $O(P_1)$ and $O(P_2)$ are incompatible
 5. $m^*(\delta_1) \neq m^*(\delta_2)$ for all pairs of internal vertices $\delta_1 \in P_1$ and $\delta_2 \in P_2$
 6. there are no alternative pair P'_1 and P'_2 of such paths with either $O(P'_1)$ or $O(P'_2)$ being proper subgraph of either $O(P_1)$ or $O(P_2)$
7. **do** the following:
 8. let $\tau_1 = \alpha \rightarrow \gamma_1 \in T$ and $\tau_2 = \alpha \rightarrow \gamma_2 \in T$ be transitions corresponding to P_1 and P_2
 9. $\hat{M} \leftarrow M, \hat{T} \leftarrow T, \hat{CA} \leftarrow CA$
 10. remove α from $\hat{M}$ and add to $\hat{M}$ two new modes α_1 and α_2
 11. assign $m^*(\alpha_1) \leftarrow m^*(\alpha), m^*(\alpha_2) \leftarrow m^*(\alpha), trans(\alpha_1) \leftarrow CA(\tau_1), trans(\alpha_2) \leftarrow CA(\tau_2)$
 12. remove transitions and constraint assignments involving α from $\hat{T}$ and $\hat{CA}$
 13. for each $\tau = \alpha \rightarrow \delta \in T$, with $\tau \neq \tau_1, \tau \neq \tau_2$ add transition $\hat{\tau} = \alpha_1 \rightarrow \delta$ to $\hat{T}$ and assign $\hat{CA}(\hat{\tau}) \leftarrow CA(\tau)$
 14. for each $\tau = \alpha \rightarrow \delta \in T$, with $\tau \neq \tau_1, \tau \neq \tau_2$ add transition $\hat{\tau} = \alpha_2 \rightarrow \delta$ to $\hat{T}$ and assign $\hat{CA}(\hat{\tau}) \leftarrow CA(\tau)$
 15. add transition $\hat{\tau} = \alpha_1 \rightarrow \gamma_1$ to $\hat{T}$ and assign $\hat{CA}(\hat{\tau}) \leftarrow CA(\tau_1)$
 16. add transition $\hat{\tau} = \alpha_2 \rightarrow \gamma_2$ to $\hat{T}$ and assign $\hat{CA}(\hat{\tau}) \leftarrow CA(\tau_2)$

17. for each $\tau = \delta \rightarrow \alpha \in T$ add transition $\hat{\tau} = \delta \rightarrow \alpha_1$ to $\hat{T}$ and assign $\hat{CA}(\hat{\tau}) \leftarrow CA(\tau) \cup CA(\tau_1)$
18. for each $\tau = \delta \rightarrow \alpha \in T$ add transition $\hat{\tau} = \delta \rightarrow \alpha_2$ to $\hat{T}$ and assign $\hat{CA}(\hat{\tau}) \leftarrow CA(\tau) \cup CA(\tau_2)$
19. $M \leftarrow \hat{M}, T \leftarrow \hat{T}, CA \leftarrow \hat{CA}$
20. **return** $G(\mathcal{F}, CA)$

Algorithm *RefineStateSpace* by its nature is greedy and heuristic and its output depends from the order in which pairs of paths P_1 and P_2 are selected in *Step 3*. Nevertheless we can guarantee that algorithm terminates and outputs state space graph of constrained frame $(\mathcal{F}, CA)$ that (in most cases) is refinement of the initial $(\mathcal{F}(\mathcal{H}), \text{Min}(\mathcal{F}(\mathcal{H})) \cup \text{External}(\mathcal{H}))$.

Let us denote by $n(\mathcal{H})$ and $m(\mathcal{H})$ correspondingly the number of modes and the number of transitions in $\mathcal{H}$. Let denote by $N(\mathcal{H})$ the number of different complete (with respect to each of the variables) orderings of transition constants of $\mathcal{H}$ that are consistent with $\text{Min}(\mathcal{F}(\mathcal{H})) \cup \text{External}(\mathcal{H})$.

Proposition 1. *Algorithm RefineStateSpace terminates after finite number of steps and outputs state space graph G with at most $n(\mathcal{H}) \times N(\mathcal{H})$ vertices.*

Proof. For each vertex of initial graph $\alpha \in V(G_0)$ consider set $S_\alpha = \{\beta \in V(G) \mid m^*(\beta) = \alpha\}$. By the design of algorithm for each pair of different vertices $\beta_1, \beta_2 \in S_\alpha$ constraints $\text{trans}(\beta_1)$ and $\text{trans}(\beta_2)$ are incompatible. Thus there are at most $N(\mathcal{H})$ vertices in each such set S_α and therefore at most $n(\mathcal{H}) \times N(\mathcal{H})$ vertices in $V(G)$.

The straightforward implementation of the algorithm of each **while** step of the algorithm can be done in $O(N(\mathcal{H})(n(\mathcal{H}) + m(\mathcal{H})))$ time, leading to comparatively high total time complexity $O((N(\mathcal{H})(n(\mathcal{H}) + m(\mathcal{H})))^2)$. However it is not difficult although technically somewhat involved to provide implementation with $O(N(\mathcal{H})n(\mathcal{H})(n(\mathcal{H}) + m(\mathcal{H})))$ running time, which already compares well with running state space analysis separately for each of $N(\mathcal{H})$ complete orderings and analyzing the whole set of $N(\mathcal{H})$ obtained graphs afterward.

The way in which algorithm constructs $(\mathcal{F}, CA)$ from initial $(\mathcal{F}_0, CA_0)$ closely corresponds to the way constrained frame refinement is defined – essentially algorithm explicitly constructs the mode and transition mappings m and t . Moreover, when mode α is split into modes α_1 and α_2 each path involving transition $\alpha \rightarrow \delta$ will be preserved by replacing α either with α_1 or α_2 (or any of them). Thus, if all runs of frame $(\mathcal{F}_0, CA_0)$ are infinite we have $(\mathcal{F}, CA) < (\mathcal{F}_0, CA_0)$. For finite runs however there is a possibility in each step to lose the last mode of the sequence. It was not relevant to our examples, but if needed algorithm can be adjusted to explicitly preserve such modes.

Proposition 2. *If there are no finite runs of initial frame $(\mathcal{F}_0, CA_0)$ then for $G(\mathcal{F}, CA)$ computed by algorithm RefineStateSpace the following holds:*

1. *For each complete (with respect to each of the variables) ordering Ord of transition constants and an attractor S in state space graph $G(\mathcal{F}_0, CA_{\text{Ord}})$*

there is an attractor $\hat{S}$ in $G(\mathcal{F}, CA)$ with the same number of vertices and preserving all paths in S .

2. For each attractor S in $G(\mathcal{F}, CA)$ there is a complete ordering Ord with state space graph $G(\mathcal{F}_0, CA_{Ord})$ containing attractor $\hat{S}$ with the same number of vertices and preserving all paths in S .

Proof. 1. We have already shown that all infinite paths and thus the whole attractor $\hat{S}$ will be preserved in $G(\mathcal{F}, CA)$. Let $\hat{S}$ be one of attractors with minimal number of modes such that subset $A \subseteq \hat{S}$ is mapped to S . If there exists non-mapped mode $\alpha \in \hat{S}$ and $\alpha \notin A$ then due to minimality of $\hat{S}$ constraints for all transitions to α must be compatible with Ord , thus we should have $\alpha \in A$.

2. Consider infinite path P in $\hat{S}$ and constraint $O(P)$. Then for every complete ordering Ord compatible with $O(P)$ there will be corresponding attractor S in $G(\mathcal{F}_0, CA_{Ord})$ to which $\hat{S}$ can be mapped. Clearly all modes of S should also have preimages in $\hat{S}$.

Experimental results

For the toy example in Figure 1 we have number of modes $n(\mathcal{H}) = 8$ and number of different orderings $N(\mathcal{H}) = 6$. By performing analysis of space graphs for all the 6 orderings we find 5 different attractors: A_1 (8 states), A_2 (4 states), A_3 , A_4 and A_5 (each with 2 states). The state graphs for 6 possible orderings correspondingly contain the following sets of attractors: A_1 ; A_2 ; A_2 and A_3 (for two different orderings); A_4 ; A_4 and A_5 .

When we use algorithm *RefineStateSpace* for analysis of this model, we find the same set of attractors. Number of states in refined graph depends on the order in which algorithm chose the pairs of paths to construct refinements. The minimal number of states that can be obtained is 40 (in this case it is not hard to show that this number is the smallest possible). Figure 3 shows the part of this state space with 20 states. For modes only the states for binding sites b_2 and b_3 are shown here and we can split the whole state space in two identical subgraphs depending from the state of binding site b_1 . For these two parts only pairs of identical states of b_2 and b_3 are connected with transitions (either in one or both directions). (Such additional simplification is possible since in this specific toy model gene y is not affected by the behavior of gene x .)

For our λ -phage model we have $n(\mathcal{H}) = 11664$ and $N(\mathcal{H}) = 42$. For each Ord of these 42 orderings the state space of $G(\mathcal{F}(\mathcal{H}), CA_{Ord})$ contains only the same 2 attractors that correspond to *lysis* (12 states) and *lysogeny* (2 states) behaviors of λ -phage. We have applied algorithm *RefineStateSpace* to partial λ -phage models, which, as expected, gave the same results. Currently we are working on more efficient implementation of the algorithm to make complete analysis of λ -phage model feasible. This will provide estimates of the size of refined state space and computational time requirements.

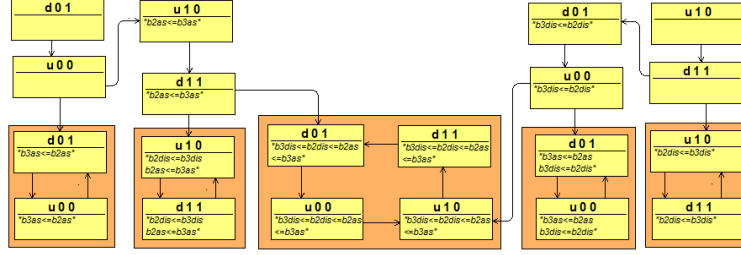


Fig. 3. Refinement of constrained state space graph with 20 modes of toy GRN from Figure 1. Half of all states (corresponding to vacant binding site b_1) are shown. From left to right attractors are shown in order A_5, A_3, A_1, A_4, A_2 . Attractors A_1 and A_2 extend to identical second half of state space corresponding to occupied site b_1 and have two times more states than shown here.

Conclusions and discussion

Whilst it is known that in general case the method we propose here may not be able to produce non-trivial results (there will be cases with the whole state space consisting just of a single attractor), we have shown that it can be useful for analysis of specific models, including comparatively complicated model of λ -phage. The results for few existing models are encouraging – all the attractors we have found by analysis of qualitative information contained in frames describe behaviors that can be achieved by underlying HSM with explicitly defined quantitative parameter values. Moreover an interesting observation is that the same attractor structure can be shared by many different orderings of transition constants (although the complete state space graphs are different).

Regarding future work, firstly there is a range of questions concerning the formalism of constrained frames that we have developed here as well as algorithms for state space analysis - e.g. what can we say about general mathematical structure of frame refinements, can we design an efficient algorithm that always computes the refinement with minimal possible number of states. Here we have not concentrated much on these questions, partially because the underlying mathematical formalism has already few times been adjusted and it is not clear as yet which parts are really relevant for analysis of HSM representing real biological systems.

Secondly, there are questions how our approach can be extended to better answer questions of biological nature. For instance, we have already indicated the objective to include in models additional information about comparative growth rates of growth/degradation functions (there are examples that show that such inclusion may be useful when current approach of frame analysis is not able to differentiate between different behaviors of underlying HSM). Other question of biological significance is identification and study of constraints that are crucial for directing the system's behavior towards specific attractor – are such constraints consistent over the whole state space, and for models of real

biological systems, can we link such constraints with some known biological events?

References

1. J. Ahmad, J. Bernot, J. Comet, D. Lime, and O. Roux. Hybrid modelling and dynamical analysis of gene regulatory networks with delays. *Complexus*, 3:231–251, 2007.
2. R. Alur, C. Belta, F. Ivancic, V. Kumar, M. Mintz, G. Pappas, H. Rubin, and J. Schug. Hybrid modeling and simulation of biomolecular networks. In *Proc. of HSCC 2001*, volume 2034 of *LNCS*, pages 19–32, 2001.
3. G. Batt, R. Ben Salah, and O. Maler. On timed models of gene networks. In *Proc. of FORMATS 2007*, volume 4763 of *LNCS*, pages 38–52, 2007.
4. G. Batt, D. Ropers, H. de Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in *escherichia coli*. *Bioinformatics*, 21 Suppl. 1:19–28, 2005.
5. A. Brazma and T. Schlitt. Reverse engineering of gene regulatory networks: a finite state linear model. *Genome Biol.*, 4:P5:1–31, 2003.
6. R. Brazma, K. Cerans, D. Ruklisa, T. Schlitt, and J. Viksna. HSM – a hybrid system based approach for modelling intracellular networks. *Gene*, 518:70–77, 2013.
7. H. de Jong, J. Geiselmann, C. Hernandez, and M. Page. Genetic network analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.
8. H. de Jong, J. Gouze, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol.*, 66:301–340, 2004.
9. J. Fromentin, D. Eveillard, and O. Roux. Hybrid modeling of biological networks: mixing temporal and qualitative biological properties. *BMC Syst. Biol.*, 4:79:11, 2010.
10. R. Ghosh and C. Tomlin. Symbolic reachable set computation of piecewise affine hybrid automata and its application to biological modelling: Delta-notch protein signalling. *Syst. Biol.*, 1:170–183, 2004.
11. S. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, 1969.
12. H. McAdams and L. Shapiro. Circuit simulation of genetic networks. *Science*, 269:650–656, 1995.
13. D. Ruklisa, A. Brazma, and J. Viksna. Reconstruction of gene regulatory networks under the finite state linear model. *Genome Inform.*, 16:225–236, 2005.
14. T. Schlitt and A. Brazma. Modelling in molecular biology: describing transcription regulatory networks at different scales. *Philos. Trans. R. Soc. Lond. B*, 361:483–494, 2006.
15. H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *Theor. Comput. Sci.*, 391:258–275, 2008.