Toward Integration of Systems Biology Formalism: The Gene Regulatory Networks Case

Raffaella Gentilini

gentilin@dimi.uniud.it

Università di Udine (DIMI), Via Le Scienze 206, 33100 Udine, Italy

Abstract

We consider the problem of integrating different systems biology formalisms, namely, the process calculi based formalism, the modeling approach based on systems of differential equations, and the one relying on automata-like descriptions (and model checking).

Specifically, we define automatic procedures for translating stochastic π-calculus descriptions of *gene regulatory networks* to *S-systems* differential equations. Tools for extracting and reasoning on (approximate) solutions of S-systems have been recently developed in the literature, and can be exploited to establish a link with automata-based systems biology and model checking techniques.

Keywords: systems biology, gene regulatory networks, S-systems, process calculi

1 Introduction

The attempt to face the many challenges coming from the Biology Realm (with its intrinsic complexity) have recently produced a number of promising converging points of apparently different sciences. *Systems Biology* [11] is probably one of the most important emerging research fields in this context, having as a major objective that of developing adherent formal methods to describe, reason, and predict on the evolution of general biological systems. First steps in this direction resulted in the attempt of applying, with some success, existing formalisms originated within various subfields of mathematics, logic, and computer science. In this paper we focus on the problem of both interfacing and integrating some of these formalisms. In particular, we deal with three major modeling approaches: the traditional approach, suggested by mathematicians, based on systems of differential equations, the recently developed approach [17, 18] relying on process calculi, and methods concerning the use of model checking and (hybrid) automata theory [1, 2, 4].

Each of the above mentioned approaches has its own merits and, in our opinion, no one represents *the* solution. The mathematical depth underlying quantitative descriptions based on systems of differential equations, the naturalness of process calculi in describing the behavior of complex systems of communicating objects, and the ability to describe and querying time evolving systems provided by automata, are the basic motivations for our attempt at integrating the three approaches.

Here we begin our work building a (preliminary) link relating the formalisms discussed, by developing procedures to translate, automatically, stochastic π-calculus [14] descriptions of gene regulation networks (as described in [6]) to/from *S-systems* differential equations systems. S-systems are specific kinds of differential equations systems having a simple canonical form, largely used in the context of extracting gene networks models from time-series experiments [16, 19, 21], and suitable to adherently describe biological pathways and networks [21]. On the other hand, S-system are at the ground of the development of (hybrid) XS-systems [2, 4], which are a kind of (hybrid) automata built from S-systems approximate solutions. Tools for *reasoning* and *querying* (hybrid) XS-systems have been realized in [2, 4]: thus, automatic procedures converting S-systems to process calculi descriptions establish, as a byproduct, a further link with automata-theory based formalisms. Motivations similar to

ours pursued the recent work in [10], dealing with translation of process algebra models of signalling pathways to ODE.

The rest of the paper is organized as follows: in Section 2 we introduce the systems biology formalisms that we would like to interface. In Section 3 we define automatic procedures to mutually translate and integrate such formalisms; the latter are applied to the repressilator network [8] in Section 4. Finally, we conclude the paper discussing, in Section 5, further motivations underlying our work.

2 Systems Biology Formalisms

2.1 ODE and S-Systems

Traditionally, the effort of mathematicians approaching the field of biological research, has been primely focused on designing formal models to assist biologists in endowing experimental data of some *mathematical interpretation*. Systems of differential equations are considered, in this context, a very suitable formal tool to accomplish the task of capturing the nature of biological phenomena. To date, the joint work of biologists and mathematicians effectively resulted in deeply adherent ordinary differential equations (ODE) systems, describing many complex pathways and biological processes [5]; on the other hand, a similar effort pursued the problem of providing formal tools both more *flexible* and suitable for interacting with the user, as well as being readily understandable for the biologists'community.

The development and study of *S-systems* [15, 20] goes exactly in the direction of developing classes of (dynamical) models for biochemical pathways characterized by a good compromise between accuracy and mathematical flexibility.

Definition 2.1 ([15]) *An* S-system *is a system of differential equations of the form*

$$\dot{X}_i = V_i^+ - V_i^- = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}} \quad (i = 1 \dots n)$$

where $DV = \{X_1, \dots, X_n\}$ *are the* dependent variables, $IV = \{X_{n+1}, \dots, X_{n+m}\}$ *are the independent variables,* $\{\alpha_i\}$ *and* $\{\beta_i\}$ *are the* production *and* degradation *rate constants, and* $\{g_{ij}\}, \{h_{ij}\}$ *are the* apparent kinetic orders*.*

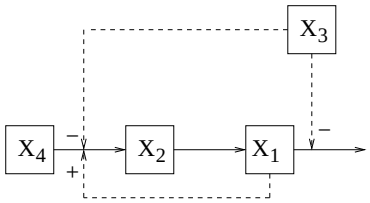
S-systems[1] are at the ground of the approaches proposed in [2, 4, 21], all aiming at setting tools for the computational analysis and the prediction of biochemical systems dynamics. In particular, Voit extensively discusses in [21] how S-systems are *canonical* dynamical systems suitable to describe with accuracy a large class of biochemical pathways: the development of techniques for *automatically* defining S-systems from biochemical maps is addressed (see example 2.2, below) and a tool for deriving (approximate) solutions of S-systems is developed.

Example 2.2 *Consider the (toy) biochemical map depicted below. In such an example (see [21], Chapter 3),* X_2 *represents the product of a reaction that uses* X_4 *as a substrate, is positively activated by* X_1 *and inhibited by* X_3 *(feedback dotted arrows). In turn, product* X_2 *is converted into product* X_1 *and its degradation is inhibited by* X_3*. Hence, the corresponding S-system has two dependent variables (*$DV = (X_1, X_2)$*) and two independent variables (*$IV = (X_3, X_4)$*). The differential equations for dependent variables are:*

[1] In [2, 4, 21] S-systems are augmented also with a set of *algebraic constraints* expressing conditions that must be additionally satisfied for the system to obey conservation of mass, stochiometric relations, etc.

$$\dot{X}_1 = \alpha_1 X_2^{g_{12}} - \beta_1 X_1^{h_{11}} X_3^{h_{13}}$$
$$\dot{X}_2 = \alpha_2 X_1^{g_{21}} X_3^{g_{23}} X_4^{g_{24}} - \beta_2 X_2^{h_{22}}$$

$$where \quad g_{12}, g_{21}, h_{11}, g_{24}, h_{22} > 0 \quad and \quad h_{13}, g_{23} < 0$$



In fact, variable X_3 affects (negatively) both degradation of product X_1 and production of X_2 (hence it appears in the terms V_1^- and V_1^- with a negative kinetic order). In turn, X_1 enhance production of X_2 (hence, it appears with positive kinetic order in V_2^+ term). We refer to [21] for an insight discussion on both suitability of "power-laws" forms of terms V^+, V^- in S-systems, and adherence of S-systems derived from maps with the above sketched procedure.

2.2 Automata-Based Systems Biology

In [3, 4] the issue of reasoning, querying, and predicting on the dynamics of biological pathways is addressed, combining automata theory and S-systems descriptions. The basic idea underlying the work in [3, 4] is to associate an S-system, $\mathcal{S}$, with a finite automata obtained by suitably encoding a set of $\mathcal{S}$ traces[2]. Fixed a time-step, Δ, each set of dependent variable values, corresponding to a given trace of $\mathcal{S}$ after k steps (i.e. at time $k\Delta$), is encoded in a state of the final automaton. Model Checking [7] techniques are then exploited to automatically analyze, in silico, the automata encoded traces. Hence, qualitative queries as *Is it possible to reach a steady-state?*, but also quantitative ones, are formulated using a suitable temporal logic language, and their truth on the built automaton is checked. The same big picture is used in [2], where *hybrid automata*, in place of classical ones, are employed to encode traces. Hybrid automata formalism allows to model systems in which continuous and discrete dynamics interact and have been recently used for systems biology purposes also in [1, 2, 9].

2.3 Process Calculi

Process calculi have been originally introduced in Computer Science as a theoretical framework to model and analyze communicating and mobile systems [12]: they are essentially *programming languages* target to describe concurrent and interacting systems. Recently, the use of Process Calculi within systems biology, as a formal tool to study biomolecular processes, was proposed in [17, 18]. In [13, 17] a variant of stochastic π-calculus [14] (see Table 1, below[3]) is *implemented* and applied to the *simulation* of a number of chemical and biological processes, showing evidence of high accuracy, intuitiveness and visibility of process-calculi based system biology.

Table 1: Syntax of the stochastic π-calculus, as implemented within `SPiM` machine in [13]

$P, Q ::= new\ x\ P$	Restriction	$\Sigma ::= 0$	Null
$\mid\ P \mid Q$	Parallel	$\mid\ \pi.P + \Sigma$	Action
$\mid\ \Sigma$	Choice	$\pi ::= !\, x(n)$	Output
$\mid\ *\pi.P$	Replication	$\mid\ ?\, x(n)$	Input
		$\mid\ \tau_r$	Delay

[2] a trace of $\mathcal{S}$ is simply an approximate solution of $\mathcal{S}$ computed, given an initial value for variables, over a time window $[0 \ldots t]$

[3] We adopt here the same notation used in [6, 13]. We refer to [6, 13] for a detailed description of the abstract machine underlying `SPiM` simulator for the execution of stochastic π-calculus

The big picture underlying each example developed in [13, 17, 18] is that of associating biological networks with networks of *communicating processes*; while in molecular realm interactions depend on structural and chemical complementarity of specific portions or *motifs*, within networks of processes the role of motifs is played by *global channels*. Each channel x is opportunely endowed of a *reaction rate*, $rate(x)$, to describe the quantitative behavior of biochemical systems. If present, compartments and modules within molecules are represented as *private channels* in processes, and biochemical modifications caused by interactions are mapped to *channel transmissions* for new communications.

Example 2.3 *As a simple example, consider the chemical reaction:*

$$Na + Cl \rightarrow Na^+ + Cl^- \tag{1}$$

To model by process calculi the chemical reaction 1 we need to establish a connection between the given chemical system and a system of communicating processes. Hence, chemical species Na and Cl will correspond to processes `Na` *and* `Cl`*. The reaction capability associated to the system composed by Na and Cl should correspond to a communication capability associated to the system of processes composed by* `Na` *and* `Cl`*. Further, a communication involving* `Na` *and* `Cl` *should result in a system composed by* `Nap` *and* `Clm` *processes, representing Na^+ and Cl^-, respectively. Hence, using the sintax given in Table 1, we declare:*

$$\texttt{Na}::=?e().\texttt{Nap}$$
$$\texttt{Cl}::=!e().\texttt{Clm}$$

were e is the channel allowing communication between `Na` *and* `Cl` *processes. Then, the system is constituted by parallel composition,* `Na` $|$ `Cl`*, which can yield to* `Nap` $|$ `Clm`*.*

A more challenging example, namely the process calculi description of gene regulatory network in [6], is reported in Section 3, as preliminary to the work developed in such a section.

3 Toward Integration of Biological Systems Models

In [21] Voit credits as one of the outstanding features of S-systems the possibility of developing simple algorithms to automatically extract equations from metabolic maps (see example 2.2). Here, we extend the picture discussing how S-systems can be automatically derived from π-calculus based descriptions of biological processes.

More precisely, our (preliminary) work focuses on π-calculus descriptions of *gene regulatory networks* and on their (automatic) translation to/from S-systems. We start from [6], where the authors defined a *compositional* approach for process calculi descriptions of simple systems of genes, in which the translation of each gene can be inhibited/enhanced by the expression of other genes in the network. Hence, the *elementary (primitive) elements* composing gene networks π-calculi models in [6] are three kinds of processes:

- the *neg gate process* ($neg(a_i, a_j)$), involving communication onto channels a_i, a_j and defined as:

$$neg(a_i, a_j) = ?a_i.\tau_\nu.neg(a_i, a_j) + \tau_\varepsilon.(tr(a_j)|neg(a_i, a_j))$$
$$tr(a_j) = !a_j.tr(a_j) + \tau_\delta.0$$

  $neg(a_i, a_j)$ represents the expression of a gene, whose transcription is inhibited by the expression of some other gene in the network. Hence, the neg gate process makes a stochastic choice between constitutive transcription ($\tau_\varepsilon.(tr(a_j)|neg(a_i, a_j))$) and inhibitory stimulation ($?a_i.\tau_\nu.neg(a_i, a_j)$). If an (input) communication is established on the channel a_i, then the process enters a stochastic delay (τ_ν) during which transcription is inhibited, and then returns to the initial state. The process $tr(a_j)$, simply defines the action of expression of a gene into a protein production: the process makes a stochastic choice between either binding to an available binding site or delaying with rate δ (i.e. being degraded). We address the reader to [6] for further details.

- the *pos gate process* ($pos(a_i, a_j)$), involving communication onto channels a_i, a_j and defined as:

$$pos(a_i, a_j) = ?a_i.\tau_\nu.(tr(a_j)|pos(a_i, a_j)) + \tau_\varepsilon.(tr(a_j)|pos(a_i, a_j))$$

  $pos(a_i, a_j)$ represents the expression of a gene, whose transcription is enhanced by the expression of some other gene in the network.

- the *nullary input gate process* ($null(a_i)$) involving (output) communication on channel a_i and defined as:

$$null(a_i) = \tau_\varepsilon.(tr(a_i)|null(a_i))$$

  $null(a_i)$ describes a gene with constitutive transcription but neither positive nor negative regulation.

A π-calculus model of genes'network is represented by parallel composition of a set of the above described elementary processes:

$$Net = G_1|G_2|\ldots|G_n$$

where each G_i is a positive, negative or nullary gate.

3.1 Automatic Translation of S-systems and π-Calculus Based Models of Gene Regulation Networks

Given the above premises, we are now ready to set a translation procedure from gene networks π-calculus descriptions[4] to S-systems. Consider a general process calculi networks of genes, as above described:

$$Net = G_1|\ldots|G_n$$

We define $size(Net)$ as the number of different channels involved in at least one gate process of the network i.e.

$$size(Net) = |\{a_i \mid \text{channel } a_i \text{ is involved in at least one gate process of Net}\}|$$

Our translation algorithm uses the value $size(Net)$, as well as the *kind* of communication established on each channel (i.e. the kind of gates composing the network) to determine the topology of the S-system corresponding to Net. The number of channels onto which communication is established, $size(Net)$, determines the number of dependent and independent variables in the resulting S-systems. In our context, dependent variables act for mRNA/proteins concentrations, whereas each independent variable represents an expressed gene DNA concentration. Hence, the number of both dependent and independent variables is set to be equal to $size(Net)$ i.e. to the number of genes being transcribed and translated in the network. On the other hand, the kinds of gates allow to set the signs of apparent kinetic orders in the system. Intuitively, *pos* (*neg*) gates correspond to positive (negative) kinetic orders. More precisely, the S-systems associated to a given $Net = G_1|\ldots|G_n$, say $S_{Net} = (IV_{net}, DV_{Net}, DE_{Net})$, can be built by applying the 4 steps procedure depicted in Figure 1.

Note that, within the network of genes defined by a compositional approach in [6], the production of a protein following a gene transcription can *only regulate the synthesis* (either positively or negatively) of other components in the network (i.e. it can affects only the V^+ terms in the differential equations composing the S-system). There is no mutually regulation of degradation within the network (this would result in having general products, involving all variables, within the V^- term of each differential equation). In other words, if we would first translate a gene network in a corresponding biochemical-like map, we would obtain a map with only three kind of arrows:

[4] we can assume our gene networks are *well defined*, i.e. (by the very nature of the described problem) that, for each couple of channels a_i, a_j, either $neg(a_i, a_j)$ or $pos(a_i, a_j)$ does not belong to the network.

S-system Extraction$(Net = G_1|\ldots|G_n)$

Step1 —*Define the set of dependent variables*

(1) $m \leftarrow size(Net) \leftarrow |\{a_i \mid \exists G_j \in Net,\ G_j \text{ uses channel } a_i\}|$
(2) Let $DV_{Net} = \{X_1, \ldots, X_m\}$

Step2—*Define the set of independent variables*

(3) Let $IV_{Net} = \{Y_1, \ldots, Y_m\}$

Step3—*Define the system of differential equations and set kinetic orders*

(4) Let DE_{Net} be the set of m differential equations in which:

$$\forall i = 1 \ldots m: \qquad \dot{X}_i = \alpha_i Y_i^{g_{ii}} \textstyle\prod_{j=1}^{m} X_j^{g_{ij}} - \beta_i X_i^{h_{ii}}$$

$where:$

g_{ii}, h_{ii} are strictly positive constants,
$g^{ij} > 0 \quad if \quad Net \equiv pos(a_j, a_i)|Net'$
$g^{ij} < 0 \quad if \quad Net \equiv neg(a_j, a_i)|Net''$
$g^{ij} = 0 \quad otherwise$

Step4— **Return** $S_{Net} = (IV_{net}, DV_{Net}, DE_{Net})$

Figure 1: Automatic translation procedure from π-calculus gene networks to S-systems.

1. arrows that link an independent variable to a dependent one;
2. feedback arrows positively influencing the synthesis of dependent variables, in correspondence to *pos gates processes*.
3. feedback arrows negatively influencing the synthesis of dependent variables, in correspondence to *neg gates processes*.

The above considerations establish the correctness of the automatic translation procedure outlined in Figure 1. Indeed, such a procedure determines precisely only the *sign* of each parameter in the output S-system. However, since S-systems can be studied (for example with respect to their steady-state) in a complete symbolic way [21], this information is far from being useless. Moreover, in case a precise numeric form of S-system models need to be determined, the output of the above procedure can be used to *speed up* the final model determination, as we discuss in the next subsection.

3.2 Parameter Estimation

In the literature, a major technique to infer S-system models (with explicit values for all parameters) from time series data relies on the use of evolutionary algorithms [16, 19, 21]. Evolutionary algorithms are stochastic search techniques that mimic the natural evolution as proposed by Charles Darwin. In the context of S-systems extraction, the rôle of individual which is the unit elements of populations and generations, is played exactly by a set of S-system parameters. Recently, in particular, the authors of [19] combine two subfamilies of such algorithms, namely genetic algorithms and evolutionary

strategies, to define an efficient *two steps* search procedure. In a first stage a genetic algorithm is used to find a suitable topology of the target S-system i.e. to find the non-zero parameters among $\{g_{ij}\}, \{h_{ij}\}$. In the second phase, an evolutionary strategy is used, which is suited for optimizing problems based on real values. The task of the second phase in [19], is precisely that of optimizing the (non-zero) parameters in the S-system topology previously suggested. The evolutionary strategy process faces the problem of minimizing the *fitness* value f:

$$f = \sum_{i=1}^{N} \sum_{k=1}^{T} \left(\frac{\widehat{x}_i(t_k) - x_i(t_k)}{x_i(t_k)} \right)^2$$

where N is the total number of genes in the regulatory system, T is the number of sampling points taken from the time-series data, and $\widehat{x}$ and x distinguish between estimated and experimental data. Interfacing the two phases in [19], allow to avoid incurring in local minimum within the overall procedure; in general [16, 19], disposing of a topology significatively speed up the convergence of this minimization process.

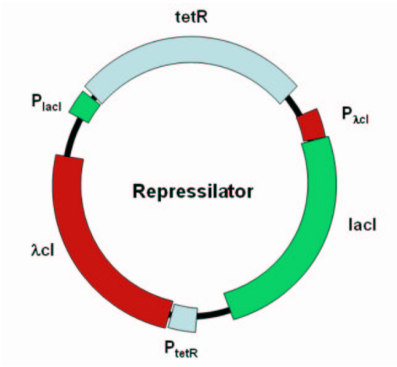
With these premises, observe the procedure in Figure 1 exactly extract the topology of the genetic network modeled with process calculi: as above argued (see [16, 19]), complete topology information allow faster convergence of evolutionary algorithms to good fitness values. In the case of π-calculus, the series of data to fit can be obtain simply by *executing* the process calculi network using a simulator[5] as, for example, the recently developed `SpiM` simulator [13]. Hence, by combining the translation routine in Figure 1 and genetic algorithms, it is possible to infer a precise S-system from a π-calculus gene network description.

3.3 Inverse Translation

Concluding this section, we observe that the inverse translation (producing a qualitative process calculi gene network description, given an S-system representation of a network of genes whose translation is mutually regulated) can be also automatically obtained, by simply reversing the translation function above developed.

4 The Repressilator Example

In this section we take into consideration the *repressilator* network [8] to show an application of the automatic translation procedures above developed.



Such a gene regulatory network, originally described and analyzed in [8], was considered for modeling purposes by supporters of both process calculi based systems biology [6, 18] and formal analysis of biological networks by means of systems of differential equations and automata-like tools [1, 2, 4]. As depicted in the figure on the right, a repressilator network [8] basically contains three genes/proteins, namely IacI, TetR, and λcI. These elements are (logically) arranged in a cyclic manner so that the protein product of one gene is repressor for the next gene. A process calculi model of the repressilator network is built and simulated, using the `SPiM` simulator, in [6]. The repressilator network consists simply of the three following processes executing in parallel:

$$RepNet = neg(a_1, a_2) | neg(a_2, a_3) | neg(a_3, a_1)$$

[5] Note here how, exactly the application of process calculi to biological systems models pursued research community to implement abstract machines for the π calculus execution.

The corresponding S-system, built using the translation procedure developed in the previous subsection, is the S-system

$$S_{RepNet} = (IV_{RepNet}, DV_{RepNet}, DE_{RepNet})$$

in which $size(RepNet) = 3$ and:

- $IV_{RepNet} = \{Y_1, Y_2, Y_3\}$;
- $DV_{RepNet} = \{X_1, X_2, X_3\}$;
- DE_{RepNet} is a the following set of three differential equations:

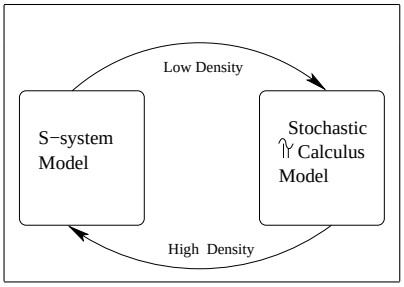
$$\begin{aligned}\dot{X}_1 &= \alpha_1 Y_1^{g_{11}} X_3^{g_{13}} - \beta_1 X_1^{h_{11}} \\ \dot{X}_2 &= \alpha_2 Y_2^{g_{22}} X_1^{g_{21}} - \beta_2 X_2^{h_{22}} \\ \dot{X}_3 &= \alpha_3 Y_3^{g_{33}} X_2^{g_{32}} - \beta_3 X_3^{h_{33}}\end{aligned}$$

  where $g_{13} < 0, g_{21} < 0, g_{32} < 0$ and all the other parameters are strictly positive.

The above built S-system has been recently computationally analyzed and "queried" by means of model checking techniques in [3, 4], within `Symphatica` tool: the authors of [3, 4] show, in particular, the possibility of testing automatically inquiries relative to the oscillatory dynamics of components.

5 Capturing Deterministic and Stochastic Behaviors of Biological Systems within Hybrid XS-Systems

A major advantage claimed by researchers developing process calculi based systems biology, is the capability of *easily* building models inherently considering stochastic effects. To date, there are in fact many evidences about the importance of noise and stochasticity in the evolution of biological processes, particularly when the number of molecules *gets small*. For these reasons, a very natural way of integrating process calculi systems biology and the analysis of biological systems by means of S-system (and automata theory) is that of building *hybrid* models, based on S-systems differential equations in presence of massive numbers of molecules, relying instead on stochastic π-calculus when the number of molecules gets small (see the figure above). Within biological processes analysis systems considering



the above suggested hybrid models, procedures for automatically translating the interfacing formalisms would be, of course, particularly useful. From a higher perspective, we conclude the paper claiming again the many positive byproducts coming from a systems biology tool integrating process calculi, S-systems, and automata-like modeling approaches, while inheriting much of the (complementary) advantages.

6 Conclusions and Future Work

In this paper we define automatic procedures for translating stochastic π-calculus descriptions of *gene regulatory networks* to/from *S-systems* differential equations. Techniques as the one developed in [3, 4] can be exploited to encode S-systems traces into an automata-like description, that can be further in-silico inquired. Hence, our work stays also as a bridge linking S-systems and π-processes Systems Biology formalisms, with automata-based modeling.

Some direction for future work are suggested from the following considerations. First, the fragment of π-calculus needed to describe gene regulatory networks in [6] is a very simple one: basically, scope extrusion and mobile connections are not considered. Second, our translation procedure is not really carried out at the level of translating individual features of π-processes, such as choice or communication, but only at the rougher level of components connected by parallel composition and links. Hence, a natural step extending the picture just outlined, consists in the definition of automatic translation procedures from/to S-systems models applying to whole stochastic π-calculus, and defined at the level of process language constructs.

References

[1] Alur, R., Belta, C., Ivančić, F., Kumar, V., Mintz, M., Pappas, G., Rubin, H., and Schug, J., Hybrid modeling and simulation of biomolecular networks, *Lecture Notes in Computer Science*, 2034:19–27, 2001.

[2] Antoniotti, M., Piazza, C., Policriti, A., Simeoni, M., and Mishra, B., Taming the complexity of biochemical models through bisimulation and collapsing: Theory and practice, *Theor. Comput. Sci.*, 325(1):45–67, 2004.

[3] Antoniotti, M., Policriti, A., Ugel, N., and Mishra, B., Model building and model checking for biochemical processes, *Cell Biochemistry and Biophysics*, 38(3):271–286, 2003.

[4] Antoniotti, M., Policriti, A., Ugel, N., and Mishra, B., Xs-systems: Extended s-systems and algebraic differential automata for modeling cellular behavior, *HiPC*, 431–442, 2002.

[5] Brauer, F. and Castillo-Chaves, C., *Mathematical Models in Population Biology and Epidemiology*, Springer, 2000.

[6] Cardelli, L., Blossey R., and Phillips, A., A compositional approach to the stochastic dynamics of gene networks, 2005.

[7] Clarke, E.M., Grumberg, O., and Peled, D.A., *Model Checking*, MIT Press, 1999.

[8] Elowitz M.B. and Leibler, S., A synthetic oscillatory network of transcriptional regulators, *Nature*, 403:335–338, 2000.

[9] Ghosh, R. and Tomlin, C., Lateral inhibition through delta-notch signaling: A piecewise affine hybrid model, *HSCC*, 232–246, 2001.

[10] Gilmore, S., Calder, M., and Hillston, J., Automatically deriving ODEs from process algebra models of signalling pathways, *Proceedings of Computational Methods in Systems Biology 2005*, 204–215, 2005.

[11] Kitano, H., Systems biology: A brief overview, *Science*, 295:1662–1664, 2002.

[12] Milner, R., *Communicating and Mobile Systems: The &pgr;-Calculus*, Cambridge University Press, 1999.

[13] Phillips, A. and Cardelli, L., A correct abstract machine for the stochastic pi-calculus, *Bioconcur 2004* ENTCS, 2004.

[14] Priami, C., Stochastic π-calculus, *Computer Journal*, 38:578–589, 1995.

[15] Savageau, M. A., Biochemical systems analysis: Dynamic solutions using a power-law approximation, 26:215–226, 1970.

[16] Shin, A. and Iba, H., Construction of genetic network using evolutionary, *Genome Inform.*, 14:94–103, 2003.

[17] Silverman, W., Priami, C., Regev, A., and Shapiro, E., Application of a stochastic name-passing calculus to representation and simulation of molecular processes, *Information Processing Letters*, 80:25–31, 2001.

[18] Silverman, W., Regev, A., and Shapiro, E., Representation and simulation of biochemical processes using the pi-calculus process algebra, *Pac. Symp. Biocomput. 2001*, 459–470, 2001.

[19] Spieth, C., Streichert, F., Speer, N., and Zell, A., Optimizing topology and parameters of gene regulatory network models from time-series experiments, *Lecture Notes in Computer Science*, 3102:461–470, 2004.

[20] Voit, O., *Canonical Nonlinear Modeling: S-System Approach to Understanding Complexity*, Van Nostrand Reinhold, 1991.

[21] Voit, O., *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*, Cambridge University Press, 2000.