Parameter Inference for Asynchronous Logical Networks Using Discrete Time Series

Hannes Klarner, Heike Siebert, Alexander Bockmayr
DFG Research Center Matheon
FB Mathematik und Informatik
Freie Universität Berlin, 14195 Berlin, Germany
klarner@zedat.fu-berlin.de

ABSTRACT

This paper is concerned with the dynamics of asynchronous logical models of regulatory networks as introduced by R. Thomas. Available knowledge about the dynamics of a regulatory network is often limited to a sequence of snapshots in the form of a discrete time series. Using CTL formulas together with the concept of partially monotone paths, a methodology is elaborated to investigate the compatibility of a given time series and a Thomas model. The approach can be used to revise the model, but also to evaluate the given data. Additionally, suggestions are made to analyze a model pool for common properties regarding component behavior and interaction types, aiming at results exploitable for experimental design.

Categories and Subject Descriptors

J.3 [**LIFE AND MEDICAL SCIENCES**]: Biology and genetics; F.4.1 [**Mathematical Logic**]: Temporal logic

1. INTRODUCTION

In molecular biology, a regulatory network is a description of interactions between components. By assigning activity levels to the components and allowing interacting components to influence their activities depending on parameter values, such networks can be used to describe the system's dynamics in a state space. Since a full set of kinetic parameters is often not available, discrete modeling frameworks with finite parameter space have been suggested as an alternative to systems of differential equations.

Formal methods can help in determining suitable values for discrete parameters, translating available data into constraints on the set of all possible parameter choices, see e.g. Batt et al. [9] or Corblin, Fanchon, Trilling [4]. In this article, we employ similar ideas to test assumptions about component interplay for consistency. In case of inconsistencies, new hypotheses are systematically derived that then can be investigated experimentally. In contrast to related work, we

additionally use our methods to evaluate the given experimental data by analyzing time series for potential ranges of poor sampling.

The paper is organized as follows. In Section 2 we recall the logical framework for regulatory networks and temporal logic. In Section 3 we introduce the notion of discrete time series as an ordered sequence of partial states. Section 4 elaborates a method of incorporating specific assumptions about monotonicity in between partial states. These are related to potential unobserved oscillations and can be used to evaluate the sufficiency of the provided data. In Section 5 we suggest a modeling workflow utilizing our methods, assessing the modeling assumptions as well as the quality of a given time series in terms of its temporal resolution, and discuss scalability and computational issues. We illustrate the procedure using an application example in Section 6, and conclude the paper discussing perspectives and future work.

2. PRELIMINARIES

This section introduces our discrete modeling framework and model checking terminology. Throughout, discrete intervals will be denoted by

$$[a, b] := \{k \in \mathbb{N} \mid a \leq k \leq b\}, \quad \text{for } a, b \in \mathbb{N}.$$

The in- and out-degrees of a vertex of a graph are denoted by $d_-(v)$ and $d_+(v)$ and its pre- and successor sets by $V_-(v)$ and $V_+(v)$, respectively.

2.1 Regulatory Networks

The discrete framework for modeling regulatory systems as introduced by Thomas in [1] consists of an edge-labeled digraph called regulatory network and a set of integer parameters.

DEFINITION 2.1 (REGULATORY NETWORK). *A regulatory network $G = (V, E, t)$ is a directed graph with vertices $V := [1, n]$ for some fixed $n \in \mathbb{N}$, edges $E \subseteq V \times V$, maximal activity levels*

$$p : V \to [0, max(1, d_+(v))],$$

and a function

$$t : E \to \mathbb{N}, \ such \ that \ t(u, v) \in [0, p(u)]$$

that assigns thresholds to the edges $e \in E$. Nodes are called components and edges are called interactions. For a component $v \in V$, a predecessor $w \in V_-(v)$ is called a regulator of

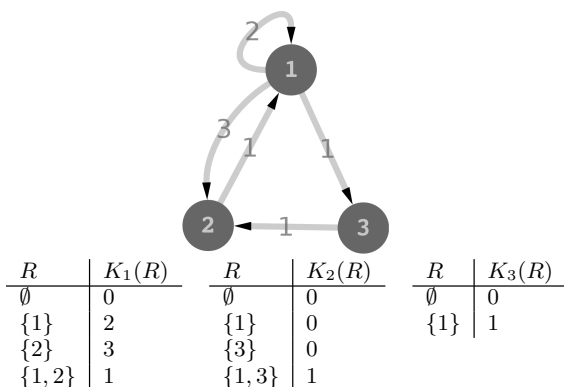v and a subset of regulators $R \subseteq V_-(v)$ is called a *regulatory context* of v.

The vertices of the graph can be interpreted as variables taking values in the respective activity level interval $[0, p(v)]$. In the simplest case all variables are boolean. The edge labels are integers that represent thresholds above which regulatory interactions become effective.

DEFINITION 2.2 (PARAMETER SET). *Given a regulatory network (V, E, t), a parameter set $K = \{K_v \mid v \in V\}$ is a set of functions*

$$K_v : 2^{V_-(v)} \to [0, p(v)].$$

K_v *is also called v-parameter subset of K.*

The network and parameter set in Fig. 1 will serve as a running example throughout the paper. Here we choose the maximal activity levels $p(v) = d_+(v)$ for all vertices. Any

R	$K_1(R)$		R	$K_2(R)$		R	$K_3(R)$
$\emptyset$	0		$\emptyset$	0		$\emptyset$	0
$\{1\}$	2		$\{1\}$	0		$\{1\}$	1
$\{2\}$	3		$\{3\}$	0			
$\{1,2\}$	1		$\{1,3\}$	1			

Figure 1: Example network and parameter set

collection of parameter sets of a regulatory network is called a parameter pool. In particular, we define:

DEFINITION 2.3 (PARAMETER SPACE). *The collection of all parameter sets of a regulatory network (V, E, t) is denoted by*

$$\mathcal{K}(V, E, t) := \{K \mid K \text{ is a parameter set of } (V, E, t)\}$$

and called the parameter space of (V, E, t).

The number of sets in the parameter space depends on the maximal activity levels of the network components and their in-degrees:

$$|\mathcal{K}(V, E, t)| = \prod_{v \in V} (p(v) + 1)^{2^{d_-(v)}}.$$

The size of the parameter space of our running example is $|\mathcal{K}(V, E, t)| = 4^4 \cdot 2^2 \cdot 2^4 = 16384$.

The dynamics of a regulatory network (V, E, t) with parameters K is represented by a directed graph, called the state transition graph. It can be thought of as the discrete analogue to all possible trajectories in the phase plane of an ODE model. The nodes of this graph represent the discrete states of the system.

DEFINITION 2.4 (STATE SPACE). *Given a regulatory network (V, E, t), the state space X is given by*

$$X = \prod_{v \in V} [0, p(v)].$$

To define the transitions between states it is convenient to turn the parameter set K into a function F on the state space X, where

$$F : X \to X, \quad x \mapsto F(x) = (f_1(x), \dots, f_n(x)).$$

The image of x under component function f_v is defined to be a particular parameter $K_v(R)$. To choose this particular parameter we define the present regulators of v in a state x.

DEFINITION 2.5 (PRESENT REGULATORS). *Given a regulatory network (V, E, t) with parameters K and its associated state space X, the present regulators $R_v(x)$ of a component $v \in V$ in state $x \in X$ are*

$$R_v(x) := \{w \in V \mid (w, v) \in E \wedge x_w \geq t(w, v)\}.$$

The present regulators of v in state x are components w that regulate v and whose activity level in state x is above the threshold $t(w, v)$. This definition is the one given by Chaouiya et al. in [3]. With this notation the image of x under F is now defined to be

$$F(x) := (K_1(R_1(x)), \dots, K_n(R_n(x))).$$

The present regulators of component 3 of the running example in state $x = (1, 1, 0)$ are $R_1(x) = \{2\}$, $R_2(x) = \emptyset$ and $R_3(x) = \{1\}$. Thus $F(x) = (2, 0, 1)$ according to the table given in Fig. 1.

There are several strategies for obtaining transitions using F. Most common are synchronous, asynchronous or priority strategies. The method described in this article is designed to work with unitary asynchronous transition rules. A simple notation for the transitions of the unitary asynchronous state transition graph is achieved with the tendencies f'_v of the component functions f_v of $F = (f_1, \dots, f_n)$, as suggested by Richard in [2].

DEFINITION 2.6 (TENDENCIES). *The tendency f' of a component function $f_v : X \to \mathbb{N}$ is defined to be*

$$f'_v(x) = \begin{cases} 1 : & f_v(x) - x_v > 0 \\ 0 : & f_v(x) - x_v = 0 \\ -1 : & f_v(x) - x_v < 0 \end{cases}.$$

The tendency of component 2 in state $(0, 1, 0)$ of the example parameter set is $f'_2(0, 1, 0) = -1$.

DEFINITION 2.7 (STATE TRANSITION GRAPH). *Given a regulatory network (V, E, t) with parameter set K, the (unitary) asynchronous state transition graph is a directed graph (X, T), where the nodes X are the elements of the state space associated with the regulatory network and the edges T are transitions between states. We have $(x, y) \in T$ iff either $y = x = F(x)$ or*

$$\exists v : f'_v(x) \neq 0 \wedge y = x + e_v f'_v(x_v),$$

where e_v is the v-th unit vector.

The behavior represented in such a state transition graph is non-deterministic. In a given state x there may be several $v \in V$ with $f'_v(x) \neq 0$, and therefore several y with $(x, y) \in T$.
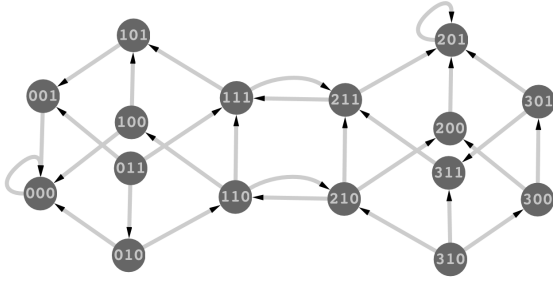
Figure 2: The state transition graph of the example parameter set.

2.2 Edge Constraints

Now, we consider information about the interaction type in the form of edge constraints. For example, interactions may be characterized as activating or inhibiting. Formally, edge constraints are additional edge labels that constrain the pool of feasible parameter sets for a regulatory network. Definitions for such edge constraints exist and are based on the observability or monotonicity of an interaction sign, usually either a '+' or a '-'. We use a more extensive set of edge constraints to allow for a more precise characterization of individual interactions. Similar ideas can be found in [4].

This more general form of edge-constraint is based on the observation that for a parameter set K, we can note for each interaction (w, v) if there is a regulatory context $R \subseteq V_-(v)$, such that adding w to R increases or decreases the value of K (as in [8]).

DEFINITION 2.8 (INCREASE AND DECREASE). *Given a parameter subset K of a regulatory network (V, E, t), we define the boolean propositions $+$ and $-$ on the set of edges $(w, v) \in E$ by*

$$+(w, v) := \exists R \subseteq V_-(v) : K_v(R) < K_v(R \cup \{w\})$$
$$-(w, v) := \exists R \subseteq V_-(v) : K_v(R) > K_v(R \cup \{w\})$$.

It has been remarked by Richard in the context of deriving global interaction graphs from dynamics (see [2]) that such a comparison of parameter values with and without a regulator w is too weak to guarantee an effect observable in the state transition graph. For stronger results a slightly more technical definition of increase and decrease could be introduced here. For boolean networks and for components v without self-regulation, i.e. $(v, v) \notin E$, the two definitions coincide.

For the parameter set of the running example, the values of $+$ and $-$ for each edge are the following:

$$+(1, 1) = 1, \quad +(3, 2) = 1, \quad +(1, 3) = 1,$$
$$+(2, 1) = 1, \quad +(1, 2) = 1, \quad -(1, 3) = 0,$$
$$-(1, 1) = 1, \quad -(3, 2) = 0,$$
$$-(2, 1) = 1, \quad -(1, 2) = 0.$$

Instead of $\neg+$ and $\neg-$ we write $\overline{+}$ and $\overline{-}$. Simple logical expressions of these propositions are used to select parameter sets, by defining the following constraints.

DEFINITION 2.9 (EDGE CONSTRAINTS).
A labeling function

$$s : E' \subseteq E \to \{+, \overline{+}, -, \overline{-}, + \wedge -, + \vee -, \overline{+} \wedge -, + \wedge \overline{-}\}$$

on a subset $E' \subseteq E$ of the edge set of a regulatory network (V, E, t) *is called edge constraint. A parameter set K satisfies the edge constraint s, if $s(w, v)$ is true for all $(w, v) \in E'$. In particular $\mathcal{K}(V, E, t, s)$ denotes all $K \in \mathcal{K}(V, E, t)$ that satisfy the edge constraint s.*

If an edge is not labeled by s, then no constraints are placed on the respective parameter values. The different labels can be interpreted as follows. $+$ and $-$ signify that an activating or inhibiting effect has been experimentally observed. It is not precluded that the respective opposite effect may also occur, depending on specific co-factors. In contrast, $+ \wedge \overline{-}$ and $\overline{+} \wedge -$ are used if the target is strictly activated or inhibited. $\overline{+}$ and $\overline{-}$ allow for the possibility that there is no interaction at all, but if so it is not activating respectively inhibiting. If the character of an interaction is not known or questionable but some effect is assumed, e.g., based on binding site properties, $+ \vee -$ is used. Finally, $+ \wedge -$ applies when the target is activated in some context and inhibited in another, reflecting the importance of co-factors.

Other logical combinations or types of edge constraints could be considered, for example, labeling the components by max or min:

$$max(v) := \exists R \subseteq V_-(v) : K_v(R) = d_+(v),$$
$$min(v) := \exists R \subseteq V_-(v) : K_v(R) = 0.$$

However, the increase and decrease edge constraints already allow for a detailed description of interactions and suffice to illustrate the underlying method.

The parameter set pool $\mathcal{K}(V, E, t, s)$ can be efficiently computed with a backtracking algorithm.
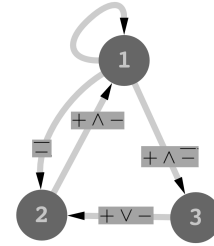
Figure 3: Edge constraints for example network, which lead to $|\mathcal{K}(V, E, t, s)| = 432$.

2.3 Model Checking

In this section, model checking is introduced as a means to analyze the state transition graph associated with a regulatory network. This has been proposed by various groups, see e.g. [8, 5, 7, 6, 4]. A Kripke structure or transition system is a state transition graph together with a labeling function that assigns atomic formulae to each node of the graph, which are defined to be true in this node. Computation Tree Logic (CTL) is a language that extends boolean propositions by temporal operators (see [15]). Boolean propositions can be evaluated at a node and so can CTL formulae. But, the temporal operators allow making statements about atoms that belong to other states, if there is a directed path in the transition graph from the first to the latter. Symbolic model checking is a fast method for finding the states in which a given CTL formula is true. We now will shortly review how

to label the states of a transition graph, define the syntax of CTL and describe the semantics of CTL formulae.

A state transition graph (X, T) can naturally be interpreted as a Kripke structure. Each state $x = (x_1, \ldots, x_n)$ has n labels of the form "$v_i \doteq x_i$". Here, we write "$\doteq$" to distinguish syntactic from semantic equality. This labeling is extended to make the formula constructions in Section 4 possible.

DEFINITION 2.10   (STATE TRANSITION SYSTEM).
Given a state transition graph (X, T) with variables $V := \{v_i \mid i \in [1, n]\}$, the set of atomic formulas consists of equalities

$$\mathcal{P} := \{ \sum_{1 \leq i \leq n} k_i v_i \doteq k \mid v_i \in V, k \in [-N, N], k_i \in \{-1, 0, 1\}\},$$

where $N := \sum_{v \in V} p(v)$. Then (X, T, L), with $L : X \to 2^{\mathcal{P}}$ and

$$L(x) = \{ \sum_{1 \leq i \leq n} k_i v_i \doteq k \mid k_i \in \{-1, 0, 1\}, k = \sum_{1 \leq i \leq n} k_i x_i\},$$

is the Kripke structure associated with the state transition graph (X, T).

A label $\sum_{1 \leq i \leq n} k_i v_i \doteq k$ captures simple expressions in the variables v_i that are true in state x. Model checking software like NuSMV (see [16]) can handle such expressions. The number N is included in the definition to emphasize that each node is only labeled with finitely many atoms. Here are a few atoms of the state $x = (0, 2, 11)$: $v_1 \doteq 0$, $v_1 + v_2 \doteq 2$, $-v_1 - v_2 + v_3 \doteq 9$.

The following definition of the syntax of CTL formulas is restricted to the temporal operators **EF** and **E**[**U**] that are needed for the method described here.

DEFINITION 2.11   (SYNTAX OF CTL FRAGMENT). *A CTL formula ϕ is defined inductively using the Backus Naur form. Let p be an element of the set of atomic formulas $\mathcal{P}$. Then*

$$\phi ::= p \mid \phi \wedge \phi \mid \boldsymbol{EF}\phi \mid \boldsymbol{E}[\phi \ \boldsymbol{U} \ \phi].$$

Given a Kripke structure (X, T, L), a state $x \in X$ and a CTL formula ϕ, the following rules determine whether ϕ is true in x.

DEFINITION 2.12   (SEMANTICS OF CTL).

- *An atomic formula $p \in \mathcal{P}$ is true in x, if p is a label of x, i.e., $p \in L(x)$.*

- *$\phi \wedge \phi'$ is true in x, if ϕ is true in x and ϕ' is true in x.*

- *$\boldsymbol{EF}\phi$ is true in x, if ϕ is true in x or if there is a path $(x, x^1, \ldots, x^n)$ in (X, T) with $n \geq 1$ and ϕ is true in x^n.*

- *$\boldsymbol{E}[\phi \ \boldsymbol{U} \ \phi']$ is true in x, if ϕ' is true in x or if there is a path $(x, x^1, \ldots, x^n)$ in (X, T) with $n \geq 1$ such that ϕ is true in x and x^i for $1 \leq i \leq n - 1$ and ϕ' is true in x^n.*

In the following sections, CTL formulas will be used to select parameter sets from given parameter pools. The selection is based on the existence of a state satisfying the formula.

DEFINITION 2.13   (ϕ-ACCEPTABLE PARAMETER SETS).
Given a CTL formula ϕ, the collection of parameter sets of a regulatory network (V, E, t) whose associated transition system contains a state in which ϕ is true is denoted by

$$\mathcal{K}(V, E, t, \phi) := \{K \mid K \in \mathcal{K}(V, E, t) \wedge \exists x \in X : \phi \text{ is true in } x\}.$$

Sometimes a transition system is said to satisfy a CTL formula ϕ, if ϕ is true in all states. Since we want to query the existence of paths starting in some state of the graph, the above definition is used.

3. DISCRETE TIME SERIES

A discrete time series for a regulatory network can be obtained by discretizing real-valued experimental data or by qualitative observations about regulatory components. The issue of choosing a suitable discretization method for experimental data is crucial (see e.g. [12]), but is not the subject of this article. Under the assumption that the regulation behaves switch-like regarding the regulator concentration, one ideally has to estimate the threshold below which the regulator is not effective and above which it becomes effective.

If estimation is not possible, statistical approaches can be used, for example mean clustering, scan-statistic or edge gradient methods as described by Shmulevich and Zhang in [11]. There is also a software implementation for the GNU project R called BoolNet by Muessel et al. [14] which automates such discretization. BoolNet is used in Section 6 to discretize the expression data of the IRMA network [10].

Including qualitative observations in the time series is a strength of discrete modeling as it may be hard to translate such assumptions into quantitative data required for continuous models.

Mathematically, a discrete time series is a matrix where rows are measurements and columns are observations for one component. Data points with questionable discretization results for certain components or observations known to be imprecise may be recorded as uncertain by the value -1. In practice this has the advantage of deriving results based on varying levels of certainty.

DEFINITION 3.1   (TIME SERIES). *A discrete time series with m measurements of n substances is a matrix $A \in N^{m \times n}$, where the entries of A are elements of $N := \mathbb{N} \cup \{-1\}$ and additionally*

$$\forall i \in [1, m] : \ \exists j \in [1, n] : \ a_{i,j} \geq 0.$$

The condition ensures that measurements without supportable entries are not included in the time series.

As a discrete time series for the running example, including 4 measurements and 3 imprecise observations, we choose

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 2 & -1 & 1 \\ -1 & 1 & 0 \\ 3 & 0 & -1 \end{pmatrix}.$$

A time series will be interpreted as encoding discrete paths. To define these paths, the partial state formulas, one for each measurement, are derived. The definition uses the set of indices whose variables are not equal to -1. Thus, uncertain variables will be excluded from the description of the paths.

DEFINITION 3.2 (PARTIAL STATES). *Given a time series $A \in N^{m \times n}$, the partial state formula of measurement $i \in [1, m]$ is*

$$\sigma_i := \bigwedge_{j \in M_i} (v_j \doteq a_{i,j}), \ \ where$$

$$M_i := \{j \in [1,n] \mid a_{i,j} \geq 0\}.$$

A partial state formula may be true in a set of states, depending on how many variables are uncertain. The paths encoded in a time series are then all paths that connect the partial states in the given order. A state transition graph that contains at least one such path is said to be able to reproduce the time series.

DEFINITION 3.3 (REPRODUCING A TIME SERIES). *A state transition graph (X, T) can reproduce a time series $A = (a_{i,j}) \in N^{m \times n}$ if there is a path $(x^1, \ldots, x^k)$ in (X, T) such that the index sequence $(1, \ldots, k)$ has a subsequence $(r_1, \ldots, r_m)$ satisfying for each $1 \leq i \leq m$ that σ_i is true in x^{r_i}.*

We say a parameter set can reproduce a time series, if this holds for the corresponding state transition graph.

The sequence of states $(x^1, \ldots, x^k)$ can be thought of as a simulation of the regulatory network from the initial state x^1. An intuitive CTL formula can be used to check if a parameter set can reproduce a time series. Such a formula is a nested sequence of partial state formulas connected via the predicates **EF** :

$$\sigma_1 \wedge \mathbf{EF}[\sigma_2 \wedge \mathbf{EF}[\ldots \sigma_{m-1} \wedge \mathbf{EF}[\sigma_m] \ldots]].$$

4. THE MONOTONE PATH FORMULAS

In this section, the paths encoded in a time series are characterized with regard to monotonicity in between successive measurements. The motivation for this is to take into account assumptions about the ratio of time elapsed between measurements on the one hand, and rates of change of components on the other. Intuitively, if for a substance the time elapsed between successive measurements is small compared to its rate of change, then we would expect its concentration to change monotonously, i.e., without oscillations.

To encode these ratios for each variable and at each measurement, we define a matrix to specify exactly which parts of the path should be monotone.

DEFINITION 4.1 (MONOTONICITY MATRIX). *Given a discrete time series $A \in N^{m \times n}$, a monotonicity matrix of A is any matrix $B = (b_{i,j}) \in \{0, 1\}^{m-1,n}$ such that*

$$\forall i, j : \ b_{i,j} = 1 \implies (a_{i,j} \geq 0 \wedge a_{i+1,j} \geq 0).$$

We say that variable j is specified to be monotone at measurement i, iff $b_{i,j} = 1$.

A time series and a monotonicity matrix define the following partially monotone paths. For technical reasons regarding the CTL construction in 4.5, we require that the path begins in a state representing the first and ends in one representing the last measurement.

DEFINITION 4.2 (*A*-*B*-MONOTONE PATHS). *Given a discrete time series $A \in N^{m \times n}$ together with a monotonicity matrix B, and a state transition graph (X, T), a*

path $(x^1, ..., x^r)$ in (X, T) is *A*-*B*-monotone, if there is a subsequence $(r_1, ..., r_m)$ of $(1, ..., r)$ with $r_1 = 1, r_m = r$ and the following two properties hold. First

$$0 \leq a_{i,j} \implies x_j^{r_i} = a_{i,j}.$$

Second, for the variables j specified to be monotone at measurement i

$$\forall t \in [r_i, r_{i+1} - 1] : \begin{cases} x_j^t \leq x_j^{t+1} : & if \ x_j^{r_i} \leq x_j^{r_{i+1}} \\ x_j^t \geq x_j^{t+1} : & if \ x_j^{r_i} > x_j^{r_{i+1}} \end{cases}.$$

A monotonicity matrix for the example time series A is

$$B = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

and an example of an *A*-*B*-monotone path is

$$((0, 1, 0), (1, 1, 0), (1, 1, 1), (2, 1, 1), (2, 1, 0), (3, 1, 0), (3, 0, 0)).$$

Again, a CTL formula is constructed to check the existence of an *A*-*B*-monotone path in a transition system. This formula is specifically designed for asynchronous transition graphs. It exploits the observation that for each couple of successive measurements, there is an expression $\sum k_i v_i$ in the marked monotone variables v_i that is increasing along any *A*-*B*-monotone path. To determine this expression we need to consider the variables that increase and decrease separately.

DEFINITION 4.3 (INDEX SETS). *Given a discrete time series $A \in N^{m \times n}$ and a monotonicity matrix $B \in \{0, 1\}^{m-1,n}$, we define for each $i \in [1, m-1]$ the index sets M_i^+ and M_i^- of increasing and decreasing variables respectively:*

$$\begin{aligned} M_i^+ &:= \{j \in [1, n] \mid b_{i,j} = 1 \wedge \ a_{i,j} \leq a_{i+1,j}\}, \\ M_i^- &:= \{j \in [1, n] \mid b_{i,j} = 1 \wedge \ a_{i,j} > a_{i+1,j}\}. \end{aligned}$$

Now we can construct the increasing expression mentioned before, define its initial value and by how much it has to increase in between measurements.

DEFINITION 4.4 (INCREASING EXPRESSION). *The increasing expression $V_i = V_i(v_1, \ldots, v_n)$, the initial value C_i and the distance d_i for $i \in [1, m-1]$ are defined to be*

$$\begin{aligned} V_i &:= \sum_{j \in M_i^+} v_j + \sum_{j \in M_i^-} (a_{i,j} - v_j), \\ C_i &:= \sum_{j \in M_i^+} a_{i,j}, \\ d_i &:= \sum_{j \in M_i^+ \cup M_i^-} |a_{i,j} - a_{i+1,j}|. \end{aligned}$$

In a state satisfying the partial state formula σ_i, the atomic formula $V_i \doteq C_i$ is true. The following *A*-*B*-monotone path formula asserts that $V_i \doteq C_i$ increases one by one until $V_i \doteq C_i + d_i$ and σ_{i+1} are true. To deal with the nested structure of the formula, it is defined recursively.

DEFINITION 4.5 (*A*-*B*-MONOTONE PATH FORMULA). *The A-B-monotone path formula $\phi_{A,B}$ for a time series $A \in N^{m \times n}$ and monotonicity matrix B is constructed recursively using the formulae $\rho_i, i \in [1, m]$. Let*

$$\rho_1 := \sigma_m,$$

and for $i \in [1, m-1]$

$$\rho_{i+1} := \begin{cases} \sigma_{m-i} \wedge \textbf{\textit{EF}}[\rho_i] & \text{if } M_{m-i}^+ \cup M_{m-i}^- = \emptyset \\ \sigma_{m-i} \wedge \gamma_{d_{m-i}+1}^{m-i} & \text{if } M_{m-i}^+ \cup M_{m-i}^- \neq \emptyset. \end{cases}$$

Here

$$\gamma_1^{m-i} := \textbf{\textit{E}}[(V_{m-i} \doteq C_{m-i} + d_{m-i}) \ \textbf{\textit{U}} \ \rho_i]$$

and if $d_{m-i} \geq 1$ then

$$\gamma_{t+1}^{m-i} := \textbf{\textit{E}}[(V_{m-i} \doteq C_{m-i} + d_{m-i} - t) \ \textbf{\textit{U}} \ \gamma_t^{m-i}]$$

for $t \in [1, d_{m-i}]$. Finally, define $\phi_{A,B} := \rho_m$.

From the above definition, a pseudo code algorithm for the construction of an A-B-monotone path formula is derived:

```
ρ₁ := σₘ
for i = 1 to m − 1 do
    if M⁺ₘ₋ᵢ ∪ M⁻ₘ₋ᵢ = ∅ then
        ρᵢ₊₁ := σₘ₋ᵢ ∧ EF[ρᵢ]
    else
        γ₁ᵐ⁻ⁱ := E[(Vₘ₋ᵢ ≐ Cₘ₋ᵢ + dₘ₋ᵢ) U ρᵢ]
        if dₘ₋ᵢ ≥ 1 then
            for t = 1 to dₘ₋ᵢ do
                γₜ₊₁ᵐ⁻ⁱ := E[(Vₘ₋ᵢ ≐ Cₘ₋ᵢ + dₘ₋ᵢ − t) U γₜᵐ⁻ⁱ]
            end for
        end if
        ρᵢ₊₁ := σₘ₋ᵢ ∧ γᵐ⁻ⁱₐₘ₋ᵢ₊₁
    end if
end for
```

Next we show that this formula characterizes the existence of an A-B-monotone path.

THEOREM 4.6 (CORRECTNESS). *Given an asynchronous state transition graph* (X, T), *its associated state transition system* (X, T, L) *and a discrete time series* $A \in N^{m \times n}$ *together with a monotonicity matrix* $B \in \{0,1\}^{m-1,n}$, *the A-B-monotone path formula is true in* (X, T, L) *if and only if there is an A-B-monotone path in* (X, T).

PROOF. By the recursive structure of ρ_m it is sufficient to consider a matrix A with just two rows. For further simplicity assume there are only increasing variables $(M_1^- = \emptyset)$. The mixed case follows the same reasoning, because every $j \in M_1^-$ appears as $v_j' := x_j^1 - v_j$ in V_1 and v_j' increases, if v_j decreases.

First, we want to show that the existence of an A-B-monotone path $(x^1, ..., x^r)$ in (X, T) implies that ρ_2 is true in x^1. For each $t \in [1, r-1]$ we have $V_1(x^t) \leq V_1(x^{t+1})$, because V_1 is the sum of variables that increase along that path. The difference $V_1(x^{t+1}) - V_1(x^t)$ is at most 1 since T contains only unitary asynchronous transitions. So there must be a partition of $[1, r]$ into $d_1 + 1$ intervals, where $d_1 := \sum_{j \in M_1^+}(x_j^r - x_j^1)$, such that V_1 is constant on each interval and increases by 1 from one interval to the next. On all states x of the first interval the formula $V_1(x) \doteq C_1$ is true and on all states x of the last interval the formula $V_1(x) \doteq C_1 + d_1$ is true. Therefore γ_t^1 for $t \in [1, d_1 + 1]$ is true on the t-th interval, counted from right to left and hence ρ_2 is true in x^1.

Second, we want to show that ρ_2 is true in $x \in X$ implies that there is an A-B-monotone path in (X, T). Since ρ_2 is true in $x^1 := x$ there is a path $(x^1, ..., x^r)$ in (X, T) such

that σ_1 is true in x^1 and $\rho_1 \doteq \sigma_2$ is true in x^r, which is the first property of an A-B-monotone path. Furthermore, $[1, r]$ can be partitioned into $d_1 + 1$ intervals such that γ_t^1 is true in the t-th interval counted from right to left. Therefore V_1 increases by 1 from one interval to the next. Since T contains only unitary asynchronous transitions, there is exactly one variable $j \in M_1^+$ that increases by 1 from one interval to the next. Therefore $x_j^k \leq x_j^{k+1}$ for all $k \in [1, r-1]$ and $j \in M_1^+$ which is the second property of an A-B-monotone path. So the path $(x^1, ..., x^r)$ is A-B-monotone. $\square$

5. WORKFLOW

In this section, we introduce a methodology to analyze compatibility of a regulatory network and a given time series. After describing a possible workflow, we conclude the section with remarks regarding computation methods and costs. The procedures are illustrated in the next section.

Let us consider a regulatory network, possibly including edge constraints, a time series and a monotonicity matrix (consisting only of zero entries in case no monotonicity assumptions are made). As a first step, we check whether there are parameter sets that reproduce the time series, i.e., we compute the parameter pool $\mathcal{K}(V, E, t, s, \phi_{A,B})$. If the model checking procedure returns a unique parameter set, we can proceed with the analysis of the model. However, this case will only occur very rarely. More commonly, the procedure either returns a large pool of parameter sets or no set at all. In the following, we look at both cases more closely

5.1 Characterizing Model Pools

If the parameter pool contains many parameter sets, the information encoded in the network and the time series was not sufficient to determine a unique specified model. One possibility to deal with this difficulty is to choose a model from the pool using meaningful criteria, e.g., some notion of minimality. A different approach is to characterize the parameter pool in order to derive information about the system strongly supported by the integrated data. We propose ideas in line with the second approach. One characteristic of a model pool are parameter values that are identical across all parameter sets. Such values may allow for new insights into how a component behaves under the influence of several regulators, clarifying synergies and redundancies in the network.

DEFINITION 5.1 (DETERMINED PARAMETER VALUES). *Given a parameter pool* $\mathcal{K}$, *the value of a component v in a regulatory context* $R \subseteq V_-(v)$ *is determined if there is a* $p \in [0, p(v)]$ *such that*

$$\forall K \in \mathcal{K} : K_v(R) = p.$$

This idea can be extended to finding the range of values for each component and regulatory context.

Even if the parameters for a given component are not completely determined, we can still try to extract further information. To get an idea about the different behaviors that a component can have in a parameter pool, we count the v-local parameter sets in $\mathcal{K}$.

DEFINITION 5.2 (BEHAVIORS). *Given a parameter pool* $\mathcal{K}$ *of a regulatory network* (V, E, t), *the behaviors* $\mathcal{K}_v$ *of component* $v \in V$ *are the set of v-local parameter sets in* $\mathcal{K}$,

$$\mathcal{K}_v := \{K_v \mid K \in \mathcal{K}\}.$$

This information can be used to study how components are tuned to work together in reproducing a time series. If any combination of component behaviors is a parameter set in the pool, then the components are said to be independent.

DEFINITION 5.3 (INDEPENDENCE). *A parameter pool $\mathcal{K}$ consists of independent components, if*

$$\prod_{v \in V} |\mathcal{K}_v| = |\mathcal{K}|.$$

Further characterization of the parameter pool could study which behaviors do not appear together and try to identify components and regulatory contexts, which, if determined, would lead to the steepest reduction in feasible parameter sets. Identifying such contexts could be used to design experiments that reduce the number of feasible parameter sets in the fastest possible way.

Characterization of the parameter pool can also focus on the edge labels. They can be arranged into a logical implication hierarchy. For example. "$+ \wedge \overline{-} \implies \overline{-}$" and we thus place $+ \wedge \overline{-}$ above $\overline{-}$ in the hierarchy diagram in Fig. 4. For each unlabeled edge of the regulatory network and edges carrying one of the constraints that may be strengthened $(+,-,+ \vee -,\overline{-},\overline{+})$, we determine the strictest label that is true for all parameter sets. This may lead to determining an effect of a regulator on its target that was formerly not known. An edge may for example be included in a network, because the source component is known to bind to the target component's promoter, but without any knowledge of the effect this binding has (i.e., with label $+ \vee -$). With a time series this label may be sharpened to $+$ and thus hypothesize an activation.
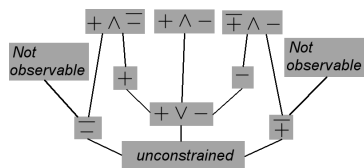

Figure 4: Hierarchy of edge constraints with stricter labels above weaker ones. "Not observable" is used to emphasize the meaning of $\overline{+} \wedge \overline{-}$.

5.2 Evaluating the Time Series

An ideal sampling frequency would result in a discrete time series capturing all value changes of the components, but usually data points are rather sparse. In order to understand the underlying system, we need to know whether the sampling was sufficient to capture its essential behavior. Here, we focus on determining potential oscillatory behavior not inferable from the time series due to coarse sampling.

Consider a network, a time series and a monotonicity matrix that are compatible, i.e., the corresponding parameter pool is not empty. We start with the assumption that the time series is sufficient to exclude the possibility of undetected oscillatory behavior. Intuitively, if sufficiently many measurements were made, it can be assumed that all variables are monotone at all measurements.

DEFINITION 5.4 (BEST FIT). *Given a regulatory network (V, E, t), a time series A and the monotonicity matrix*

B, where

$$b_{i,j} = \begin{cases} 1: & 0 \leq a_{i,j}, a_{i+1,j} \\ 0: & else \end{cases},$$

a parameter set that satisfies the A-B-monotone path formula is called a best fit of (V, E, t) to A.

Recall that the entries or positions (i, j) of B represent the value transition of the j-th component from measurement i to measurement $(i+1)$, and that the entry 1 signifies a monotone value change. If no best fits of (V, E, t) to A exist, we can be sure that there is a set of positions of B, such that all parameter sets in the considered pool produce at least one unobserved oscillation in one of the positions. In these positions the temporal resolution of A is too coarse to capture the behavior of the network. A trivial such set is the set of all positions, but there may be a smaller set, ideally with only a single position. Starting with the originally considered monotonicity matrix B, a heuristic approach to finding a non-trivial set is to introduce additional monotonicity constraints position by position. If such an added constraint does not result in a reduction of the parameter pool, we discard the corresponding position, since all models agree with the assumed monotonicity for that position, and we need no extra sampling between the corresponding data points. We introduce a measure for the impact of an additional monotonicity constraint as follows.

DEFINITION 5.5 (SELECTIVITY). *Given a regulatory network (V, E, t), a time series A and a monotonicity matrix B, we define for each $1 \leq i \leq m, 1 \leq j \leq n$ such that*

$$b_{i,j} = 0 \text{ and } 0 \leq a_{i,j}, a_{i+1,j},$$

the monotonicity matrix B' by

$$b'_{i',j'} := \begin{cases} 1: & i' = i, j' = j \\ b_{i,j}: & else \end{cases}$$

and the selectivity of position (i, j) by

$$S(i,j) := 1 - \frac{|\mathcal{K}(V, E, t, \phi_{A,B'})|}{|\mathcal{K}(V, E, t, \phi_{A,B})|}.$$

All positions that have selectivity 1 hypothesize obligatory oscillations of component j in between measurements i and $i + 1$, which indicates the need for additional data points between the measurements. If no such positions exist, we choose the set $\{(i, j) \mid S(i, j) > 0\}$ as places of interest for new measurements.

5.3 Reviewing Structure and Data

So far we have considered the case that we have no contradictions in our modeling assumptions and data, resulting in viable choices of parameter sets. If a network is not compatible with a time series and the possibly additionally provided monotonicity matrix, i.e., the corresponding parameter pool is empty, there are two possible lines of investigation, depending on whether the correctness of the network structure or of the data is questioned. In both cases, the idea is to check what minimal changes can lead to compatibility.

Regarding the structure, we may, in a first step, relax the constraints on the interactions and instead label every edge with the observability label $+ \vee -$. Thus we include no assumptions on the character of an interaction, but only

require it to be observable. We now test if the weakened assumptions result in a non-empty parameter pool.

DEFINITION 5.6 (STRUCTURAL COMPATIBILITY). *A regulatory network (V, E, t) is structurally compatible with a time series A and monotonicity matrix B, if for the labeling function*

$$s : E \to \{+ \vee -\},$$

there is a parameter set $K \in \mathcal{K}(V, E, t, s, \phi_{A,B})$ that can reproduce the time series.

If the network is structurally compatible, we know that the contradiction must be caused by the assumed edge constraints and so a review of the edge labels (see 5.1) of the compatibility pool should reveal that there is an interaction whose label has changed. If it is not, we may go further and introduce additional edges or remove existing ones.

Regarding the data, we proceed similarly by first lifting monotonicity constraints in B (if there are any) and then replacing particular values in A with the imprecise -1.

5.4 Computation and Scalability

The computational steps in the workflow are (1) to exhaustively generate all parameter sets satisfying the edge constraints (not the whole parameter space), (2) to translate a parameter set into a model checker input file, and (3) to pass it to a model checker, together with the A-B-monotone path formula. For model checking we use NuSMV ([16], see also [8] and [9]). For computation of the parameter sets, we apply a backtracking algorithm with failure on constraint violation.

Regarding scalability and computation times, we first note that the state space is exponential in the number of components, which places a strong limit on the possible number. Second, we compute a large part of the parameter space, depending on how restricting the edge labels are. Efficient algorithms considering partial parametrizations only have been introduced for PADE models (see [9]). Similar approaches would be desirable for the Thomas formalism.

As standing, analysis is limited to structures of about 30,000 states, e.g. 15 binary components or 9 ternary components. For such models the time per model check is impacted considerably by the nesting depth of a given CTL formula, which in our case increases linearly with the length of a time series and monotonicity constraints. Model checking a 30,000 state model and a time series of 5 measurements takes about 1 second on a 2.27GHz Laptop.

Given these restrictions imposed by the time per model check, the computation time for the parameter sets is negligible. However, it should be noted that even with the most restrictive edge labels ($+ \wedge \overline{-}$ and $- \wedge \overline{+}$) on edges targeting a binary component, there are already 6,894 local parameter sets for only 5 regulators. For a ternary component, the number of such regulators is limited to 4, resulting in 7,008 local parameter sets.

6. APPLICATION: THE IRMA NETWORK

We apply the workflow of the previous section to a biological network called IRMA, for which several time series are available. A corresponding search for consistent parameters of a qualitative PADE model is described by Batt et al. [9].

The IRMA regulatory network consists of 5 genes with gene control and protein-protein interactions, which has been inserted into the genome of Saccharomyces cerevisiae (see Cantone 2009 [10]). Several populations of this genet-
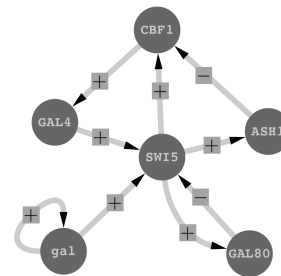
Figure 5: The IRMA regulatory network.

ically modified yeast were grown and subjected to perturbations by adding or removing galactose from the growth medium. Altogether 11 real-valued time series are available: 5 repetitions of the switch-on perturbation (adding galactose) and 4 repetitions of the switch-off perturbation (removing galactose) plus two averaged time series for each category.

A comprehensive analysis would include all available time series. Since we aim for a clear illustration of our approach, we restrict analysis to the averaged switch-off time series. In addition, we only consider a boolean model.

We binarized the expression data for the galactose removal experiment using the scan-statistic method described in [14]. Additionally, we added values for *gal* based on qualitative observations. The first entry of its profile is left uncertain, because although the cells were washed, we are not sure if galactose was still present in the cytoplasm or not. This resulted in the discrete time series

$$A = \begin{pmatrix} CBF1 & ASH1 & GAL4 & GAL80 & SWI5 & gal \\ 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Matching the binarized data, we chose a boolean representation, i.e., $p(v) = 1$ for every variable v. The state space is then

$$X = [0, 1]^6 \text{ with } |X| = 64, \text{ and}$$

$$|\mathcal{K}(V, E, t)| = 2^2 \cdot 2^2 \cdot 2^4 \cdot 2^2 \cdot 2^2 \cdot 2^8 = 2^{20} = 1.048.576.$$

The network edges and edge-constraints were adopted from [10] and interpreted as '+' and '−', i.e., as observable activations or inhibitions. We then computed all parameter sets that satisfy the edge-constraints and reproduce the time series without any monotonicity assumptions:

$$|\mathcal{K}(V, E, t, s)| = 404 \text{ and } |\mathcal{K}(V, E, t, s, \phi_{A,0})| = 73.$$

We proceeded by characterizing the parameter pool $\mathcal{K}(V, E, t, s, \phi_{A,0})$. All parameters of components with a unique regulator, namely $GAL4$, $ASH1$ and $GAL80$, coincide for all parameters sets, i.e., the component behavior is completely determined. The labels of edges targeting these components can be strengthened to $+ \wedge \overline{-}$, i.e., they are recognized as non-ambiguous activating influences. For $SWI5$ one parameter is determined: $K_{SWI5}(\{GAL4\}) = 1$, suggesting that $GAL4$ alone is sufficient to activate $SWI5$, as opposed to galactose which may require $GAL4$ for up-regulation of $SWI5$ as the parameter $K_{SWI5}(\{gal\})$ is in the range $[0, 1]$.

Regarding the behaviors of $CBF1$ and $SWI5$ as defined in Sect. 5.1, there are 4 for the latter and 33 for the former. The set $\mathcal{K}(V, E, t, s, \phi_{A,0})$ is not independent, since $4 \cdot 33 = 132$, but there are only 73 sets in the pool. Therefore, not every behavior of $SWI5$ is compatible with every behavior of $CBF1$. Identification of conflicting behaviors can then be utilized for experimental design. Development of strategies that allow to identify a component and corresponding behavior whose parameter determination would result in a maximal decrease of the parameter pool is an issue for future work.

Continuing in the workflow, we assessed the quality of the time series. There are no best fits of the IRMA network to the time series, but computing the selectivity of positions (i, j) in A we found 8 positions to have a selectivity of 1 and hypothesize the following oscillations.

Name	Begins oscillation at measurement
$CBF1$	1,8,11
$SWI5$	5,7,8,11,12

The real-valued expression profiles show that $SWI5$ does indeed oscillate, but that the oscillations are below the threshold that the binarization method computed. In this particular case, the result emphasizes the need of revising the chosen threshold. However, it also illustrates nicely the potential of our method to evaluate sufficiency of measurements, since similar results would be obtained if the data points between 5 and 15 in the $SWI5$ plot were simply missing. Based on our analysis the importance of providing additional measurements for that time span would be highlighted.

For $CBF1$ the expression curve shows a decline with two steady intervals around measurements 10 and 15. Here, the real-valued data shows no oscillation, but rather different plateaus. Our results point out the time points where changes of activity levels result in qualitatively observable effects, and thus indicate the need for a finer representation of activity levels than a simple boolean view. Investigating the relation between the predictions for oscillations generated by our method and the need for an expanded component value range will be an objective of future work.

Since the boolean model for the IRMA network can reproduce the chosen time series, we imposed additional assumptions to illustrate the workflow in case of inconsistencies (Sect. 5.3). We considered that value changes in $GAL80$ involve transcription processes. Let us assume that the transcription of $GAL80$ is slow, so that it is not expected to significantly change concentration within the sampling rate of 10 minutes, i.e., there will be no oscillations between sampling points.

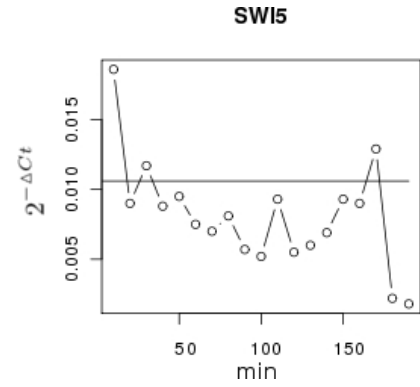The entries of a monotonicity matrix B encoding this as-

Figure 6: Real-valued expression profile of $SWI5$ where the horizontal line is the binarization threshold obtained by the scan-statistic method.
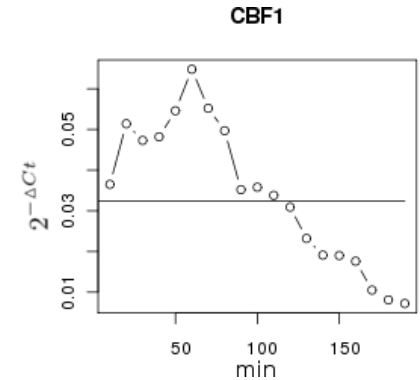

Figure 7: Real-valued expression profile of $CBF1$ where the horizontal line is the binarization threshold obtained by the scan-statistic method.

sumption are 1 in the column corresponding to the $GAL80$ expression profile. We set all remaining entries of B to zero, imposing no further monotonicity constraints. The corresponding parameter pool $\mathcal{K}(V, E, t, s, \phi_{A,B})$ is empty. We decided to proceed by revising the structure of the internal components, taking the activating effect of gal on $SWI5$ as given. The IRMA network is structurally compatible with A and B. We now try to derive valid information from the resulting parameter pool. Of the 12,960 parameter sets in the pool where all internal edges of the network, i.e. not (gal, gal) and $(gal, SWI5)$, are relaxed to $+ \vee -$, 144 satisfy $\phi_{A,B}$. Interestingly there are no determined parameter values in this pool, but two interactions are stricter than assumed in every parameter set:

$$s(ASH1, CBF1) = -, \quad s(SWI5, CBF1) = +.$$

This illustrates how we can recover information from the parameter pool supported by the available data. In summary, we can observe that the reasonable assumption that the switch-off series has captured all oscillations of $GAL80$ validates the original labels targeting $CBF1$.

7. CONCLUSION

In this paper we study the compatibility of a model of a regulatory network and its observed behavior in the form of a discretized time series. On the formal level, we slightly extend the usual edge labels (e.g. [8]) with boolean propositions on edges (similar to [4]) and introduce time series that may be partially exact or monotone. On the methodological level, a workflow is suggested that branches in places where given assumptions may or may not be satisfied.

In contrast to related work, we also use our methods to assess the quality of the considered time series. In case of consistency of the network structure and the time series, we investigate the temporal resolution of the time series by defining a best fit. For such parameter sets additional measurements would not reveal much further information, because in between measurements all variables approach their target activities without oscillating. However, we show that if no best fits exist, oscillations can be predicted for particular variables in particular time intervals. We have shown the potential of this approach using the IRMA network. In addition, the results hint at the possibility of using the same methods to assess the discretization threshold of individual components, as well as the number of thresholds used for a component. This will be further elucidated in future work.

While we obtain satisfactory results for networks of small and medium size, we certainly have to increase the computation efficiency to tackle larger models. Future research will focus on development of more powerful implementations of our ideas.

8. REFERENCES

[1] R. Thomas, 1991. Regulatory networks seen as asynchronous automata: a logical description. J. Theor. Biol. 153(1), p.1-23.

[2] A. Richard, 2010. Negative circuits and sustained oscillations in asynchronous automata networks. Advances in Applied Mathematics 44, p.378-392.

[3] C. Chaouiya, E. Remy, B. Mossé, D. Thieffry, 2003. Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. Springer-Verlag, LNCIS 294 : p.119-126.

[4] F. Corblin,E. Fanchon, L. Trilling, 2010. Applications of a formal approach to decipher discrete genetic networks. BMC Bioinformatics (11) 1 p.385.

[5] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, V. Schächter, 2004. Modeling and querying biomolecular interaction networks. Theor. Comput. Sci. (325) 1 p.25-44.

[6] P.T. Monteiro, D. Ropers, R. Mateescu, A.T. Freitas, H. de Jong, 2008. Temporal logic patterns for querying dynamic models of cellular interaction networks. Bioinformatics. (24) 16. p.i227-i223.

[7] J. Fromentin, J.-P. Comet, P. Le Gall, O. Roux, 2007. Analysing Gene Regulatory Networks by both Constraint Programming and Model-Checking. EMBC07, 29th IEEE EMBS Annual International Conference, August p.23-26, p.4595-4598.

[8] G. Bernot, J-P. Comet, A. Richard, J. Guespin, 2004. Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. J. Theor. Biol. 229(3), 339-347.

[9] G. Batt, M. Page, I. Cantone, G. Goessler, P. Monteiro, H. de Jong, 2010. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking.. Bioinformatics 26 (18): i603-i610.

[10] I. Cantone, L. Marucci, F. Iorio, M. A. Ricci, et al., 2009. A yeast synthetic network for in vivo assessment of reverse-engeneering and modeling approaches. Cell 137, p.172-181.

[11] I. Shmulevich, W. Zhang, 2002. Binary analysis and optimization-based normalization of gene expression data. Bioinformatics, Vol.18 no.4 p.555-565.

[12] E.S. Dimitrova, J.J. McGee, R.C. Laubenbacher, 2005. Discretization of Time Series Data. Journal of computational biology a journal of computational molecular cell biology 17, p.853-868.

[13] E. Dimitrova, L.D. Garcia-Puente, F. Hinkelmann, A.S. Jarrah, R. Laubenbacher, B. Stigler, M. Stillman, P. Vera-Licona, 2009. Parameter estimation for Boolean models of biological networks. Theoretical Computer Science, p.1-19.

[14] C. Muessel, M. Hopfensitz, D. Zhou, H. Kestler, 2010. R package 'BoolNet'. CRAN repository: http://cran.r-project.org/web/packages/BoolNet.

[15] M. Huth, M. Ryan, 2000. Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, Cambridge.

[16] A. Ciamatti, E.M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pisotre, M. Roveri, R. Sebastiani, A. Tacchella, 2002. NuSMV2: An OpenSource Tool for Symbolic Model Checking. In Proceedings of International Conference on Computer-Aided Verification (CAV 2002). Copenhagen Denmark, p.27-31.