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

***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. This choice on one hand limits us to discrete data, but on the other, 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 the calcium stimulation of the E-cadherin protein.**

This work has the following contributions. First, we built an interaction graph linking a signaling molecule, E-cadherin, 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 method proposed in. Second, we proposed an automatic transformation of selected known biological patterns present in PID in order to generate PH modules. Additionally, we added necessary constraints to the PH model to avoid biologically incoherent dynamical behaviors. Third, we proposed a way of estimating temporal and stochastic parameters from time-series expression data to model the measured genes. These parameters were used for the stochastic simulation of the model. Finally, we discretized the experimental data to allow the comparison with simulation results.

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. Moreover, we generated dynamical predictions, consquences of the time-series data, on key cellular processes for our case study.

Keywords*—*Model construction, data integration, parameters estimation, large-scale network.

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 stochastic approaches, 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 formalisme have been introduce. stochastic Petri Nets(SPN) are one approach suitable 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 formalisme introduce by [11] and use by [12] for the modeling of biological systems. Stochastic pi-calculus have a good expressivity and are well adapted pour the use of compositionnal approach.

For modeling and analyzing the biological system we rely on 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 pa-

rameter 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 methodological and preliminar results of this work are: (i) automatic PH generation from a biological system composed of biochemical reactions, and extracted from public databases; (ii) discretization approach of time-series expression data, so we can reproduce these traces by using in a first attempt the PH stochastic simulation, and afterwards perform static reachability analyses to satisfy this data; and (iii) estimation of the the temporal and stochastic parameters of the simulation, based on statistical analyses of the full-compendium of time-series expression data. The biological system used as a case-study for this work is a cell-based model of skin differentiation, which is of key importance in wound healing.

March 20, 2015

II. Data and Methods

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

- **Formalization of biological model.** Building a computational hybrid model from biological network by automatic detection of know biological patterns.
- **Temporals 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 network: 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 were: (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.

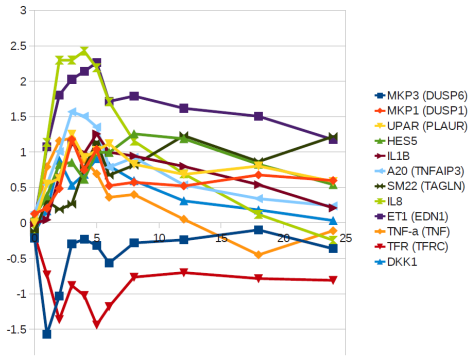


Fig. 3. **Plot of 12 selected genes**

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.*
- $E = \{e_1, e_{,\ldots,e_m}\}$ *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 state, biological process and positive condition. On the other side, transient nodes can be transcriptions, translocation, modification and compound. Edges are of different types. We have activation(agent), inhibition, output, input and famillyMemberOf.

2) Time-series microarray data description: To illustrate our approach, we used the time-series microarray data from Calcium stimulated keratinocyte cells measured at 10 time-points. A 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 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 it's currently in the process of getting 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 $a_i \rightarrow b_j \nearrow b_k$, 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.

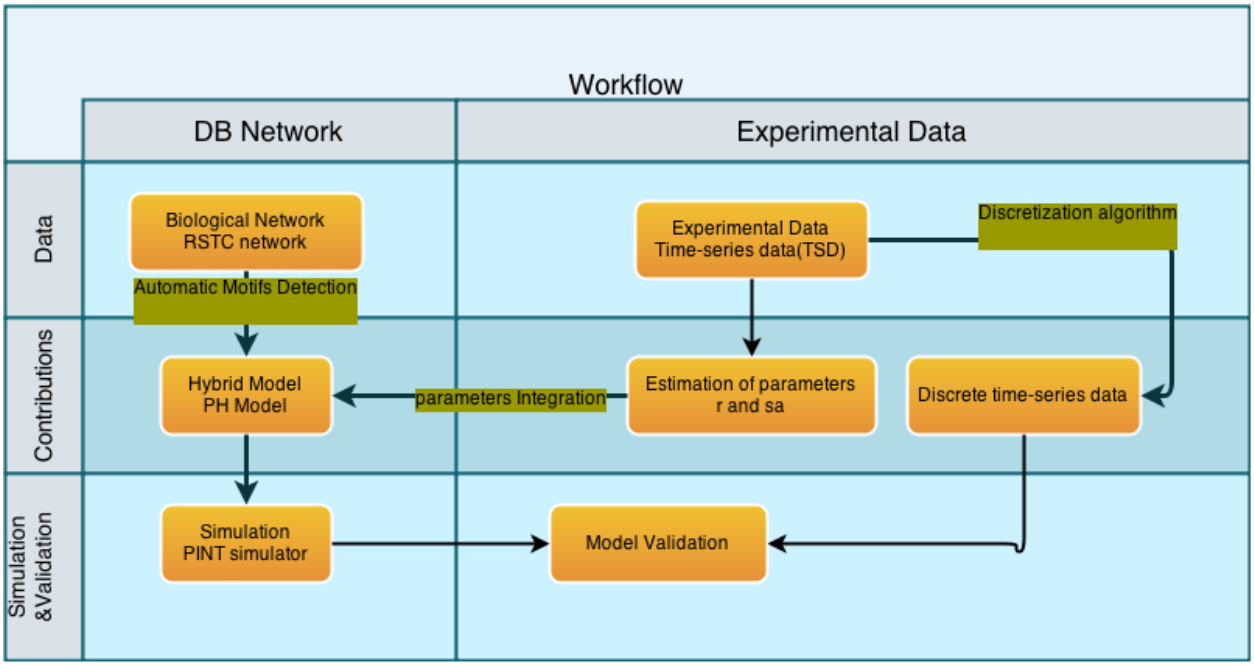


Fig. 1. **Work-flow of our method**

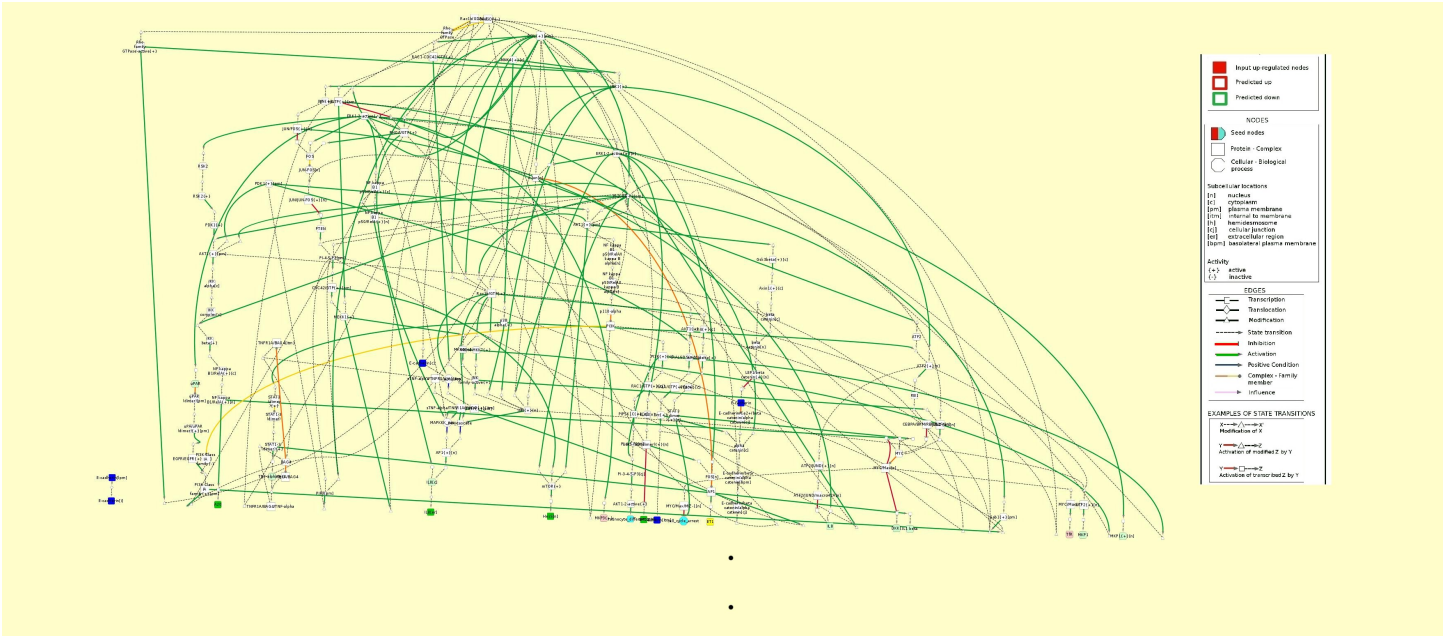


Fig. 2. **RSTC network**

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* $a \neq b \Rightarrow L_a \cap L_b = \emptyset$*;*
- $\mathcal{H} = \{a_i \rightarrow b_j \Rsh b_k \in L_a \times L_b \times L_b \mid (a, b) \in \Sigma^2 \wedge b_j \neq b_k \wedge a = b \Rightarrow a_i = b_j\}$ *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 the play of the action h in s, with $(s \cdot h)[b] = b_k$ and $\forall c \in \Sigma, c \neq b, (s \cdot h)[c] = s[c]$.

a) Modeling cooperation & synchronization.: 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).

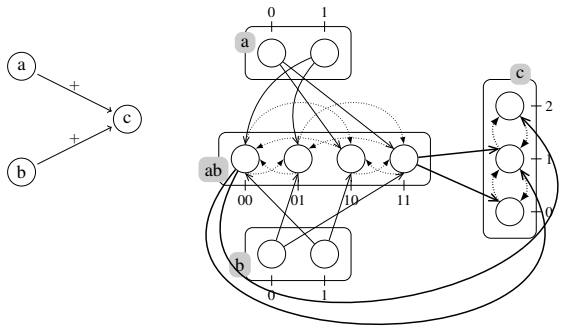


Fig. 4. (left) Biological pattern example. Nodes are components and edges are interactions For instance, components a and b cooperate to activate c. (right) equivalent PH model. A PH example with four sorts: three components (a, b and c) and a cooperative sort (ab). Actions targeting processes of c are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

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., $a_1 \rightarrow ab_{00} \Rsh ab_{10}$ 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. Unlike cooperation sort which allow us to model the fact that two components cooperate to hit another component, we introduce the notion of synchronization sort, which will implement another type of cooperation. If we refer to our example Fig. 4, and asume that ab is our *synchronization sort* between sorts a and b, defined with also 4 processes. Then, component c will be activated(c_1 bounce to c_2) if each one of a and b are also activated. So each one of this processes ab_1, ab_2, ab_3 can activate c_1. But to be desactived, we need both a and b to be desactived. We'll see later why it's important to introduce this new rule for the cooperation between componnets.

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.$ab_{00} \rightarrow c_2 \Rsh c_1$, $ab_{00} \rightarrow c_1 \Rsh c_0$) while a and b activate c through the cooperative sort ab (e.g. $ab_{11} \rightarrow c_0 \Rsh c_1$ $ab_{11} \rightarrow c_1 \Rsh c_2$). Indeed, the reachability of c_2 and c_0 is conditioned by a cooperation of a and b, as explained above.

C. From RSTC to PH/Model construction

In this work, we aim to see a biological model according to his dynamic. For that, we choose to build a formal model from a biological model reprensented as in an RSTC network.

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

Require: Net
Ensure: return all the patterns associated to the given network Net

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

Fig. 5. **Pattern detection in an RSTC Network**

Require: Net, n
Ensure: Detect a pattern associated to a given node (n in this case)

```
switch (n)
case TerminalNode:
    added node to the pattern
    numberPredecessor= n.getNumberOfPredecessor() {To get the number of predecessor of node n}
    switch (numberPredecessor)
    case 1:
        for all in comming edge (n-m) do
            get the node m
            switch (m)
            case TerminalNode:
                added node to the pattern Pat
                Set the code of pattern Pat
                return Pat
            end switch
        end for
    end switch
end switch
```

Fig. 6. **Detection of Pattern associated to the node n**

1) Modeling the RSTC network as a PH model: In order to model the RSTC network as a PH model we selected known biological regulatory patterns (atomic set of biological components and their interacting roles), represented as biochemical reactions in the RSTC network, and proposed their PH representation.

We build an algorithm that will automatically browse the graph node by node and detect all patterns in the graph. More precisely, for each node(output node of the pattern) we will call a recursive procedure, that will allow us to detect a minimal set of node(input node of the pattern) that has an direct influence on that node. This set of node plus the output node and the way there are linked are a pattern for us. The type of a pattern is determine by the type of the output node, the type of regulations come on that node and the type of input nodes of the pattern. So the algorithm that detect patterns return the pattern and its type to another procedure which one will traduce the pattern into the process hitting formalism. This transformation will take care of different case(cooperation, synchronization, simple activation, simple inhibition,...)

For example a molecule a that cooperates with a molecule b to activate a molecule c Fig. 4 (top), 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 (bottom) a, b, c and ab. Sorts a, b and c represent

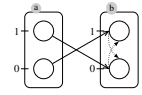
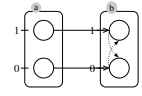
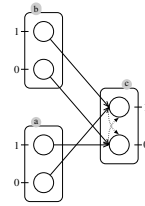
Require: out, Pat { Pat is The pattern to be translate into the PH Model, out is the output file}
Ensure: The correspondant PH Model of the given pattern Pat will write into the file out

```
nocp = Pat.getNumberOfComponents() {Number of the components of the pattern Pat}

switch (nocp)
case 2:
    switch (type)
    case A:
        out.write("activation")
    case I:
    out.write("inhibition")
    default:
        out.write("unknow pattern")
    end switch
case 3:
    switch (type)
    case C:
        out.write("cooperation")
    case S:
    out.write("synchronization")
    default:
        out.write("unknow pattern")
    end switch
end switch
```

Fig. 7. **Pattern in PH Model**

TABLE I. EXAMPLES OF PATTERNS

Biological Patterns	PH Transformations
Simple activation	
Simple inhibition	
activation or inhibition	

components a, b and c. We introduce the cooperative sort ab to characterize constraints on components a and b. In our RSTC network, we found 11(to be precise...) regulatory patterns (see Appendix 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

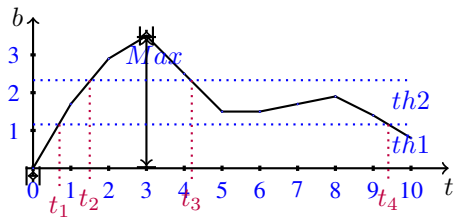


Fig. 8. Illustration of estimation of temporal parameters:$r_i = \frac{1}{t_i - t_{i-1}}$

modeling when taking into account the temporal and stochastic dimensions of biological reactions by performing simulations. 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 sa. These two parameters will be estimated according to the expression profile of time-series data of the experiment describe in II-A2.

To avoid over fitting in the estimation of these parameters, we propose that each component of the PH, representing a measured gene in the network, will take the estimated values of the parameters of its respective cluster in the experimental data.

1) The first step is to cluster the data set. The goal of the clustering process is to partition genes into groups such that the profiles contained in the same group (cluster) are similar to each other and as different as possible from the profiles assigned to the other clusters. The particularity here is to choose the best clustering criteria.
2) For each cluster obtained in the previous step, estimate the value of r and sa associated to the cluster.
3) For each component of the PH model associated to the measured gene, determine its cluster, and assign it r and sa, the previously estimated parameters.

In our time-series data the components of the PH which need to be associated specific parameters (step 3) are the 12 genes present in our RSTC network.

3) From data to action parameters: Before this work, it were not possible to infer temporal parameters from TSD and integrate them automatically into a model build in to the process hitting formalism. It's now possible. For components that we have measured, we know how to estimate and integrate parameters. For others that we don't have measured, we can give default parameters.

4) Discretization of times-series data: Because PH simulation is discrete we need to discretize continuous experimental data, so we can compare our simulation outputs. The goal of this method was to better determine, according to the gene expression level, when a given molecule is activated or inhibited. To do this, we introduced 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 limited the possible expression

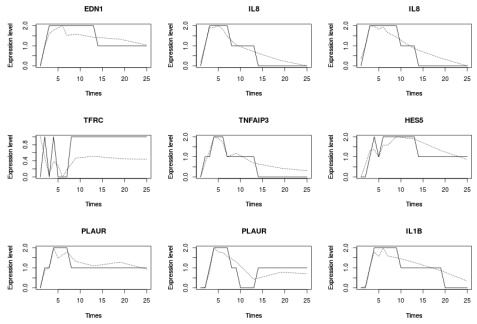


Fig. 9. Illustration of discretization of Experiment Data

levels for a molecule to $\{0, 1, 2\}$. We made this choice because when we look our time-series data (see 3), we can clearly observe a high level of activity between $0h$ and $5h$ on one hand, and a slow level of activities between $5h$ and $10h$ on the other hand. In our discretization method, our goal is to capture this two times of activities. For achieve this goal, we propose the following discretization method. For each time-series, we introduce two thresholds *th1* and *th2* as in figure 8. The value of this two thresholds is compute has follow $th1 = \frac{1}{3}(MaxLevel - MinLevel)$ and $th2 = \frac{2}{3}(MaxLevel - MinLevel)$. Therefore, all the expression level of a time-series between $[0 - th1]$ will have level 0, all those between $[th1 - th2]$ will have level 1 and the last group will have level 2. By this way, we can automatically determine the differents level of expression of all our time-series data. To illustrate the result of our discretization algorithm, we plot in Fig. 9 the expression of 9 seleted genes from the times-series data with their respective discrete plots.

D. Simulation: Initials conditions

To better understand the results of simulation, we need to give initial conditions for the components of our network. Because the components of the network are group 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 will be active for a duration of average x units times. We made this choise to take into account the average time of calcium stimuli effect.
- **Signaling layer: signaling proteins.** At this layer, the components will be activated and inhibited according to the same rate and the same stochasticity absorption factor. The values of this parameters where selected by considering the time of signal transduction from the entry node(E-cadherin) to the output node (genes).
- **Transcription layer: Transcription factors.** At this stage, the signal will come from signaling proteins for activation. But for the inhibition, in addition of the signal come from signaling proteins, we introduce an auto-inhibition which will represented a degradation of the transcription factor.

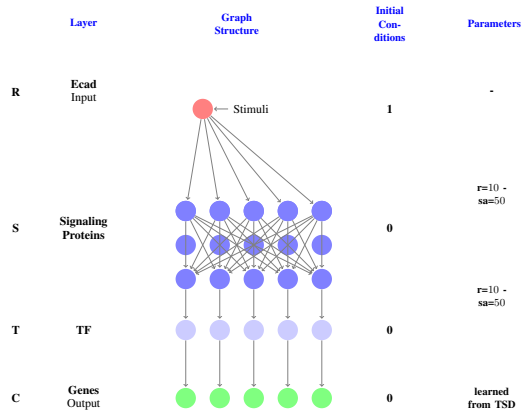


Fig. 10. Initial conditions

- **Cell-state: Gene and cellular process.** The genes will be activited or inhibited according to the estimated values from time-series data.

We summarise the initial conditions of our simulation in figure 10 were we can clearly see the initial values choose at each layer for our model.

III. Results

A. From PID to PH: On 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]. For the PINT code generation we use two procedures as describe in the method section. The first procedure that detect the patterns and his code. Then the second procedure that generate the PINT code by taking in consideration the type of the pattern to better refine the dynamic of the system on a structural point of view. This refinement is done by introducing the synchronization sort which is a generalization of the cooperation sort. this allow us to avoid artificial oscillation in the dynamic of the components of our system.

B. Simulation experience

In this section we'll present the results of the simulations of the model. It will be present in two main sub sections. In the first, we'll present the result of simulation of model without the inclusion of the synchronization gate and in the second we will present result with synchronization gate. Both results present common characteristics and show us the coherence of our approach of modelization. The common characteristics are the following:

- We can modeled large scale network in which signal goes from an input node to the output nodes.
- We can reproduce tendances of the dynamic of components as illustrate in 11 and 12.
- The model take in accout the stochastic and time aspect of the bihaviours of the biological system.

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

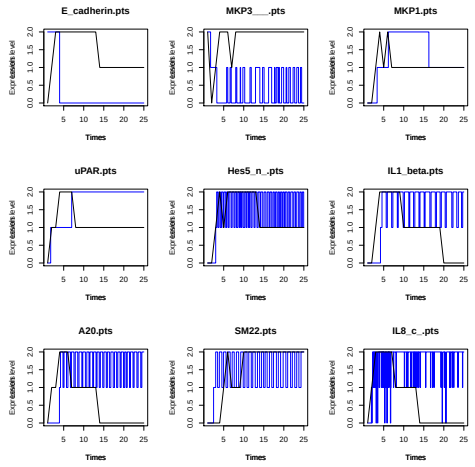


Fig. 11. **Results of simulations without introducing the synchronization gate. In black line the expect behaviours come from the discretization of time-series data. In Blue line the simulation behavior.**

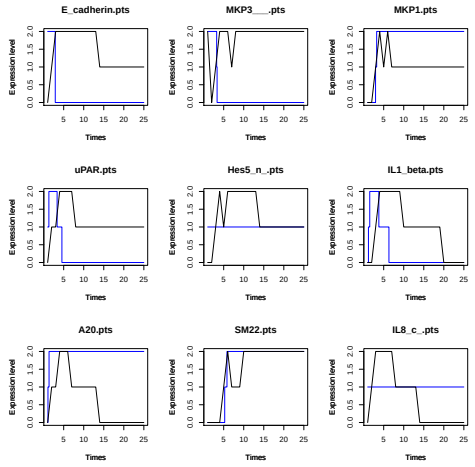


Fig. 12. **Results of simulation by introducing the synchronization gate. In black line the expect behaviours come from the discretization of time-series data. In Blue line the simulation behavior.**

1) Without the introduction of synchronization gate: The main important remark that we can made in this result 11 is the occurence of oscillations. This is not the expect result but it's coherent with the simulator properties. It's important to note that the intensity of the oscillation is link with the size of the concurence. Despite the presence of the oscillation, the model also reproduce the same tendances namely the dynamic of components, the signal transduction and the taking into acount the stochastic and time aspect of the model.

2) With the introduction of synchronization gate: On the contrary of the previous result, figure 12 show us that with the introduction of the synchronization gate, we can reduce the impact of concurence by the introduction of the synchronization and the cooperation gates. The result show us a clearly elimination of the observed oscillation in figure 11.

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 PH from a biological system composed of biochemical reactions, extracted automatically from public databases, relevant to keratinocyte stimulation induced by Calcium. We then proposed a method to discretize time-series gene expression data, so they can be confronted to the PH simulations and logically explained by the PH static 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. 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 analysis 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 our biological partners 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.

[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.