Integrating time-series data on large-scale cell-based models: application to skin differentiation

Louis Fippo Fitime
Olivier Roux
and Carito Guziolowski
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.
Email: Louis.Fippo-Fitime@irccyn.ec-nantes.fr
Carito.Guziolowski@irccyn.ec-nantes.fr

***Abstract*—In this work we propose an automatic way of generating and verifying formal models of signaling and transcriptional events, gathered in large-scale regulatory networks, by integrating temporal and stochastic aspects of the expression of some biological components, measured during a real experiment. To achieve this we rely on the Process Hitting (PH) formalism and an stochastic simulation of the built PH hybrid model. On one hand this choice limits us to discrete data and on the other hand, allows us to handle models built upon large-scale interaction graphs, to express different levels of logic rules between a node and its direct predecessors, and to reproduce dynamic behaviors of the components with minimal stochastic parameter estimation. The model proposed is based on a real case study of keratinocyte differentiation, in which gene time-series data was generated upon Calcium stimulation.**

This work has the following contributions. First, we built an interaction graph linking a signaling molecule, E-cadherin (Calcium sensitive protein), to genes present in our time-series data and to key cellular processes for our case study, such as keratinocyte-differentiation and cellular-proliferation. This graph was automatically extracted from the Pathway Interaction Database (PID) using a previously proposed method. Second, we propose an automatic transformation of selected known biological patterns present in PID in order to generate PH modules; adding necessary constraints to the PH model to avoid oscillations. Third, we proposed a way of estimating temporal and stochastic parameters from time-series expression data to model the measured genes. These parameters are used for the stochastic simulation of the model. Finally, we discretized the experimental data to allow the comparison with simulation results for the above mentioned case study analysis.

We show that our approach allows us to reproduce different dynamic behaviors of the components in such a large network, by stochastically simulating this hybrid model. We are able to generate, for some cases, a close match to the experimental results with very few parameter calibration.

***Keywords*—Times-series data integration, large-scale biological network, stochastic and concurrent models.**

I. Introduction

The comprehension of 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, which analysis requires to preliminary build a mathematical or computational model. By just considering qualitative regulatory effects between components, biologic regulatory networks depict fairly well biological systems, and can be built upon public repositories such as the Pathways Interaction Database [1], and hiPathDB [2] for human regulatory knowledge.

This work aims to propose a dynamical model of large-scale systems based on the formal integration (complete validation/invalidation) of high-throughput experimental time-series data. So far this idea has been addressed separately by approaches that either: (a) focus first on modeling at small-scale the system and then on refining or improving it through the fitting with some data points, such as methods based on differential equations [3], [4], [5], (b) integrate in an efficient and complete fashion large-scale models and high-throughput data regardless from the system dynamics [6], [7], or (c) fit dynamical data to middle-scale networks using an stochastic sampling of the space of behaviors and therefore without guarantee on finding global optima [8]. Therefore, with this work we intend to fill the gaps between the previously cited methodologies and converge to a more realistic model of biological behavior.

For modeling and analyzing stochastic and concurent biological systems many formalisms have been introduced such as stochastic Petri Nets which is a suitable approach for the representation of parallel systems [9]. They have been successfully and systematically applied in many areas, and the specification Petri Nets allows an accurate modeling of a wide range of systems including biological systems [10]. The major problem of Stochastic Petri Nets is that they do not generally lead to compact models. In addition, they do not provide results to deal with the state space explosion and are thus generally computationally expensive. Another approach is Stochastic pi-calculus introduced by [11] and used in [12] for the modeling of biological systems. Stochastic pi-calculus has a good expressivity and is well adapted for the use of compositionnal approach.

For modeling and analyzing the biological system we rely on a Stochastic pi-calculus formalism which is the Process Hitting (PH) framework [13], since it is especially useful for studying systems composed of biochemical interactions, and provides stochastic simulation as well as efficient static methods to model 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. So far, this method has been successfully demonstrated only on very well-known systems and without exploiting high-throughput measures. We believe, however, that the use of high-throughput data has become unavoidable with the advent of massive, publicly available data sets in the form of well-standardized DNA microarray data and, more recently, in the form of phospho-proteomics data.

The main results of this work are: first, we built an interaction graph linking a signaling molecule, E-cadherin (Calcium sensitive protein), to genes present in our time-series data and to key cellular processes for our case study, such as keratinocyte-differentiation and cellular-proliferation. This graph was automatically extracted from PID. Second, we propose an automatic transformation of selected known biological patterns present in PID in order to generate PH modules; adding necessary constraints to the PH model to avoid oscillations. Third, we propose a way of estimating temporal and stochastic parameters from time-series expression data to model the measured genes. These parameters are used for the stochastic simulation of the model. Finally, we discretized the experimental data to allow the comparison with simulation results for the above mentioned case study analysis.

II. Data and Methods

The general workflow for integrating time-series data and model is depicted in Fig. 1 and comprises the following steps:

- **Formalization of biological model.** Building a computational hybrid model from biological network by automatic detection of known biological patterns.
- **Temporal parameters estimation.** Estimation of temporal parameters from times-series data to calibrate the model.
- **Integration of temporal parameters.**
- **Discretization of times-series data.** Discretization of time series data to compare with discrete simulation results.
- **Simulation and validation.** Compare the results of simulation with the discretized times-series data.

A. Data

1) 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, generated from the Pathway Interaction Database (PID). In order to build this network, we selected a set of seed nodes related to the biological process studied. The seed nodes for our case study are: (1) *E-cadherin*, which is a protein having Ca 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 NCI-PID database by using a subgraph algorithm to link the seed nodes [14]. In Fig. 2 we show the RSTC network obtained.

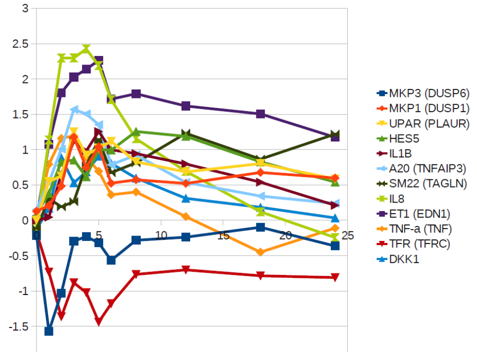


Fig. 3. **Plot of 12 selected genes. X axis represents time duration of the experiment measured in hours. Y axis represents the log_2 expression level of genes with respect to control.**

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

2) Time-series microarray data description: To illustrate our approach we use the time-series microarray data from Calcium stimulated keratinocyte cells measured at 10 time-points. 200 transcripts were selected for their dynamic patterns, that is, their fold expression with respect to the non-stimulated cell was significant in at least one time point. We included in our model a set of 12 of the 200 selected (see Fig. 3), because we were able to retrieve the regulatory mechanisms upstream these 12 genes from public repositories of biochemical reactions. The full dataset (data not shown) was produced by the German Cancer Research Center (DKFZ) and is currently in the process of being published.

B. The Process Hitting Framework

Process Hitting (PH) gathers a finite number of concurrent processes grouped 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 by $a_i \to b_j \Rsh b_k$, which is read as "a_i *hits* b_j to make it bounce to b_k", where

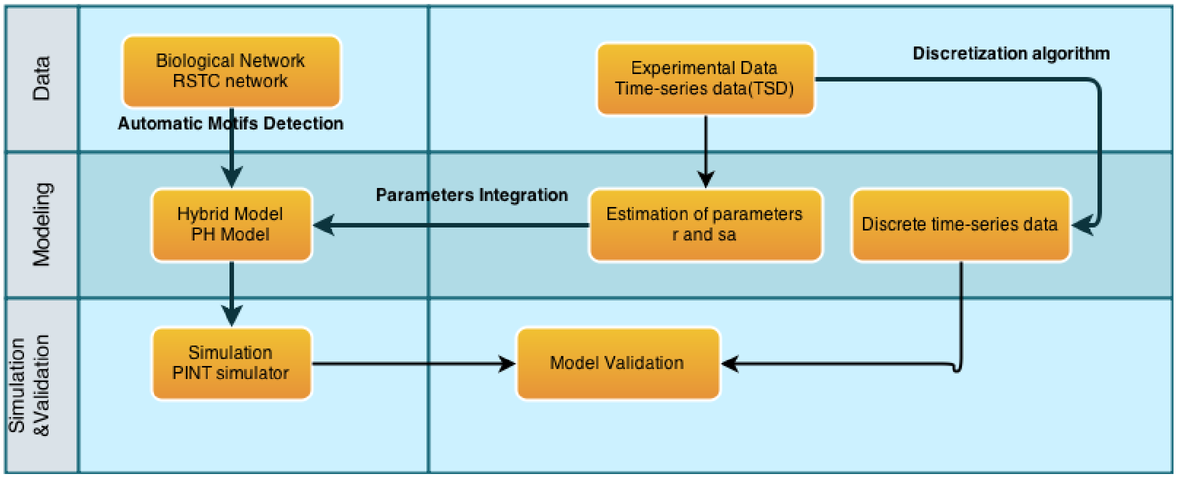


Fig. 1. **Workflow for integrating stochastic and temporal information in a large-scale model of a biological network.**

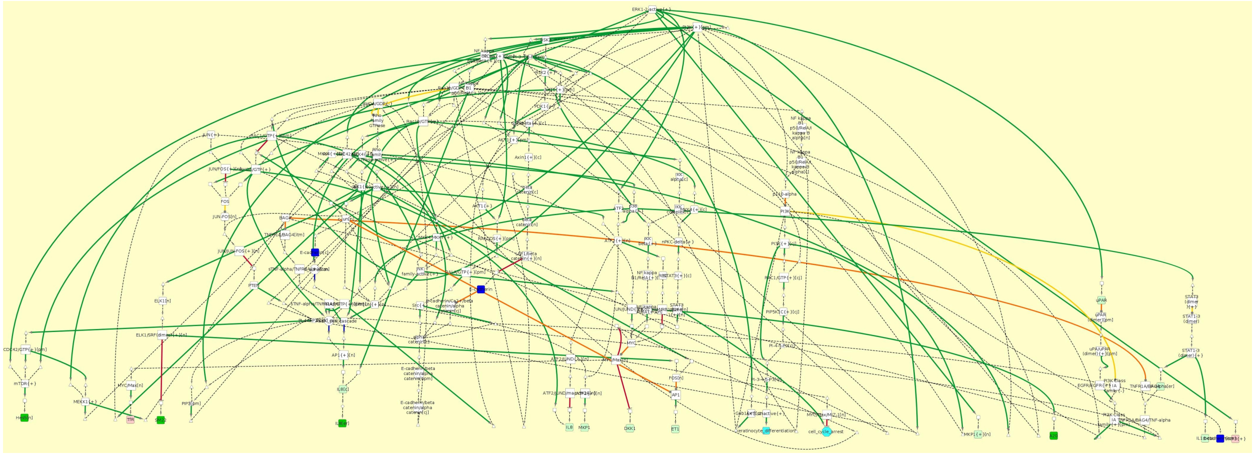


Fig. 2. **RSTC network. Interaction graph that describe the biological case study. It is composed of 293 nodes and 375 edges (interactions). The set of nodes are composed of terminal nodes (proteins, complexes, genes, cellular state, biological processes and positive conditions) and of transient nodes (transcriptions, translocations, modifications and compounds). The set of edges are composed of interactions of type activation (agent), inhibition, output, input and protein-family-member.**

a_i, b_j, b_k are processes of sorts a and b, called respectively *hitter*, *target* and *bounce* of the action.

Definition 2 (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* $a \neq b \Rightarrow L_a \cap L_b = \emptyset$*;*
- $\mathcal{H} = \{a_i \to b_j \upharpoonright b_k \in L_a \times L_b \times L_b \mid (a, b) \in \Sigma^2 \wedge b_j \neq b_k \wedge a = b \Rightarrow a_i = b_j\}$ *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 $h = a_i \to b_j \upharpoonright b_k \in \mathcal{H}$ 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 $\forall c \in \Sigma, c \neq b, (s \cdot h)[c] = s[c]$.

1) Modeling cooperation: As described in [13], the cooperation between processes to make another process bounce can be expressed in PH by building a *cooperative sort*. Fig. 4 shows an example of a cooperative sort ab between sorts a and b, defined with 4 processes (one for each sub-state of the presence of processes a_1 and b_1). 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 hit ab, which makes it bounce to the process reflecting the status of the sorts a and b (e.g., $a_1 \to ab_{00} \upharpoonright ab_{10}$ and $a_1 \to ab_{01} \upharpoonright ab_{11}$). 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.

2) Modeling synchronization: The synchronization sort implement another type of cooperation. If we refer to the example of Fig. 4, and asume that ab is the *synchronization sort* between sorts a and b, defined with also 4 processes. Then, component c is activated (c_1 bounces to c_2) if either a and b are activated. Therefore, each one of these processes ab_1, ab_2, ab_3 can

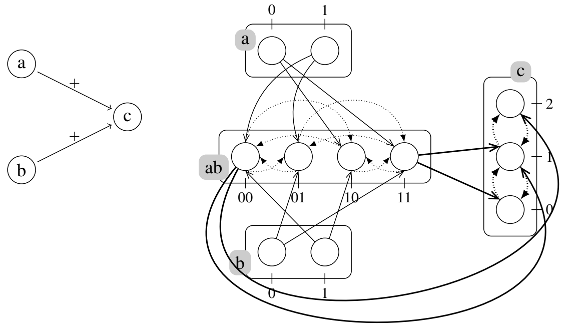


Fig. 4. **(Left) Biological pattern example. Nodes are components and edges are interactions For instance, components a and b cooperate to activate c. (Right) equivalent PH model. A PH example with four sorts: three components (a, b and c) and a cooperative sort (ab). Actions targeting processes of c are in thick lines.**

activate c_1, in order to inhibit c, both a and b are needed. Notice that this rule is a combination of OR logical gates for activation and AND logical gates for inhibition. The interest of this rule is that it avoids oscillations between the processes of a sort regulated independantly by two predecessors. Oscillations occur because each predecessor can independantly activate the target component when it is active, but if one component is inhibited, it will inhibit the target component. This competition between the two predecessors generates the 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. $ab_{00} \rightarrow c_2 \upharpoonright c_1$, $ab_{00} \rightarrow c_1 \upharpoonright c_0$) while a and b activate c through the cooperative sort ab (e.g. $ab_{11} \rightarrow c_0 \upharpoonright c_1$ $ab_{11} \rightarrow c_1 \upharpoonright c_2$). Indeed, the reachability of c_2 and c_0 is conditioned by a cooperation of a and b as explained above.

C. Model construction (from RSTC to PH)

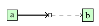
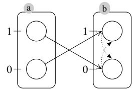
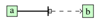
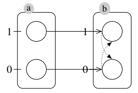
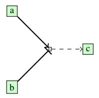
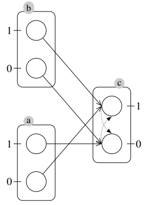
In this work we aim to model a biological system according to its dynamic. For that purpose we choose to build a PH model from the biological system reprensented as an RSTC network.

In the following we present our automatic approach to generate a PH model from an RSTC network.

1) 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 propose their PH representation as illustrated in Algorithm 1.

Algorithm 1 uses two procedures, detectPattern (see Algorithm 2) that automatically browses the graph node by node and detects all 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 arrive to that node and the type of input nodes of the pattern. Consequently, Algorithm 2 returns the pattern and its type to another procedure (Algorithm 3) which translates the pattern into the PH formalism. This transformation takes care of different cases (cooperation, synchronization, simple activation, simple inhibition, etc.)

TABLE I. EXAMPLES OF PATTERNS

Biological Patterns	**PH Transformations**
Simple activation  	
Simple inhibition  	
activation or inhibition  	

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 $Net$
for all Node $n$ in $Net$.getSetOfNodes() do
    $Pat$ = detectPattern ($Net$, $n$)
    patternInPHModel ($out$, $Pat$)
end for
```

For example a molecule a cooperatting 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 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 I.

2) Estimating the parameters for the PH-simulation from time-series gene expression data: The simulation of the execution of the PH actions is done stochastically. Therefore, we need to relate each action with temporal and stochastic parameters introduced into the PH framework to achieve dynamic refinement [13]. This is an important aspect of the modeling when taking into account the temporal and stochastic dimensions of biological reactions by performing simulations.

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.
switch ($n$)
case TerminalNode:
    add node n to the pattern $Pat$
    numberPredecessor= $n$.getNumberOfPredecessor()
    switch (numberPredecessor)
    case 1:
        for all $p$ in setOfPredecessor (n) do
            switch ($p$)
            case TerminalNode:
                add node n to the pattern $Pat$
            case TransientNode:
                detectPattern ($Net$, $p$);
            end switch
        end for
        Set the code of pattern $Pat$;
        return $Pat$;
    case 2:

    end switch
case TransientNode:
    numberPredecessor= $n$.getNumberOfPredecessor()
    switch (numberPredecessor)
    case 1:
        for all $p$ in setOfPredecessor (n) do
            switch ($p$)
            case TerminalNode:
                added node to the pattern $Pat$;
            case TransientNode:
                detectPattern ($Net$, $p$);
            end switch
        end for
        Set the code of pattern $Pat$;
        return $Pat$;
    case 2:

    end switch
end switch
```

On the one hand, we consider the probability of a reaction to occur, and on the other hand, we consider stochastic parameters in the aim at observing an expected behavior. In the PH framework to play an action we need two essential parameters: the rate r or the temporal parameter because $t = r^{-1}$ if we assume that t is the average time of playing an action and the stochasticity absorption factor sa. The stochasticity absorption factor is introduced to control the higher variance around the mean duration of playing an action. These two parameters are estimated according to the expression profile of time-series data of the experiment described in Section II-A2.

3) From data to action parameters: For the model components which have a measurement in the Time Series Data we estimate and integrate r and sa parameters in the PH model. For others, we assign default parameters. In other 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 in which we assume that a component moves from one process to another. Therefore the action that leads this change must be played with 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, the small the variance around r is.

4) Discretization of times-series data: Because PH simulation is discrete, we need to discretize continuous experimental

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 correspondant 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
```

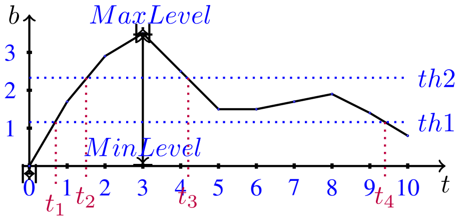


Fig. 5. **Illustration of the estimation of temporal parameters from time series data: $r_i = \frac{1}{t_i - t_{i-1}}$. $MaxLevel$ represents the maximum expression. $MinLevel$ represents the minimum expression. In this example 0 is the minimum level. $th1$ and $th2$ represent respectively the first and the second threshold.**

data so we can compare the simulation outputs. The goal of this method is to better determine, according to the gene expression level, when a given molecule is activated or inhibited. To do this we introduce the new analog concept of Significant Increase or Decrease to characterize the fact that a level of a molecule increases or decreases when crossing a threshold of significance; we limite the possible expression levels for a molecule to $\{0, 1, 2\}$. This choice was made because when we look at time-series data (see Fig. 3), one can clearly observe a high level of activity between $0h$ and $5h$ and a slow level of activity between $5h$ and $10h$ on the other hand. The goal of the discretization method is to capture these two activation times. For each time-series, we introduce two thresholds *th1* and *th2* as in Fig. 5. The value of this two thresholds is computed as follows $\mathit{th1} = \frac{1}{3}(MaxLevel - MinLevel)$ and $\mathit{th2} = \frac{2}{3}(MaxLevel - MinLevel)$. Therefore, expression level in the range $[0 - th1]$ is at level 0, while expression level in the range $[th1 - th2]$ has level 1 and the last range is at level 2. Thus we can automatically determine the different levels of expression of the TSD. To illustrate the result of the discretization algorithm we plot in Fig. 6 the expression of 9 selected genes from the TSD with their respective discrete plots.

D. Simulation: Initials conditions

To run the simulation we need to set initial conditions for the components of our network. Because the components of the network are grouped into layers, the initial conditions will be the same in the different layers. In the following we will present the initial conditions that we have chosen for the different components.

- **Receptor layer: E-cadherin.** We choose the pulse signal for the input node E-cadherin, that is active for a duration of 5 units times in average. We made this choice to take into account the average time of the Calcium stimuli effect.
- **Signaling layer: signaling proteins.** At this layer, the components are activated and inhibited with the same rate and the same stochasticity absorption factor. Nevertheless, the choice of inhibition parameters must ensure that the time interval in which the inhibition action is firing is greater than the time interval of firing activation action on the same component. Moreover, it should not be any overlap between the two time intervals. These two conditions can be seen as reachability constraints from the entry node (E-cadherin) to the output node genes. The values of these parameters are selected by considering the delay of signal transduction from the entry node (E-cadherin) to the output node (genes).
- **Transcription layer: transcription factors.** At this stage to model activation over a transcription factor, the signal comes from signaling proteins; however, for inhibition, it additionally come from an auto-inhibition which represents the degration of the transcription factor. The signal comes from signaling proteins for activation. But for the inhibition, in addition to the signal come from signaling proteins, we introduce an auto-inhibition which representes the degradation of the transcription factor.

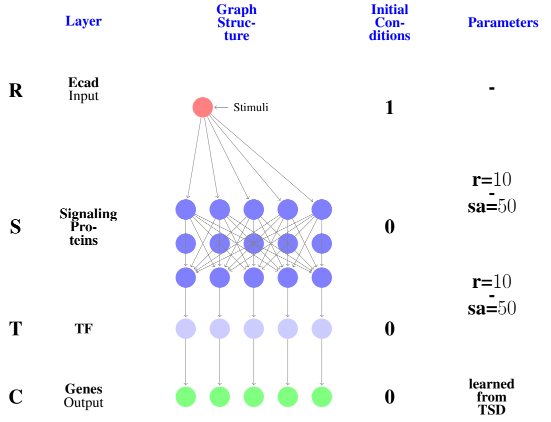


Fig. 7. **RSTC network structure and initial conditions assigned to each node in the layer**

- **Genes.** The genes are activited or inhibited according to the estimated values from time-series data.

The initial conditions of the simulation are summarized in Fig. 7 where we can clearly see the initial values chosen at each layer for our model.

III. Results

A. From PID to PH: An automatic generation of PH code of the network

To simulate of the model we generated a PINT code to be simulated by the PINT simulator [1]. PINT implements static analyses for computing dynamical properties on very large-scale Automata Networks. For the PINT code generation we use two procedures as described in the method section. The first procedure that detects the patterns and its code; the second procedure that generates the PINT code by taking in consideration the type of the pattern to better refine the dynamic of the system from a structural point of view. This refinement is done by introducing the synchronization sort which is a generalization of the cooperation sort. This allows to avoid artificial oscillations in the dynamics of the components of the system.

B. Simulation

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

1) Without the introduction of synchronization sort: One can easily notice in Fig. 8 the occurence of oscillations. This is not the expected behavior from the biological system but it is coherent with the choice of the modeling and the way the simulator works as explained in II-B2: cooperation sorts are used to model multiple regulators of a common target where this is clearly identifiable. In other cases we left the components act independently. It is important to notice that the intensity of the oscillation is linked with the size of the concurrence, i.e. the number of predecessors that a node in the network has. Despite the presence of the oscillations, the

[1] Available at http://process.hitting.free.fr

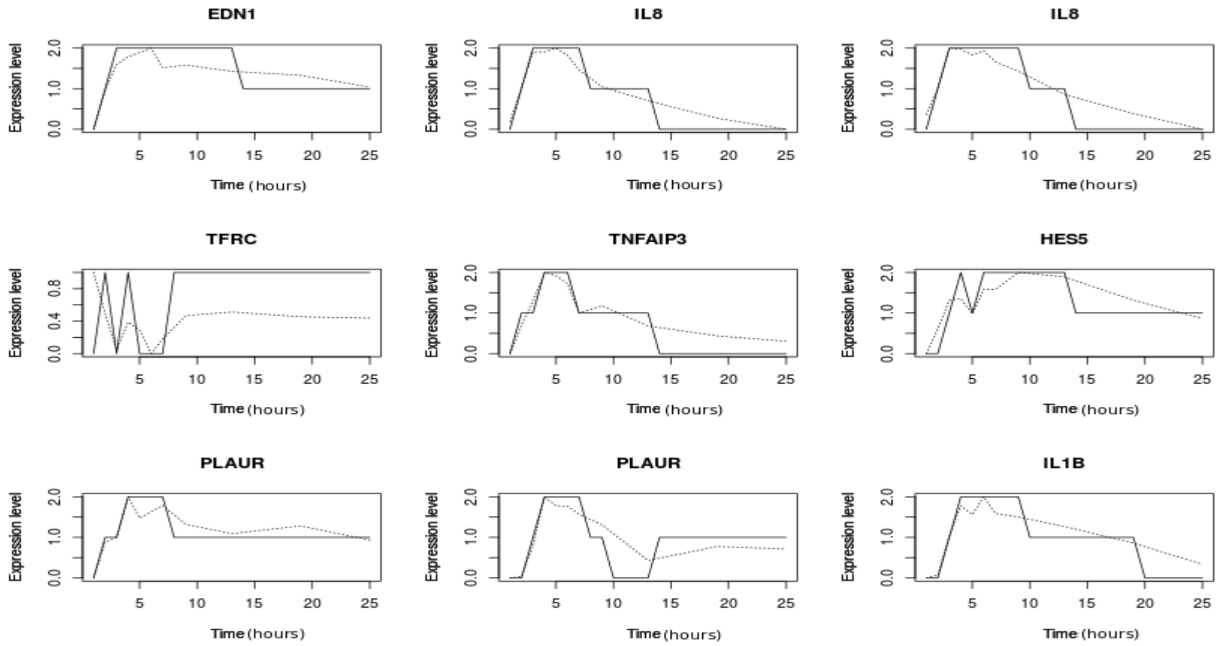


Fig. 6. **Illustration of discretization of experimental data. Dashed lines are continuous real TSD, while continous lines are discrete data**

model reproduces expected dynamical behavoirs namely the dynamic of components, the signal transduction and takes into account the stochastic and time aspect of the model.

2) With the introduction of synchronization sort: From Fig. 9 we can see that the introduction of the synchronization sort significatively reduces the impact of concurrence by the introduction of the synchronization. The result shows a clear elimination of the previously observed oscillation in Fig. 8. One can see that the expression levels of genes *uPAR*, *MKP1*, *IL1 beta* are well reproduced, while *Hes5* and *IL8* are not activated. This result can be observed when the activation signal has not been able to propagate through the network due to the non determinism and concurrency.

IV. 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 stochastic pi-calculus PH model from a biological system composed of biochemical reactions, extracted automatically from public databases, relevant to keratinocyte differentiation induced by Calcium. We then 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. Finally we described 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. Ours results show that: (1) we reproduce accuretely 7 out of 12 of the genes, (2) we can model large-scale networks in which signal goes from an input node to the output nodes, (3) we can reproduce tendences of the dynamic of components and take into accout the stochastic and time aspect of the behaviors of the biological system. As concrete perspectives of this work we intend to *(i)* validate the RSTC network topology by confronting its *in-silico* simulation with real measurements of its components; *(ii)* compare the stochastic simulation results with reachability static analyses over the same PH components mapped to the 12 measured genes; and finally *(iii)* search for key-regulators up-stream the 12 genes which will control the dynamics of the system, to provide concrete hypotheses to test experimentally.

References

[1] C. F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow, "Pid: the pathway interaction database," *Nucleic acids research*, vol. 37, no. suppl 1, pp. D674–D679, 2009.

[2] N. Yu, J. Seo, K. Rho, Y. Jang, J. Park, W. K. Kim, and S. Lee, "hipathdb: a human-integrated pathway database with facile visualization," *Nucleic acids research*, vol. 40, no. D1, pp. D797–D802, 2012.

[3] J. J. Tyson, K. C. Chen, and B. Novak, "Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell," *Current opinion in cell biology*, vol. 15, no. 2, pp. 221–231, 2003.

[4] G. Batt, D. Ropers, H. De Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider, "Validation of qualitative models of genetic regulatory networks by model checking: Analysis of the nutritional stress response in escherichia coli," *Bioinformatics*, vol. 21, no. suppl 1, pp. i19–i28, 2005.

[5] M. Mobashir, B. Schraven, and T. Beyer, "Simulated evolution of signal transduction networks," *PloS one*, vol. 7, no. 12, p. e50905, 2012.

[6] C. Guziolowski, S. Videla, F. Eduati, S. Thiele, T. Cokelaer, A. Siegel, and J. Saez-Rodriguez, "Exhaustively characterizing feasible logic models of a signaling network using answer set programming," *Bioinformatics*, vol. 29, no. 18, pp. 2320–2326, 2013.

[7] A. Mitsos, I. N. Melas, P. Siminelakis, A. D. Chairakaki, J. Saez-Rodriguez, and L. G. Alexopoulos, "Identifying drug effects via pathway alterations using an integer linear programming optimization formulation on phosphoproteomic data," *PLoS computational biology*, vol. 5, no. 12, p. e1000591, 2009.

[8] A. MacNamara, C. Terfve, D. Henriques, B. P. Bernabé, and J. Saez-Rodriguez, "State–time spectrum of signal transduction logic models," *Physical Biology*, vol. 9, no. 4, p. 045003, 2012.

[9] M. K. Molloy, "Performance analysis using stochastic petri nets," *Computers, IEEE Transactions on*, vol. 100, no. 9, pp. 913–917, 1982.

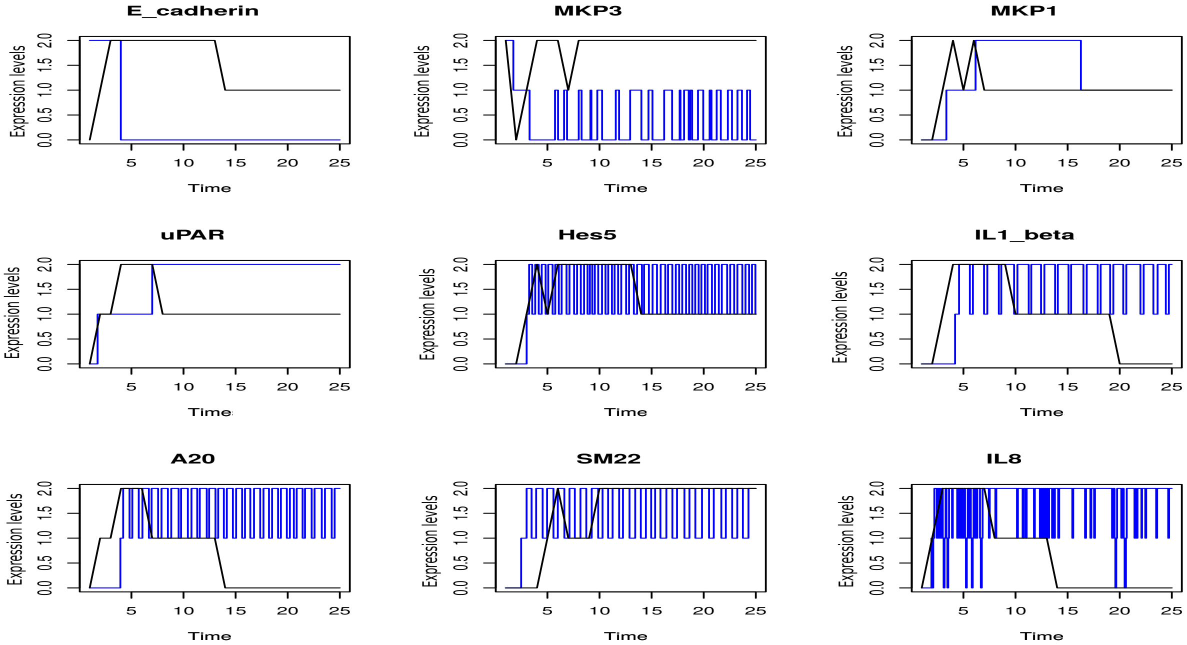


Fig. 8. **Results of simulations without introducing the synchronization sort. In black line the expected behaviors come from the discretization of time-series data. In Blue line the simulation behavior.**

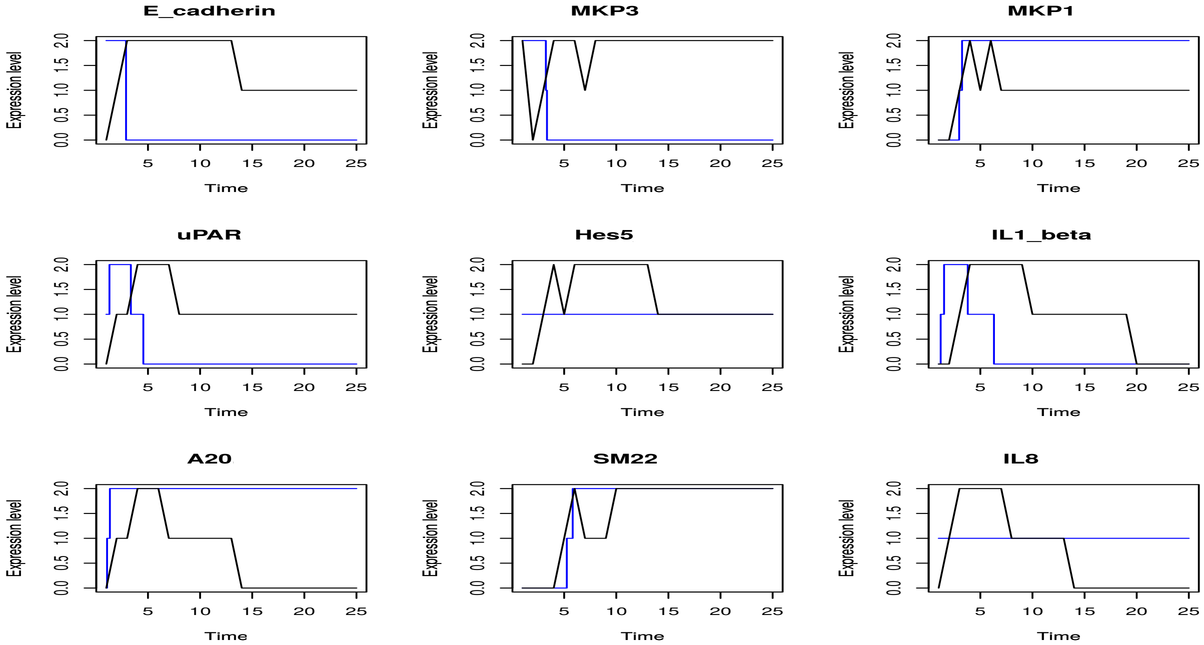


Fig. 9. **Results of simulations by introducing the synchronization sort. In black line the expected behaviors come from the discretization of time-series data. In Blue line the simulation behavior.**

[10] M. Heiner, D. Gilbert, and R. Donaldson, "Petri nets for systems and synthetic biology," in *Formal methods for computational systems biology*. Springer, 2008, pp. 215–264.

[11] C. Priami, "Stochastic π-calculus," *The Computer Journal*, vol. 38, no. 7, pp. 578–589, 1995.

[12] M. Maurin, M. Magnin, and O. Roux, "Modeling of genetic regulatory network in stochastic π-calculus," in *Bioinformatics and Computational Biology*. Springer, 2009, pp. 282–294.

[13] L. Paulevé, M. Magnin, and O. Roux, "Refining dynamics of gene regulatory networks in a stochastic π-calculus framework," in *Transactions on Computational Systems Biology XIII*. Springer, 2011, pp. 171–191.

[14] C. Guziolowski, A. Kittas, F. Dittmann, and N. Grabe, "Automatic generation of causal networks linking growth factor stimuli to functional cell state changes," *FEBS Journal*, vol. 279, no. 18, pp. 3462–3474, 2012.