Integrating time-series data in large-scale discrete cell-based models

Louis Fippo Fitime[1], Christian Schuster[2], Peter Angel[2], Olivier Roux[1], and Carito Guziolowski[1]

[1] LUNAM Université, École Centrale de Nantes, IRCCyN UMR CNRS 6597
(Institut de Recherche en Communications et Cybernétique de Nantes)
1 rue de la Noë – B.P. 92101 – 44321 Nantes Cedex 3, France.
louis.fippo-fitime@irccyn.ec-nantes.fr
http://www.irccyn.ec-nantes.fr/en/
[2] Division of Signal Transduction and Growth Control (A100), DKFZ-ZMBH Alliance,
Deutsches Krebsforschungszentrum, Heidelberg, Germany

Abstract. In this work we propose an automatic way of generating and verifying formal hybrid models of signaling and transcriptional events, gathered in large-scale regulatory networks.This is done by integrating temporal and stochastic aspects of the expression of some biological components. The hybrid approach lies in the fact that measurements take into account both times of lengthening phases and discrete switches between them. The model proposed is based on a real case study of keratinocytes differentiation, in which gene time-series data was generated upon Calcium stimulation.
To achieve this we rely on the Process Hitting (PH) formalism that was designed to consider large-scale system analysis. We first propose an automatic way of detecting and translating biological motifs from the Pathway Interaction Database to the PH formalism. Then, we propose a way of estimating temporal and stochastic parameters from time-series expression data of action on the PH. Simulations emphasize the interest of synchronizing concurrent events.

Keywords: time-series data, large-scale network, hybrid models, compositional approach, stochastic simulation

1 Introduction

Unraveling and describing the mechanisms involved in the regulation of a cell-based biological system is a fundamental issue. These mechanisms can be modeled as biological regulatory networks, whose analysis requires to preliminary build a mathematical or computational model. By just considering qualitative regulatory effects between components, biological regulatory networks depict fairly well biological systems, and can be built upon public repositories such as the Pathways Interaction Database [23] and hiPathDB [30] for human regulatory knowledge. In this work we built a hybrid model of signaling and transcriptional events, gathered in large-scale regulatory networks, for which stochastic simulation parameters were inferred from gene expression time-series data.

High-throughput experimental data has been used since more than one decade ago to infer biological regulatory models. A variety of methods were proposed to infer dynamic or static models of protein signaling or gene regulations depending on the nature of the experimental data. We can cite methods that infer static gene regulatory models from steady-state gene expression datasets of over-expression or knock-down perturbations using statistical models generating small-scale (ten species) models [8] or middle-scale (maximum 100 species) models [19]. Additionally, we can cite methods that recovered gene regulatory dynamic models from time-series data using kinetic modeling [5,20] generating small-scale models. Recently, static boolean models for middle and large-scale (over 100 species) signaling protein networks have been derived from a *prior* network and fitted to steady-state multiple perturbation phosphoproteomics data [10,16] using combinatorial optimization through logic and integer linear programing to explore the vast search space of candidate boolean models. When using time-series multi-perturbation phosphoproteomics data, results can be extended to reconstruct middle-scale dynamic signaling models via the use of stochastic search approaches [14] that do not guarantee an exhaustive exploration of the search space of candidate models. The approach presented in this work confronts a prior signaling and gene regulatory large-scale network, obtained from publicly curated databases, to time-series gene expression data, by using discrete automaton models and stochastic simulations. Our built model verifies the agreement of expression traces over time given a signed (activations/inhibitions), directed and cyclic prior graph.

The advantages and complementariness of our method with respect to the afore cited approaches are that it allows us to define a logic that integrates signaling and transcription events (imposing different regulatory rules on these events), it also integrates multi-valued states of the system components, and importantly it deals with the complexity of large-scale dynamic models.

Several conceptually different approaches are available for modeling Biological Regulatory Network (BRN) dynamics. The most common approach is ordinary differential equations (ODE) ¡¡¡¡¡¡¡ HEAD that describe deterministic (population average) behavior in a continuous manner. Even for simple model including a simple interaction between two components, the analytical solution is impossible. Thus we must refer to simulation as the only practical method. Furthermore, continuous models require quantitative knowledge in terms of kinetic coefficients, which are unknown and very difficult to measure. Thereby, various abstraction approaches have been developed to make BRN models more convenient for analysis. Synchronous Boolean model was first proposed by Kauffman [12] and an alternative asynchronous model was proposed by Thomas [27]. Following these two papers, many other models have been proposed [25,26,7,6] for modeling dynamic of BRN. All of these models are purely qualitative and discrete, thus do not incorporate quantitative time or other quantities. As well, discrete models have been extended to integrate quantitative aspects. Time aspect have been introduced by [24,4,1,29]. It relies on timed automaton implemented. Anyways, this models do not take into account the stochastic aspects of the influences of BRN. ======= that describes deterministic (population average) behaviour in a continuous maner. Even for simple models including a simple interaction between two components, the analytical solution is impossible. Thus we must refer to simulation as the only practical

method. Futhermore, continuous models require quantitative knowledge in terms of kinetic coefficients, which are unknown and very difficult to measure. Thereby, various abstraction approches have been developed to make BRN models more convenient for analyses. The synchronous Boolean model was first proposed by Kauffman [12] and an alternative asynchronous model was proposed by Thomas [27]. Folowing these two papers, many other models have been proposed [25,26,7,6] for modeling the dynamic of a BRN. All of these models are purely qualitative and discrete, thus do not incorporate quantitative time or other quantities. Therefore, discrete models have been extended to integrate quantitative aspects, such as time, which was introduced by [24,4,1,29]. These approaches rely on timed automata implementations. These models, however, do not take into acount the stochastic aspects of the influences of a BRN. ¿¿¿¿¿¿¿ 12c241a1d88dedd17eca66bd84aa7ba99f981d1c

In the context of modeling and analyzing stochastic and concurrent biological systems various formalisms have been introduced such as Stochastic Petri Nets which is suitable for the representation of parallel systems [17]. They have been successfully applied in many areas; in particular, the specification of Petri Nets allows an accurate modeling of a wide range of systems including biological systems [11]. The major problem of Stochastic Petri Nets is that, generally, they do not lead to compact models. In addition, they do not provide results to deal with the state space explosion and are thus computationally expensive when modeling large-scale biological networks. The Stochastic pi-calculus formalism was introduced by [21] and used in [15] for the modeling of biological systems. Stochastic pi-calculus has a rich expressiveness and is well adapted for the use of compositional approach. In this work we rely on this formalism through the Process Hitting (PH) framework [18], since it is especially useful for studying systems composed of biochemical interactions, and provides stochastic simulation as well as efficient algorithms, based on the verification of state reachability, to study dynamical properties of the system. The PH framework uses qualitative and discrete information of the system without requiring enormous parameter estimation tasks for its stochastic simulation. This framework has been previously used to verify dynamical properties on biological systems without integrating high-throughput experimental data.

In this work we provide a method to build a time-series data integrated PH model and we evaluate the prediction power of this model concerning the simultaneously predicted traces of 12 gene components of the system upon system stimulation. The main results of this work are: (1) automatic generation of PH models integrating gene transcription and signaling events, with and without synchronization of concurrent events, from the Pathways Interaction Database, (2) parameter estimation from time-series data and parameter integration in the PH model, and (3) comparison of the PH model predictions and experimental results. To illustrate our approach, we used a time-series dataset of human keratinocytes cells, which shows the fluctuations of mRNA expression across time upon Calcium stimulation. This dataset was built to study keratinocytes differentiation, a time-dependent process in which the sequence of activation of signaling proteins is not yet completely understood. The method proposed in this paper remains general and can be applied to other case-studies.

2 Data and Methods

The general work-flow for integrating time-series data in a PH model is depicted in Fig. 1, in the following sections we detail some of the work-flow steps.

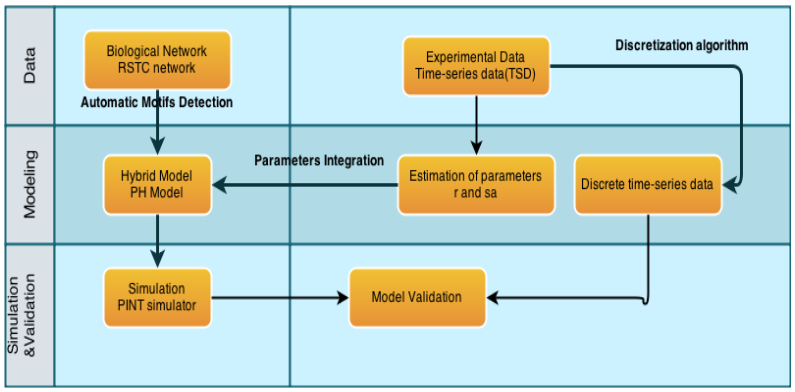


Fig. 1. Integrating stochastic and temporal information in a large-scale discrete biological model. These parameters are rate (r) and the stochasticity absorption factor (sa) which will be presented later in 2.3.

2.1 Data

Interaction graph. The interactions of the biological system under study were represented in an RSTC network, which stands for multi-layer Receptor-Signaling-Transcription-Cell state network and that was generated from the Pathway Interaction Database (PID). In order to build this network one needs to select a set of seed nodes related to the biological process studied. For our case study, the seed nodes were: (1) *E-cadherin*, which is a protein having Calcium binding domains and which plays an important role in cell adhesion; (2) the 12 significantly differentially expressed genes accross the 10 time-points; and (3) the cell states of keratinocytes-differentiation and cell-cycle-arrest. The network was extracted automatically from the whole content of the PID database by using a subgraph algorithm to link the seed nodes [9]. In Fig. 2 we show the RSTC network obtained.

Definition 1 (RSTC Network). *A* RSTC Network N *is a couple* (V, E)*, where:*

- $V = V_T \bigcup V_I$ *is the finite set of* nodes*; with* $V_T = \{v_{1t}, v_{2t}, \ldots, v_{n1t}\}$ *the set of terminal nodes;* $V_I = \{v_{1i}, v_{2i}, \ldots, v_{n2i}\}$ *the set of transient nodes.*
- $E = \{e_1, e_2, \ldots, e_m\}$ *is the set of edges.* $E \subseteq (V_T \times V_T) \bigcup (V_T \times V_I) \bigcup (V_I \times V_T)$

In this definition, terminal nodes can be either mRNA expression or proteins or complexes or cellular states or biological processes or positive conditions. On the other side, transient nodes can be either transcriptions or translocations or modifications or compounds. Edges are of different types: activation (agent), inhibition, output, input and protein-family-member.

Definition 2 (Pattern).
A pattern can be defined as an atomic set of biological components and thier interacting roles.

¡¡¡¡¡¡¡ HEAD The first column of table 1 shows some examples of patterns that can be found in a RSTC Network. ======= The first column of TAble 1 shows us some examples of patterns that we can found in a RSTC Network. ¿¿¿¿¿¿¿ 12c241a1d88dedd17eca66bd84aa7ba99f981d1c

Time-series microarray dataset. We use the time-series microarray data from Calcium stimulated human keratinocyte cells measured at 10 time-points (1h, 2h, 3h, 4h, 5h, 6h, 8h, 12h, 18h, 24h). The expression levels were measured in log_2; the expression of a gene at an specific time point is compared with respect to a control condition (gene expression in a kerationocyte cell without Calcium stimulation). We selected genes, which mRNA expression e was significantly ($log_2(e) \geq 1$) up-regulated or significantly ($log_2(e) \leq -1.0$) down-regulated in at least one time point compared to control. From this procedure 200 mRNA expression transcripts were selected. We included in our model a subset of 12 of the 200 selected (see Fig. 3) because these 12 genes had upstream regulatory mechanisms when querying the PID database and therefore were connected in the interaction graph to the E-cadherin node.

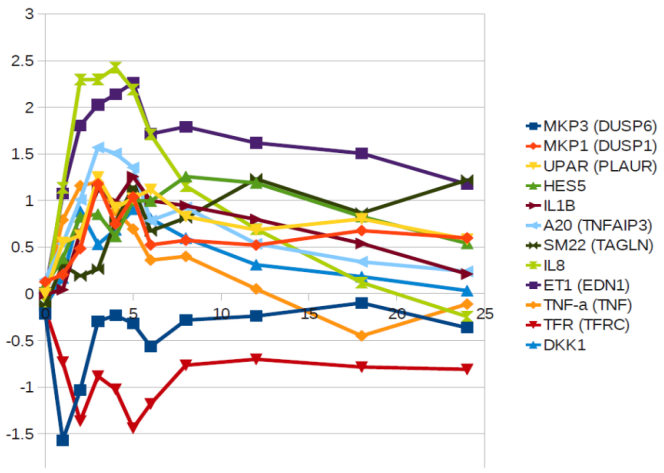


Fig. 3. Relative expression of selected mRNA upon Calcium stimulation.
The X axis represents time duration of the experiment measured in hours. The Y axis represents the log_2 expression level of genes with respect to control.

2.2 The Process Hitting Framework

In order to model the dynamics of the system, we use the Process Hitting framework [18]. The Process Hitting (PH) gathers a finite number of concurrent processes grouped

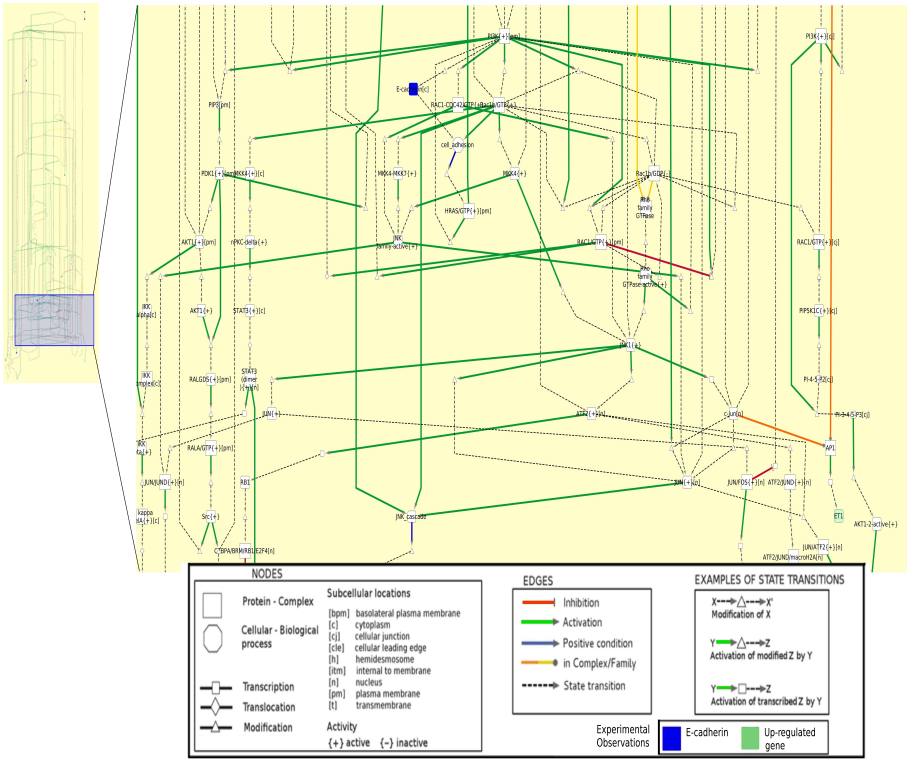


Fig. 2. Interaction graph linking E-cadherin with 12 genes of the time-series dataset. Blue nodes correspond to E-cadherin entities, red or green, to time-series genes, and cyan nodes to cellular processes. The graph is composed of 293 nodes and 375 edges (interactions). The set of nodes are composed of terminal nodes (proteins, complexes, mRNA expression, cellular state, biological processes and positive conditions) and of transient nodes (transcriptions, translocations, modifications and compounds). The set of edges are composed of interactions of type activation, inhibition, output, input and protein-family-member.

into a finite set of sorts. A sort stands for a component of a biological system while a process, which belongs to a unique sort, stands for one of its expression levels. At any time exactly one process of each sort is present. A state of the PH corresponds to such a set of processes. We denote here a process by a_i where a is the sort and i is the process identifier within the sort a. The concurrent interactions between processes are defined by a set of *actions*. Actions describe the replacement of a process by another of the same sort conditioned by the presence of at most one other process in the current state. An action is denoted $a_i \rightarrow b_j \Rsh b_k$, which is read as "$a_i$ *hits* b_j to make it bounce to b_k", where a_i, b_j, b_k are processes of sorts a and b, called respectively *hitter*, *target* and *bounce* of the action.

Definition 3 (Process Hitting). *A* Process Hitting *is a triple* $(\Sigma, L, \mathcal{H})$*, where:*

- $\Sigma = \{a, b, \dots\}$ *is the finite set of* sorts*;*
- $L = \prod_{a \in \Sigma} L_a$ *is the set of states with* $L_a = \{a_0, \dots, a_{l_a}\}$ *the finite set of* processes *of sort* $a \in \Sigma$ *and* l_a *a positive integer, with* $a \neq b \Rightarrow L_a \cap L_b = \emptyset$*;*
- $\mathcal{H} = \{a_i \rightarrow b_j \Rsh b_k \in L_a \times L_b \times L_b \mid (a, b) \in \Sigma^2 \wedge b_j \neq b_k \wedge a = b \Rightarrow a_i = b_j\}$ *is the finite set of* actions*.*

Given a state $s \in L$, the process of sort $a \in \Sigma$ present in s is denoted by $s[a]$. An action $h = a_i \rightarrow b_j \Rsh b_k \in \mathcal{H}$ is *playable* in $s \in L$ if and only if $s[a] = a_i$ and $s[b] = b_j$. In such a case, $(s \cdot h)$ stands for the state resulting from playing the action h in s, with $(s \cdot h)[b] = b_k$ and $\forall c \in \Sigma, c \neq b, (s \cdot h)[c] = s[c]$. In order to model the fact that a molecule in the interaction graph is influenced by various molecules, two types of modeling-scenarios can be proposed: cooperation and synchronization.

Modeling cooperation. The cooperation between processes to make another process bounce can be expressed in PH by building a *cooperative sort* [18]. Fig. 4 shows an example of a cooperative sort ab between sorts a and b, which is composed of 4 processes (one for each sub-state of the presence of processes in a and b). For the sake of clarity, processes of ab are indexed using the sub-state they represent. Hence, ab_{01} represents the sub-state $\langle a_0, b_1 \rangle$, and so on. Each process of sort a and b hits ab, which makes it bounce to the process reflecting the status of the sorts a and b (e.g., $a_1 \rightarrow ab_{00} \Rsh ab_{10}$ and $a_1 \rightarrow ab_{01} \Rsh ab_{11}$). Then, to represent the cooperation between processes a_1 and b_1, the process ab_{11} hits c_1 to make it bounce to c_2 instead of independent hits from a_1 and b_1. The same cooperative sort is used to make a_0 and b_0 cooperate to hit c_1 and make it bounce to c_0. Cooperation sort allows to model the fact that two components cooperate to hit another component.

Modeling synchronization. The synchronization sort implements another type of cooperation. If we refer to the example of Fig. 4 left, we can similarly construct a *synchronization sort* ab between sorts a and b, defined with also 4 processes. Then, component c is activated (c_1 bounces to c_2 or c_0 bounces to c_1) if either a or b are activated. Therefore, each one of these processes ab_{01}, ab_{10}, ab_{11} can activate c. In order to inhibit c, both sorts, a and b, need to be in the sub-state 0, i.e. ab_{00}. Notice that this rule is a combination of OR logical gates for activation and AND logical gates for inhibition. Imposing the synchronization sort to model a target component regulated independently by

multiple predecessors avoids oscillations in the behavior of the target component over time. These oscillations appear because each predecessor can independently activate the target component when it is active, but when one predecessor is inhibited, it inhibits the target component. This competition between the predecessors generates oscillations on the target component.

Example 1. Fig. 4 represents a PH $(\Sigma, L, \mathcal{H})$ with $\Sigma = \{a, b, c, ab\}$, and:

$$\begin{aligned} L_a &= \{a_0, a_1\}, \\ L_b &= \{b_0, b_1\}, \\ L_{ab} &= \{ab_{00}, ab_{01}, ab_{10}, ab_{11}\}, \\ L_c &= \{c_0, c_1, c_2\}. \end{aligned}$$

This example models a Biological Regulatory Network (BRN) where the component c has three qualitative levels, components a and b are Boolean and ab is a cooperative sort. In this BRN, ab inhibits c at level 2 through the cooperative sort ab (e.g. $ab_{00} \rightarrow c_2 \not\uparrow c_1$, $ab_{00} \rightarrow c_1 \not\uparrow c_0$) while a and b activate c through the cooperative sort ab (e.g. $ab_{11} \rightarrow c_0 \not\uparrow c_1$ $ab_{11} \rightarrow c_1 \not\uparrow c_2$). Indeed, the reachability of c_2 and c_0 is conditioned by a cooperation of a and b as explained above.

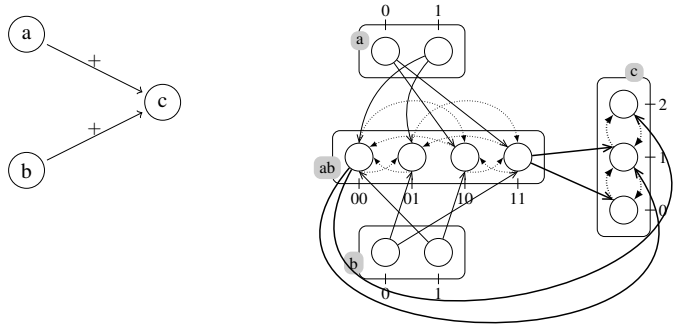


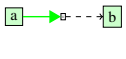
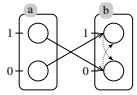
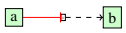
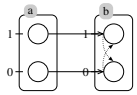
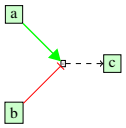
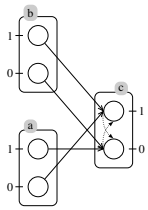
Fig. 4. (Left) biological pattern example. Nodes represent molecules (components) and edges, interactions. In this pattern components a and b cooperate to activate c. **(Right) equivalent PH model** with four sorts: three components (a, b and c) and a cooperative sort (ab). Actions targeting processes of c are drawn as thick lines.

2.3 Model construction (from RSTC to PH)

Modeling the RSTC network as a PH model. In order to model the RSTC network as a PH model we select known biological regulatory patterns (atomic set of biological components and their interacting roles), represented as biochemical reactions in the RSTC network and we propose their PH representation. Table 1 shows some examples of this transformation.

The automatic pattern selection and PH model generation algorithms use two procedures. The first one takes the graph as parameter argument and automatically browses

Table 1. Examples of patterns

Biological Patterns	PH Transformations	Descriptions
Simple activation 		This pattern model the activation of the component b by the component a.
Simple inhibition 		This pattern model the inhibition of the component b by the component a
Activation or inhibition 		This pattern model either the activation of the component c by the component a or the inhibition of the component c by the component b

it node by node and detects all the patterns in the graph. For each node (output node of the pattern) we call a recursive procedure, that allows to detect a minimal set of nodes (input node of the pattern) that has a direct influence over that node. This set of nodes plus the output node and the way input and output are linked form a pattern. The type of a pattern is determined by the type of the output node, the type of regulations that come to that node and the type of input nodes of the pattern. Consequently, the algorithm of patterns detection returns the pattern and its type to another procedure which translates the pattern into the PH formalism. This transformation takes care of different cases (cooperation, synchronization, simple activation, simple inhibition, etc.)

For example a molecule a cooperating with a molecule b to activate a molecule c (Fig. 4, left), is a regulatory pattern because it is a protein-complex biochemical reaction that appears at recurrent times. We model this pattern by four sorts (Fig. 4, right) a, b, c and ab. Sorts a, b and c stand for components a, b and c. The cooperative sort ab is introduced in order to characterize constraints on the components a and b. In the RSTC network, we find 25 regulatory patterns. We show some examples in Table 1.

Estimating the parameters for the PH-simulation from time-series gene expression data. Since the simulation of the execution of the PH actions is done stochastically, we need to relate each action with temporal and stochastic parameters introduced into the PH framework to achieve dynamic refinement [18]. To fire an action in the PH

framework we need to provide two parameters: (1) the rate $r = t^{-1}$, where t is the mean time for firing an action, and (2) the stochasticity absorption factor sa, which is introduced to control the variance of firing time of an action.

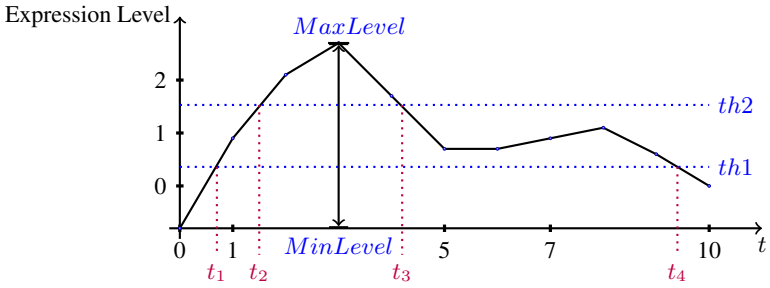


Fig. 5. Estimating temporal parameters from time series data: The mean firing time of an action that makes a component (mRNA expression) change of sub-state is estimated as $r_i = \frac{1}{t_i - t_{i-1}}$. $MaxLevel$ represents the maximum expression of a mRNA expression, while $MinLevel$, its minimum expression. The thresholds $th1$ and $th2$ define the PH discrete sub-states (e.g. 0,1,2) of a component according to its gene expression data.

For the model components which have a measurement in the time-series data we estimate the r and sa parameters and they are introduced in the PH model. The other components are assigned default parameters. In order to estimate r_i and sa_i for each action $h_i \in \mathcal{H}$, we need to know the different times t_i when the action could be fired as illustrated in Fig. 5. Each t_i represents the time at which we assume that a component moves from one process to another. Therefore the action that leads this change must be played at the rate $r_i = \frac{1}{t_i - t_{i-1}}$. The integer sa represents the window of firing the action at rate r : the larger the sa is, the smaller the variance around r is. Studies [2,3,22] have proposed more elaborated methods for parameters estimation from gene expression data. These methods are well adapted in the case of biochemical reactions where the concept of threshold is implicit. In the proposed case we assume an explicit threshold. Thus a basic estimation algorithm can be used for temporal and stochastic parameter estimation.

Discretization of time-series data. Because the outputs of a PH simulation are discrete traces of PH components, we discretized continuous experimental data to facilitate the comparison with simulation outputs. When looking at the time-series data (see Fig. 3) one can distinguish a high level of ¡¡¡¡¡¡¡ HEAD activity in early hours [$0h$-$5h$] and a low level in late hours [$5h$-$10h$]. Trend has been confirmed by using the SMA (Simple Moving Average) function of R package TTR which allows to smooth time-series data. We use the SMA function with parameter $n = 2$ and we have more than 50% of time-series data which present these two levels of activities. ======= activity in early hours [$0h$-$5h$] and a low level in late hours [$5h$-$10h$]. This

trend was confirmed by the SMA (Simple moving average) function of the R package TTR which allows us to smooth time-series data. We used the SMA function with paramter $n = 2$ and we observed that more than 50% of the time-series data presented these two levels of activities. ¿¿¿¿¿¿ 12c241a1d88dedd17eca66bd84aa7ba99f981d1c We implemented a discretization method to capture these two activation times. For each time-series, we introduced two thresholds *th1* and *th2* (see Fig. 5) were introduced: *th1* $= \frac{1}{3}(MaxLevel - MinLevel)$ and *th2* $= \frac{2}{3}(MaxLevel - MinLevel)$. In this way, the expression level in the range $[0 - th1]$ is at level 0, the one in the range $[th1 - th2]$ is at level 1, and the one in the last range is at level 2.

2.4 Simulation

We set the same initial conditions to PH components belonging to the same network layer, chosen from the RSTC structure. These initial conditions are detailed below and summarized in Fig. 6.

- **Receptor layer: E-cadherin.** We choose the pulse signal for the input node E-cadherin to be active for a duration of 5 time units in average. This choice was made in orders to take into account the average time of the Calcium stimuli effect.
- **Signaling layer: signaling proteins.** The components in this layer are activated and inhibited with the same rate and the same stochasticity absorption factor. The actions between a controller component A and a controlled component B are constrained so that B is first activated by A and then inhibited. That is, the time interval in which an inhibition action from A to B fires is greater than the time interval in which an activation action from A to B fires. Additionally, these two time intervals must not overlap. These constraints can be seen as reachability constraints from the entry node (E-cadherin) to the output nodes (mRNA expression). The values of these parameters are selected by considering the delay of signal transduction from the entry node to the output nodes.
- **Transcription layer: transcription factors.** In this layer, the activation/inhibition over a transcription factor (TF) comes from signaling proteins; however, for all TFs we introduced an auto-inhibition action that represents their degradation over time.
- **mRNA expression.** The mRNA expression are activated or inhibited according to the estimated values from time-series data.

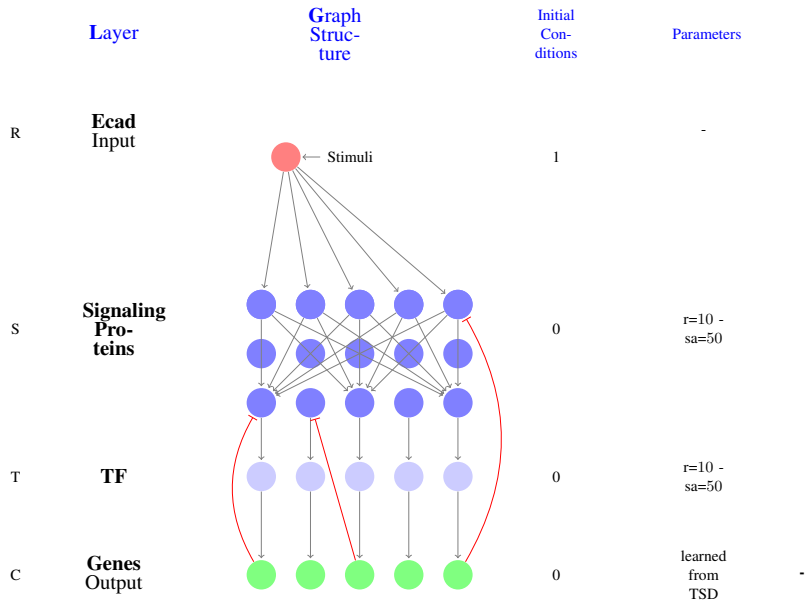


Fig. 6. RSTC network structure and initial conditions assigned to each node in the layer

2.5 Automatic analysis of simulation traces

¡¡¡¡¡¡¡ HEAD Due to the stochastic and concurrent aspects of the system, each execution of the model can generate dynamics which are different, one of the others. Therefore, to validate the proposed model, it was decided to analyze the traces generated by each component for a set of simulations of the model. The idea is to compute the percentage of traces that reproduce the expected dynamic of the system. To achieve this goal, we take each trace generated at each simulation for a given component and forward it to an automaton ($\mathcal{A}_i$) recognizing the set of acceptance traces of that component. Thus, the number of accepted traces ($Trace_{accp}$) can be counted. Therefore the percentage of accept traces is $\frac{Trace_{accp}}{Trace_N}$ if $Trace_N$ is the total number of simulation. Following this we introduce the concept of tolerance in accepting traces. It's mean that an automaton can accepted a trace with a difference of one or more level (s) at each state. But in the case study we just use a tolerance T_1 that allow to accept a difference of one between the input trace and the expected trace at each state. ======= Due to the stochastic and concurrent aspects of the system, each execution of the model can generate a different dynamic trace. Therefore, to validate the proposed model, we analysed the traces generated by each component for a set of simulations of the model. The idea was to calculate the percentage of traces that reproduced the expected dynamic of the system. To achieve this goal, we take each trace generated at each simulation for a given component and passed it to an automata ($\mathcal{A}_i$) that recongnize the experimental trace of that component. Thus we can count the number of accepted traces ($Trace_{accp}$); the percentage of accepted traces is $\frac{Trace_{accp}}{Trace_N}$ if we assume that $Trace_N$ is the total number of

simulations. Following this we introduce the concept of tolerance in accepting traces. It means that an automata can accept a trace with a difference of one or more levels at each state. In our case study we used a tolerance T_1 that allows accepting a difference of one between the simulated trace and the expected trace at each state. ¿¿¿¿¿¿¿ 12c241a1d88dedd17eca66bd84aa7ba99f981d1c

3 Results

3.1 Automatic generation of PH model from the PID network

PH models are written using the PINT[3] format. PINT implements stochastic simulations and static analyses for computing dynamical properties on very large-scale PH models. For the PINT code generation two procedures were used. The first procedure detects motifs (controller and controlled components) in graphs from the Pathway Interaction Database; the second, generates the PINT code by choosing an adequate concurrency rule, based on synchronization sorts, to represent the motif dynamic in PH. With this method it is possible to convert the whole content of the PID database into a PH model, as well as individual pre-selected pathways, as is the case for the system under study. It is implemented in Java and available upon request.

3.2 Simulation of Calcium stimulated biological system

We simulated the model with and without the inclusion of the synchronization sort. In the following, we present the results of the simulation.

Without the introduction of the synchronization sort. One can notice in Fig. 7 the occurrence of oscillations. Whereas it is not the expected behavior from the biological system, it is coherent with the choice of the modeling and the way the simulator works as explained in Section 2.2. In this simulation, cooperation sorts were used to model multiple controllers of a common controlled (target) component. It is important to notice that the intensity of the oscillation is linked with the size of the concurrence, i.e. the number of controllers a controlled component has. Despite the presence of the oscillations, the model reproduces expected dynamical behaviors namely the dynamics of components, the signal transduction and takes into account the stochastic and time aspect of the model.

[3] `http://process.hitting.free.fr`

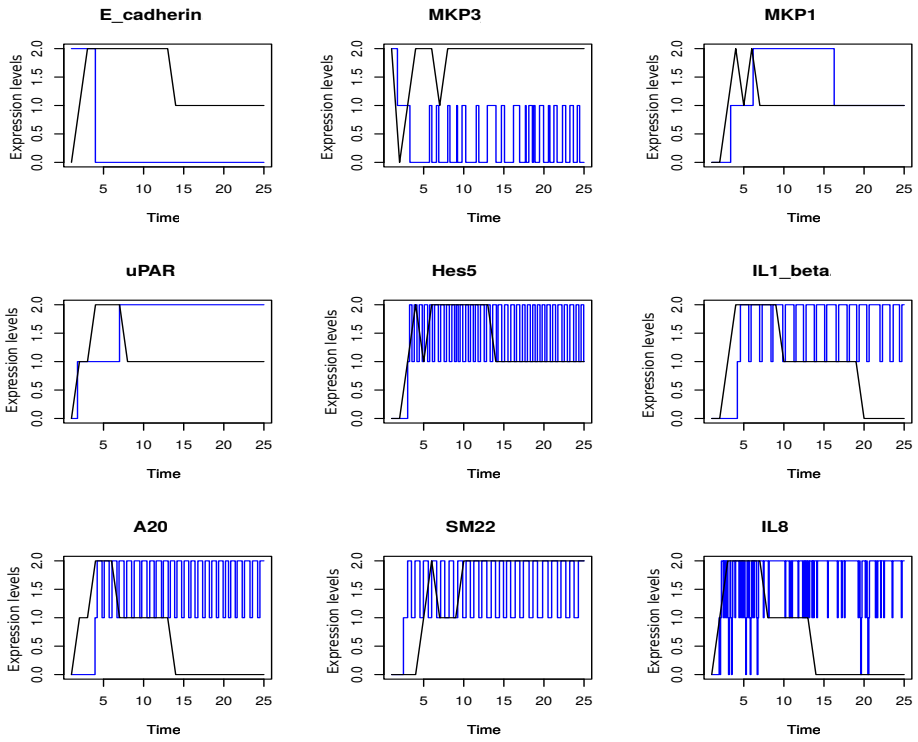


Fig. 7. Results of system simulations without introducing the synchronization sort for 9 genes. The traces representing the discretized time-series data are shown as black lines. The traces representing the simulated traces are shown as blue lines.

With the introduction of the synchronization sort. In Fig. 8 we can see that the introduction of the synchronization sort significantly reduces the impact of concurrency. The result shows a clear elimination of the previously observed oscillations (Fig. 7). Comparing the simulated results with the ones observed experimentally, we found four different cases. We found 5 simulation traces (IL8, uPAR, IL1_beta, ET1, A20) that matched the sequence of all the component expression levels perfectly. In this case, delays exist among simulation and experiment but these delays are not comparable since experimental time-points are measured in hours and simulation-units for the simulated PH model. We found 6 simulation traces (MKP1, MKP3, Hes5, SM22, TfR, DKK1) that matched the sequence of experimental discrete expression levels missing one expression-level. We found 1 components (TNF-alpha) in which at least 2 expression levels are missed.

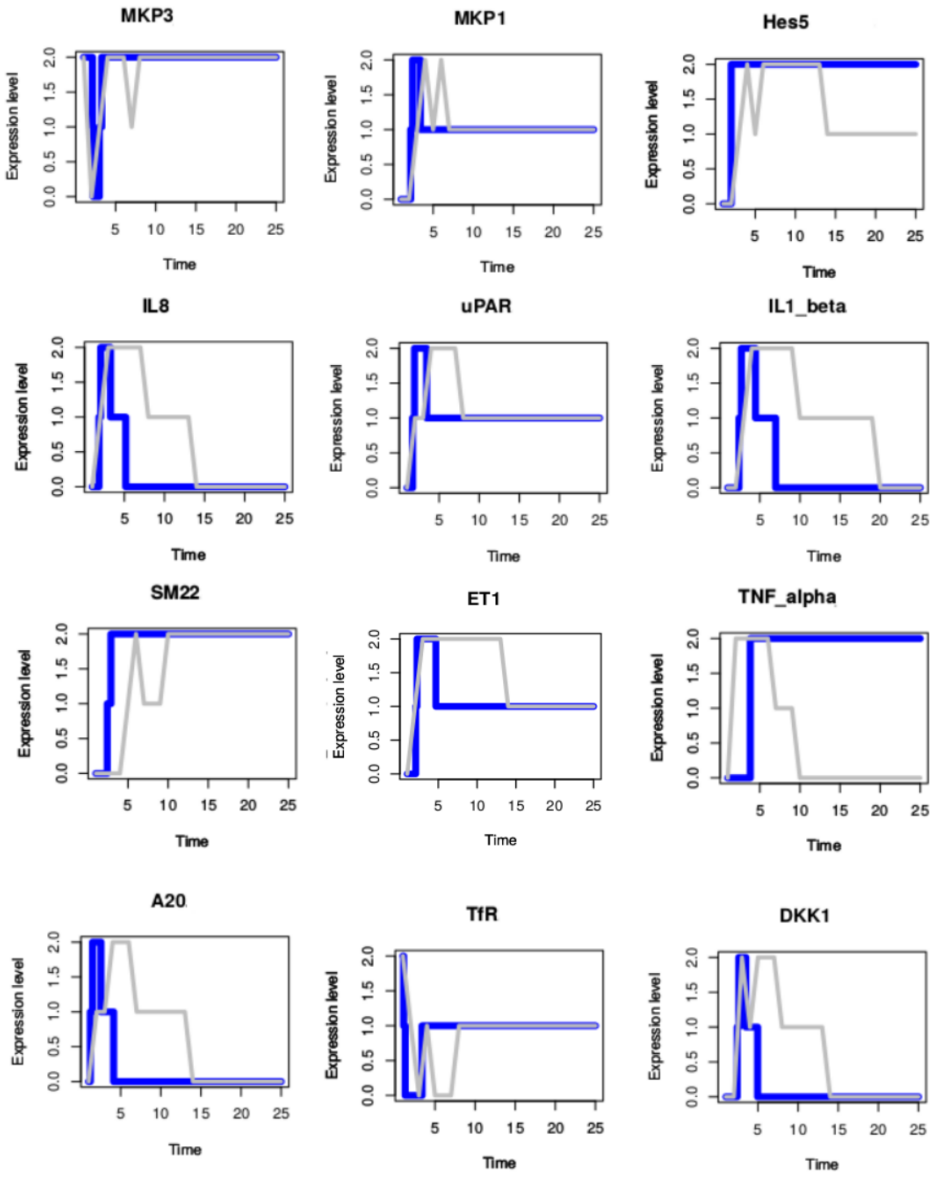


Fig. 8. Results of simulations by introducing the synchronization sort. The gray traces represent the experimental expected behaviors from the discretization of the time-series data. The blue traces show the simulated behavior.

Simulating biological processes. To validate our model, we studied the prediction of non-observed components of such a system and we focused on biological processes linked to Calcium stimulation, such as keratinocyte-differentiation, cell-adhesion and cell-cycle arrest. Our results are shown in Fig. 9 and confirm literature experimental evidences on these processes. In the case of keratinocyte-differentiation, this was a functional behavior measured on the cultured cells upon Calcium stimulation, so there was experimental evidences of this effect before measuring the gene expression. In the case of cell-cycle arrest, the switch-on of this component represents the fact that the E-cadherin stimulated model predicts the stop of growth, as confirmed by literature in human and mouse keratinocytes [13]. Finally, the cell-adhesion component is predicted to switch-on, also in according to published evidence [28] in human and mouse keratinocytes.

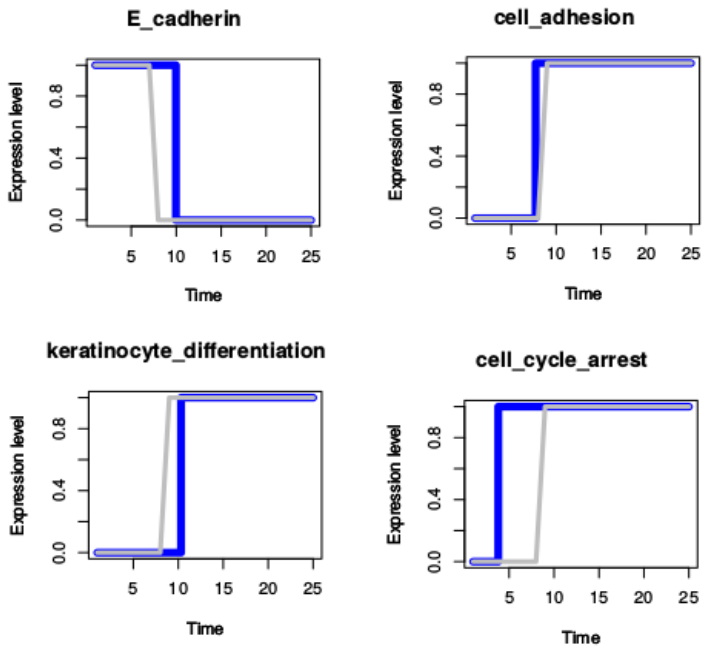


Fig. 9. Results of the prediction of biological processes. The gray traces represent the experimental and literature-based evidence. The blue traces show the simulated behavior of E-cadherin and three biological processes.

3.3 Model validation: traces analysis

¡¡¡¡¡¡¡ HEAD To validate the results of the simulations, we automatically analyzed the traces generated by a set of 100 simulations. Table 3 shows the results of the percentage of acceptance for the traces of each of the 12 mRNA expression. One can observed

¡¡¡¡¡¡¡ HEAD
Table 2. Percentage of acceptance traces. First column represents the Automaton ($\mathcal{A}_i(w)$, where $\mathcal{A}_i$ is the Automaton and w is the word recognized by $\mathcal{A}_i$) that is used to check if a given trace is acceptance for a component in the second column. One can observed that many components can be recognized by the same Automaton. In the third column we have the percentage of acceptance traces. Finally in the fourth column are the percentage of acceptance with a tolerance of one level (T_1)

=======
Table 3. Percentage of accepted traces. The first column represents the Automata ($\mathcal{A}_i(w)$, where $\mathcal{A}_i$ is the Automata and w is the word recongnized by $\mathcal{A}_i$) that is used to check if a given trace is accepted for a component in the second column. One can observe that many components can be recognized by the same Automata. In the third column we show the percentage of accepted traces and in the fourth column, the percentage of acceptance with a tolerence of one level (T_1).

¿¿¿¿¿¿¿ 12c241a1d88dedd17eca66bd84aa7ba99f981d1c

Automate	**components**	% **of acceptance**	% **of acceptance** T_1
$\mathcal{A}_2(01210)$	A20	91	100
$\mathcal{A}_2(01210)$	IL1_beta	81	100
$\mathcal{A}_2(01210)$	IL8	93	100
$\mathcal{A}_2(01210)$	TNF_alpha	0	0
$\mathcal{A}_3(01211)$	uPar	76	99
$\mathcal{A}_3(01211)$	ET1	8	19
$\mathcal{A}_4(0121210)$	DKK1	13	43
$\mathcal{A}_5(0121211)$	Hes5	0	17
$\mathcal{A}_5(0121211)$	MKP1	9	97
$\mathcal{A}_6(0212)$	SM22	11	100
$\mathcal{A}_7(02010)$	MKP3	11	98
$\mathcal{A}_8(02121)$	Tfr	0	94

that, there are components with a good rate of acceptance (A20, IL1_beta, IL8, uPar). The reason is probably that they have traces that are more simple to reproduce. Another observation is that they are others components which have low rate of acceptance (ET1, DKK1, MKP1, MKP3). We think that their traces are more complicate to reproduce. But in all the case the rate of acceptance increase with tolerance in automatic verification. ======= To validate the results of the simulations, we automatically analysed the traces generated by a set of 100 simulations. Table 3 shows the results of the percentage of acceptance for the traces of each of the 12 mRNA expressions. One can observe that there are components with a good rate of acceptance (A20, IL1_beta, IL8, uPar), which may be easy to reproduce. Traces (ET1, DKK1, MKP1, MKP3) with a lower rate of acceptance may be more difficult to reproduce. In all the cases, the rate of acceptance increases when allowing a tolerance of 1 in the automatic verification. ¿¿¿¿¿¿¿ 12c241a1d88dedd17eca66bd84aa7ba99f981d1c

4 Conclusion

This work describes the steps towards the integration of time-series data in large-scale cell-based models. We proposed an automatic method to build a timed and stochastic PH model from pathways of biochemical reactions present in the Pathway Interaction Database (PID). As a case-study we built a model combining signaling and transcription events relevant to keratinocyte differentiation induced by Calcium, which linked E-cadherin nodes and 12 genes, which expression profiles was measured upon Calcium stimulation over time. The interaction graph represented by the model had 293 nodes and 375 edges. ¿¿¿¿¿¿¿ 12c241a1d88dedd17eca66bd84aa7ba99f981d1c We proposed a method to discretize time-series gene expression data, so they can be integrated to the PH simulations and logically explained by the PH stochastic analyses. Additionally, we implemented a method to automatically estimate the temporal and stochastic parameters for the PH simulation, so this estimation process will not be biased by over fitting.

Our results show that we can observe dynamic effects on 11 out of 12 genes, for which 5 of them represent accurate predictions, and 6 of them missed few dynamic levels. This error may be also a result from the incompleteness of the regulatory information in PID. Moreover, when observing the predicted behavior of biological processes linked to Calcium stimulation, our predictions agreed with experimental and literature-based evidences. Overall, with this work we show the feasibility of modeling and simulating large-scale networks with very few parameter estimation and having good quality predictions.

As perspectives of this work we intend to study the effects of computing automatically the concurrent rules on this system. Also, we intend to improve the model prediction quality by empirically obtaining the dynamics of the system components by performing large stochastic simulations, as well as by implementing static analysis of quantitative properties by adding probabilistic features to the PH static solver.

5 Acknowledgements

This work was supported by a PhD grant from the CNRS and the French region *Pays de la Loire* and grants from the German Ministry for Research and Education (BMBF) funding program MedSys (grant number FKZ0315401A) and AGENET (FKZ0315898).

A Algorithm of patterns detection

Here are the algorithms that allow to detect and construct a process hitting model from an RSTC network. These algorithms have a polynomial time running that correspond to the running time of the procedure 2.

Proposition 1. *Algorithm 2 has a time complexity of* $\mathcal{O}(|V|\log(h))$. *Where* h *is the average height of the patterns in the RSTC network. In the worst case* $h = \log_V(|V|)$.

Algorithm 1 : Algorithm for Pattern detection in an RSTC Network in order to generate the equivalent model in the PH formalism

Require: Net {The RSTC network}
Ensure: generate the PH Model associated to Net
1: **for all** Node n in Net.getSetOfNodes() **do**
2: Pat = detectPattern (Net, n)
3: patternInPHModel (out, Pat)
4: **end for**

Algorithm 2 : Algorithm for pattern detection, function detectPattern (Net, n)

Require: Net, n {Net is the network and n is the current node}
Ensure: Build a set of nodes associated to node n that we call pattern.

```
 1: switch (n)
 2: case TerminalNode:
 3:     add node n to the pattern Pat
 4:     numberPredecessor= n.getNumberOfPredecessor()
 5:     switch (numberPredecessor)
 6:     case 1:
 7:         for all p in setOfPredecessor (n) do
 8:             switch (p)
 9:             case TerminalNode:
10:                 add node n to the pattern Pat
11:             case TransientNode:
12:                 detectPattern (Net, p);
13:             end switch
14:         end for
15:         Set the code of pattern Pat;
16:         return Pat;
17:     case 2:
18:
19:     end switch
20: case TransientNode:
21:     numberPredecessor= n.getNumberOfPredecessor()
22:     switch (numberPredecessor)
23:     case 1:
24:         for all p in setOfPredecessor (n) do
25:             switch (p)
26:             case TerminalNode:
27:                 added node to the pattern Pat;
28:             case TransientNode:
29:                 detectPattern (Net, p);
30:             end switch
31:         end for
32:         Set the code of pattern Pat;
33:         return Pat;
34:     case 2:
35:
36:     end switch
37: end switch
```

Algorithm 3 : Algorithm for writing a given pattern into a file, function patternInPHModel (out, Pat)

Require: out, Pat {Pat is The pattern to be translated into the PH Model, out is the output file}
Ensure: The correspondent PH Model of the given pattern Pat will write into the file out
$nocp$ = Pat.getNumberOfComponents() {Number of the components of the pattern Pat}
$tabPat$ = Pat.getTableOfPattern() {return the components of the pattern in $tabPat$}
switch ($nocp$)
case 2**:**
switch ($code$)
case A**:**
out.write ($tabPat[1]$ 1 $\rightarrow$ $tabPat[0]$ 0 1 r_a sa_a); {Component $tabPat[1]$ activates component $tabPat[0]$ with $r = r_a$ and $sa = sa_a$ }
case I**:**
out.write ($tabPat[1]$ 0 $\rightarrow$ $tabPat[0]$ 1 0 r_i sa_i); {Component $tabPat[1]$ inhibits component $tabPat[0]$ with $r = r_i$ and $sa = sa_i$ }
end switch
case 3**:**
switch ($code$)
case C**:**
out.write (coop ([$tabPat[2]$;$tabPat[1]$)] $\rightarrow$ $tabPat[0]$ 0 1); {Cooperation between $tabPat[1]$ and $tabPat[2]$ to activate $tabPat[0]$}
case S**:**
out.write (coop ([$tabPat[2]$;$tabPat[1]$)] $\rightarrow$ $tabPat[0]$ 0 1); {Synchronization between $tabPat[1]$ and $tabPat[2]$ to activate $tabPat[0]$}
default:
out.write ((*unknow pattern*));
end switch
end switch

References

1. Jamil Ahmad, Olivier Roux, Gilles Bernot, Jean-Paul Comet, and Adrien Richard. Analysing formal models of genetic regulatory networks with delays. *International Journal of Bioinformatics Research and Applications (IJBRA)*, 4(3):240–262, 2008.
2. David Spieler Aleksandr Andreychenko, Linar Mikeev and Verena Wolf. Approximate maximum likelihood estimation for stochastic chemical kinetics. *EURASIP Journal on Bioinformatics and Systems Biology*.
3. Gregory Batt, Michel Page, Irene Cantone, Gregor Goessler, Pedro Monteiro, and Hidde de Jong. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. *Bioinformatics*, 26(18):i603–i610, 2010.
4. Grégory Batt, Ramzi Ben Salah, and Oded Maler. *On timed models of gene networks*. Springer, 2007.
5. Hauke Busch, David Camacho-Trullio, Zbigniew Rogon, Kai Breuhahn, Peter Angel, Roland Eils, and Axel Szabowski. Gene network dynamics controlling keratinocyte migration. *Molecular systems biology*, 4(1):199, 2008.
6. Claudine Chaouiya, Elisabeth Remy, Brigitte Mossé, and Denis Thieffry. Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. In *Positive Systems*, pages 119–126. Springer, 2003.
7. Hidde De Jong, Johannes Geiselmann, Céline Hernandez, and Michel Page. Genetic network analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.
8. Timothy S Gardner, Diego Di Bernardo, David Lorenz, and James J Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–105, 2003.
9. Carito Guziolowski, Aristotelis Kittas, Florian Dittmann, and Niels Grabe. Automatic generation of causal networks linking growth factor stimuli to functional cell state changes. *FEBS Journal*, 279(18):3462–3474, 2012.
10. Carito Guziolowski, Santiago Videla, Federica Eduati, Sven Thiele, Thomas Cokelaer, Anne Siegel, and Julio Saez-Rodriguez. Exhaustively characterizing feasible logic models of a signaling network using answer set programming. *Bioinformatics*, 29(18):2320–2326, 2013.
11. Monika Heiner, David Gilbert, and Robin Donaldson. Petri nets for systems and synthetic biology. In *Formal methods for computational systems biology*, pages 215–264. Springer, 2008.
12. Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.
13. C. Kolly, M. M. Suter, and E. J. Muller. Proliferation, cell cycle exit, and onset of terminal differentiation in cultured keratinocytes: pre-programmed pathways in control of C-Myc and Notch1 prevail over extracellular calcium signals. *J. Invest. Dermatol.*, 124(5):1014–1025, May 2005.
14. Aidan MacNamara, Camille Terfve, David Henriques, Beatriz Peñalver Bernabé, and Julio Saez-Rodriguez. State–time spectrum of signal transduction logic models. *Physical Biology*, 9(4):045003, 2012.
15. Mylène Maurin, Morgan Magnin, and Olivier Roux. Modeling of genetic regulatory network in stochastic π-calculus. In *Bioinformatics and Computational Biology*, pages 282–294. Springer, 2009.
16. Alexander Mitsos, Ioannis N Melas, Paraskeuas Siminelakis, Aikaterini D Chairakaki, Julio Saez-Rodriguez, and Leonidas G Alexopoulos. Identifying drug effects via pathway alterations using an integer linear programming optimization formulation on phosphoproteomic data. *PLoS computational biology*, 5(12):e1000591, 2009.

17. Michael K. Molloy. Performance analysis using stochastic petri nets. *Computers, IEEE Transactions on*, 100(9):913–917, 1982.
18. Loïc Paulevé, Morgan Magnin, and Olivier Roux. Refining dynamics of gene regulatory networks in a stochastic π-calculus framework. In *Transactions on Computational Systems Biology XIII*, pages 171–191. Springer, 2011.
19. Andrea Pinna, Nicola Soranzo, and Alberto de la Fuente. From knockouts to networks: Establishing direct cause-effect relationships through graph analysis. *PLoS ONE*, 5(10):e12912, 10 2010.
20. Riccardo Porreca, Eugenio Cinquemani, John Lygeros, and Giancarlo Ferrari-Trecate. Identification of genetic network dynamics with unate structure. *Bioinformatics*, 26(9):1239–1245, 2010.
21. Corrado Priami. Stochastic π-calculus. *The Computer Journal*, 38(7):578–589, 1995.
22. R. Altman S. Reinker and J. Timmer. Parameter estimation in stochastic biochemical reactions. *IEE Proceedings - Systems Biology*, 153, 2006.
23. Carl F Schaefer, Kira Anthony, Shiva Krupa, Jeffrey Buchoff, Matthew Day, Timo Hannay, and Kenneth H Buetow. Pid: the pathway interaction database. *Nucleic acids research*, 37(suppl 1):D674–D679, 2009.
24. Heike Siebert and Alexander Bockmayr. Incorporating time delays into the logical analysis of gene regulatory networks. In *Computational Methods in Systems Biology*, volume 4210 of *LNCS*, pages 169–183. Springer, 2006.
25. El Houssine Snoussi. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. *Dynamics and stability of Systems*, 4(3-4):565–583, 1989.
26. Denis Thieffry and René Thomas. Dynamical behaviour of biological regulatory networksâii. immunity control in bacteriophage lambda. *Bulletin of Mathematical Biology*, 57(2):277–297, 1995.
27. René Thomas. Boolean formalization of genetic control circuits. *Journal of Theoretical Biology*, 42(3):563 – 585, 1973.
28. C. L. Tu, W. Chang, and D. D. Bikle. The calcium-sensing receptor-dependent regulation of cell-cell adhesion and keratinocyte differentiation requires Rho and filamin A. *J. Invest. Dermatol.*, 131(5):1119–1128, May 2011.
29. S Van Goethem, J-M Jacquet, Lubos Brim, and D Šafránek. Timed modelling of gene networks with arbitrarily precise expression discretization. *Electronic Notes in Theoretical Computer Science*, 293:67–81, 2013.
30. Namhee Yu, Jihae Seo, Kyoohyoung Rho, Yeongjun Jang, Jinah Park, Wan Kyu Kim, and Sanghyuk Lee. hipathdb: a human-integrated pathway database with facile visualization. *Nucleic acids research*, 40(D1):D797–D802, 2012.