Modelling the dynamics of biosystems

Corrado Priami and Paola Quaglia

Date received (in revised form): 10th June 2004

Corrado Priami
received his PhD degree in computer science from the University of Pisa, Italy. He is currently full professor at the Department of Computer Science and Telecommunications of the University of Trento, Italy, where he is the director of the Bioinformatics Research Group.

Paola Quaglia
received her PhD degree in computer science from the University of Pisa, Italy. She is currently research associate at the Department of Computer Science and Telecommunications of the University of Trento, Italy, and a member of the Bioinformatics Research Group.

Keywords: *models for molecular networks, systems biology, process algebras*

Abstract

The need for a more formal handling of biological information processing with stochastic and mobile process algebras is addressed. Biology can benefit this approach, yielding a better understanding of behavioural properties of cells, and computer science can benefit this approach, obtaining new computational models inspired by nature.

INTRODUCTION

The convergence between life sciences and computer science is becoming rapidly more and more evident. A triggering event for the speed-up of the convergence between life sciences and computer sciences as well as the new development in bioinformatics is the paradigm shift in biological investigation. The classical reductionist approach is not enough to define a model of a whole system starting from the knowledge of its minimal components owing to the large amount of information that can be made available by high-throughput tools. For instance it is not possible to infer the behaviour of a cell even if all of its genes are known.

In recent times, Leroy Hood and others introduced the concept of *systems biology* which they defined as the building of models of biological systems and then tuning/validating them via experiments that provide feedback. Hence reductionism is replaced by hypothesis-driven investigation. Systems biology agrees with the vision proposed by Robin Milner in his Turing Award lecture of computer science as an experimental science.[1] Computer systems are first modelled (generation of hypotheses), then implemented and tested (experiments) to refine/validate the model (feedback loop). Matching the two definitions and abstracting from experiments (wet biology on one side and *in silico* simulation on the other), we could state that *systems biology is computer science in the applicative domain of life sciences*.

Systems biology is well integrated with the new frontiers of biological research that are looking for *functions* (behaviour) of biological components and systems (functional genomics, functional proteomics, etc, are examples of buzzwords). Since any gene and any protein can be viewed as a functional unit that operates concurrently with hundreds of thousands of other functional units, possibly interacting with them by exchanging (chemical) messages, biological systems should be considered as information devices with their own computational models.

The shift from structure to function in biology imposes a similar shift in the bioinformatics realm. Although the term *bioinformatics* introduced by Hwa Lim in the 1980s was intended for *the study of the information content and information flow in biological processes and systems*, the research has mainly concentrated on the content (ie the structure) rather than on the flow (ie the behaviour). Therefore, if in the past the computer science field mainly addressed algorithms and static databases under the word bioinformatics, the behaviour challenge is now calling for (concurrent and distributed) programming and simulation. This view is also supported by Peter Sorger (director of MIT's Computational and Systems Biology Initiative), who states that 'string-based foundation of sequence-centric

Corrado Priami,
Dipartimento di Informatica e Telecomunicazioni,
Università di Trento,
via Sommarive 14,
38050 Povo (TN), Italy

Tel: +39 0461 882085
Fax: +39 0461 882093
E-mail: priami@dit.unitn.it

bioinformatics will not hold up as more biologists begin studying pathways and networks and open access to models will be much more critical than open access to databases'.

Systems biology is explicitly addressing the information flow that governs the whole behaviour of systems and assumes that thousands of simultaneous threads of computations are active in a system (eg metabolic networks, gene regulatory networks, signalling pathways). The effect of an interaction between the components of a system can change the future behaviour of the whole system. Some interactions can occur only if the interacting components are correctly located one to each other (eg they are close enough or they are not divided by membranes).

Finally, systems biology seems to be the roadmap to study how the pieces that biologists studied for decades fit together to obtain the so-called *big picture* of life. The challenge is here the capability of transcending molecular biology and understanding organisms as complex *interacting* systems. The enabling technology should be able to handle integration of large data sets, and infer from them dynamical models on which it is possible to carry out analysis and simulation to provide feedback to biologists and drive their real experiments.

Summing up, it is becoming widely accepted that computational models are at the core of systems biology research and that, quoting Leroy Hood, 'the big challenge in building such models is integrating information from the different levels of a system, such as DNA sequence data with mRNA data, protein sequence and structure, pathways, and networks.' This paper moves a step ahead and proposes concurrency theory and process algebras (the field of computer science developed to program and study mobile and distributed systems) as the key ingredient to face the challenge.

The paper is organised as follows. In the next section the state of the art in modelling biological systems is briefly surveyed. The main features of process algebras are then described, ie formalisms that are normally used to model distributed systems and have recently been adopted and extended to model biological systems. The added value that process algebras may provide to modelling biological entities is then commented on.

MODELLING BIOSYSTEMS: THE STATE OF THE ART

Various computational approaches are actively developed and used to model and study molecular networks. They can be roughly divided into several groups.

The main approaches are the following.

Biochemical kinetic models

- **Biochemical kinetic models** that describe different molecular systems from a pure biochemical perspective (examples are the models proposed by Sauro,[2] McAdams and coworkers[3–5] and Voit[6]). These use either continuous, mass-action differential equations or corresponding discrete, stochastic models to simulate and analyse molecular pathways. While these models are capable of handling (predominantly quantitative) aspects of molecular systems, their intricacies and sensitivity to numerical parameters often restrict their applicability to highly specified small systems and deter molecular biologists.

Generalised models of regulation

- **Generalised models of regulation** that describe and simulate gene regulatory circuits, using binary Boolean networks, introduced by Kauffmann[7] (examples are the models used in the works by Sanchez, Thieffry, Mendoza and coworkers,[8–11] and Akutsu *et al.*[12,13]). These approaches allow an abstract study of general properties of large networks but suffer from limited predictive power and from being bound to the investigation of regulatory circuits.

Functional object-oriented databases

- **Functional object-oriented databases** that store information on molecular pathways (eg EcoCyc,[14]

MPW,[15] KEGG,[16] CSNDB,[17] aMAZE,[18] GeNet,[19] TRANSFAC,[20] INTERACT,[21] DIP,[22] BIND,[23] SPAD[24] and FlyNets[25]). These use sophisticated object-oriented schemas that provide a biologically appealing hierarchical view of molecular entities. Most are equipped with graph-based visualisation tools and querying tools of variable levels of sophistication, from simple queries to pathway reconstruction tools. Functional databases provide an excellent solution for organising, manipulating and (sometimes) visualising pathway data. However, they provide little, if any, dynamic capabilities (eg simulation) and their analytical querying tools are thus seriously limited.

Exchange languages

- **Exchange languages** have recently been developed to promote the integration of models and tools from various sources (examples are the languages proposed by Kazic[26] and Finney *et al.*[27]). Most, but not all, are XML-based mark-up languages (eg CellML and SBML). While these languages could prove highly useful for the integration of various tools for pathway informatics, they currently often lack in expressivity (which varies from basic chemical reactions to more complex hierarchical models). They are not easily readable (and not intended for direct use by biologists) and do not incorporate specific tools for analysis or simulation.

Approaches based on formal methods

- **Approaches based on formal methods** have gained increasing importance during the past few years. Notable examples use existing formalisms from concurrent computation, often with a strong graphical component. The most comprehensive works used Petri nets (examples are the works by Goss, Matsuno, Kuffner, Hofestadt and coworkers[28–32]) for representation, simulation and analysis of metabolic pathways. In another work a 'Pathway Logic' was developed based on the Maude platform, and used for rudimentary qualitative analysis.[33] Recent studies used Statecharts to build qualitative graphical models for various signalling pathways.[34–36] Unlike the typical simplistic graph-based representation used for pathway visualisation, Statecharts provide a rich and expressive language with clear semantics. The recent control theory literature contains several relevant works on, for example, the general problem of constructing an automaton from a differential equation models,[37] and hybrid systems models for biochemical reactions.[38] These recent attempts highlight the promise in using formal methods for pathway informatics. However, the specific models and tools described above are limited in their ability either to handle quantitative data for building and analysing the model (eg Statecharts) or to represent complex systems as composition of the specifications of a set of simpler subsystems (eg Petri nets).

Integrated frameworks with GUI

- **Integrated frameworks with GUI**. Several initiatives are underway to promote more comprehensive solutions. Some of these efforts, such as the Systems Biology Workbench,[39] are aimed at providing the infrastructure to promote exchange and integration of independent, separate solutions rather than develop the tools and models themselves. Other projects use well-established models to develop simulation environments (eg E-Cell[40]) or extensive databases (eg EcoCyc or aMAZE) equipped with sophisticated tools for visualisation and querying. An issue here is hiding formal details from biologists through meta-modelling tools.

In conclusion, each of these approaches captures some of the information regarding pathways and their

components, and provides certain capabilities for their analysis. However, none provides both a comprehensive, quantitative, dynamic and biologist-friendly language and serious tools for analysis and simulation. With the notable exception of Statecharts (above), the visualisation is limited to simple and directed graphs and does not capture the richness of biological information. Importantly, some of the tools have hardly been tested with real biological data and have not been actively used to derive new knowledge by biologists. Furthermore, certain key questions, mostly in static analysis and pathway comparison, are hardly addressed or not addressed at all by any of the existing tools.

Process algebras and molecular networks

It is suggested that process algebras for mobility (see next section) may encompass most of the limitations mentioned above. Process algebras provide the basis to study in a more systematic way hypotheses on properties of complex systems of biochemical reactions. Some research work by Bhalla and Iyengar,[41] for example, aims at proving that a sort of 'learned behaviour' of biological systems is in fact stored within the mechanisms regulating intracellular biochemical reactions constituting signalling pathways. For this kind of study both qualitative and quantitative features of the system under study should be taken into account.[42]

THE PROCESS ALGEBRA APPROACH

Mobile process algebras

The abstract characteristics of biological systems are the same as those of distributed and mobile systems. Many processes are active simultaneously over a set of physical resources for which they compete while cooperating to accomplish a common goal. Acquisition of a resource from a process or reception of a message upon which choices have to be taken can surely affect the future behaviour of the whole system and even change the logical interconnection structure among processes. Trust barriers and administrative domains work as membranes that can be passed only by those processes that possess the right keys – hence the concept of localisation of processes is an important one.

Mimicking the description of mobile and distributed systems in a biological domain, it can be stated that processes are the biological components. Sharing of channels establishes the interconnection topology of the system and represents the interaction potentials of components together with their affinity. Scopes of channels or explicit binders represent the boundaries within which interactions through such channels may occur. Since channel names can be sent as data along channels, the interconnection topology varies dynamically, so modelling the impact of an interaction on the future behaviour of the whole system. The above interpretation immediately provides a dynamic description of the temporal as well as causal evolution of the system in hand: we only need to run the program.

The features above can be handled by mobile process algebras, the first of which (π-calculus) was introduced by Milner, Parrow and Walker in the 1980s to model rigorously mobile systems.[43] They are made up of few operators to compose elementary actions (say α) over distributed channels (denoted hereafter by their names, given in lower-case letters). These operators are: sequentialisation ($\alpha.P$), parallel composition ($P|Q$), name declaration ($\nu\, x$), and recursion ($rec\, x.P$).

The intuitive meaning of sequentialisation is that the atomic action α is the first that the process $\alpha.P$ can execute. Atomic actions can be the output of a name b over a channel a ($a!b$), the reception of a datum on a channel a that will replace the placeholder x in the prefixed process ($a?x$), or an internal action of the system (τ). Reception $a?x$ binds the free occurrences of the variable x within P. The ν operator in $(\nu x)P$ declares x to be private to P. The parallel composition $P|Q$ allows the processes P and Q to be executed independently of

one another and they can communicate if they share a communication channel. For instance $a!b.P \mid a?y.Q$ can perform a communication by sending b over the channel a from the left hand process to the right hand one, yielding $P \mid Q\{b/y\}$, with $\{b/y\}$ being the substitution of b for the free occurrences of y in Q. Eventually, $recx.P$ stands for the possible unfolding of process P as many times as needed. Sometimes iteration is represented through the *bang* operator ($!P$) that is interpreted as many copies of P as needed.

Structural operational semantics

The formal semantics of process algebras is usually given in terms of the logics based Plotkin's structural operational semantics.[44] The dynamic behaviour of systems is then expressed in terms of transition systems, ie oriented and labelled graphs where the nodes are the states of the system and the arcs represent, via their labels, the actions that make the system change from one state to the other.

Transition systems

For instance the process (program) $P = (\nu p)[(p!a.nil) \mid (p?x.x!b.nil)]$, where *nil* represents deadlock, ie complete inability to perform any action, generates the transition graph reported in Figure 1. The intuition is as follows. Since p is a private name (as shown by the top-level occurrence of (νp)), P cannot offer it to its external environment, namely the execution of both the output action $p!a$ and the input action $p?x$ is forbidden. The single possible move of P corresponds to the internal communication of its parallel subcomponents, which gives rise to the synchronisation action τ and transforms the system in the new state $P' = (\nu p)(nil \mid (a!b.nil))$. The top-level output action $a!b$ can now be executed, leading P' to the process $(\nu p)(nil \mid nil)$. This latest process, being deadlocked, cannot move further.

Compositionality

Process algebras have been extensively used in the computer science community for the specification and verification of concurrent and mobile systems. Also, a number of techniques have been studied and developed to check the specified systems against the mathematical representation of desirable properties such as for a system expressing a mobile phone setting, 'the phone call will eventually be delivered through the base station'. Although the field is quite young (and computer science as a whole is a young discipline indeed), the results obtained are very promising. Recently, also enterprises of various sizes are developing commercial languages and environments which are based on the formal ground of process algebras.

An important property of mobile process algebras is *compositionality*. The meaning or the behaviour of a complex system is expressed in terms of the meaning of its components. This allows one to concentrate on the basic operations that a system can perform and to obtain the whole behaviour through composition of these basic building blocks. Compositionality is surely the key issue needed by systems biology to become effective. Indeed the huge amount of data available, and the complexity of the interacting networks analysed, make it impossible to define *formally* the behaviour of biological systems when they are considered as a whole.

Notice that ordinary differential equations (ODEs), although being the most accepted models for representing dynamical biological systems,[45] cannot deal with compositionality. In the ODE-based approach, a biomolecular system is abstracted in terms of quantifiable properties of its components. The

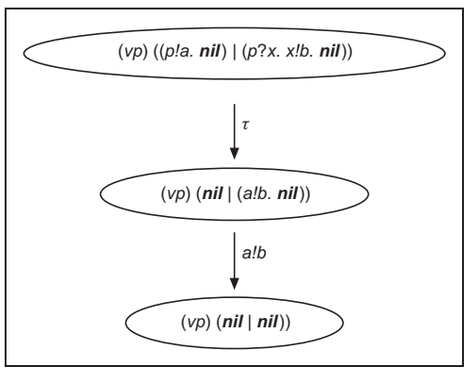

Figure 1: Transition system associated with $(\nu p)[(p!a.nil) \mid (p?x.x!b.nil)]$

ODE *vs* process algebras

formalism allows the representation of the time-dependent concentration of components as functions of the concentrations of the other components. In tuning the above functions one needs to know both the kinetics of the modelled reaction and a number of parameters (eg production and degradation constants) coming from *in vivo* or *in vitro* measurements. Once the underlying kinetics and parameters are clear, the whole physical phenomenon is described by a system of ODEs. As an example, imagine that System 1 and System 2 in Figure 2 are separately modelled by the set of ODEs E1 and E2, respectively. Then it is not necessarily true that the system of ODEs obtained by joining together the equations in E1 and E2 represents the kinetics of the system made of both System 1 and System 2. For instance, it might well be the case that some biochemical components of System 2 are missing in System 1 and hence the equations in E1 do not take care of them. The compositionality of mobile process algebras allows us to separately specify the two systems and then obtain the whole behaviour simply by putting them in parallel (System1 | System2).

Stochastic process algebras

The abstraction provided by process algebras was shown to be successful in modelling several scenarios from life sciences, including transcriptional circuits, metabolic pathways and signal transduction networks.[46–49] Nonetheless the calculi used to achieve those results are (or enhance) formal models originally designed to specify distributed interacting systems. For this reason, they generally do not possess either graphical/linguistic support for the peculiarities of biological interactions, or operators specifically thought of for representing that sort of cooperation. Think, for example, of membranes, enclosing surfaces, shapes, energy, bidirectional communication, reversibility of reactions, and affinity. Some promising efforts in this direction have recently been made in the computer science research community (see, for instance, works by Regev, Danos, Priami and coworkers[50–53]). Most of them provide, besides the usual process algebra interpretation of interaction as communication, some means to model 'borders' of entities. Some of the above formalisms also set the mathematical bases for well-founded reasoning about reversibility and available energy. In particular, some recent works by Cardelli on modelling membrane interactions are good examples of the flexibility of the approach.[54] These works show that, even if to date most of the successful applications of process algebras to the description of biological phenomena focuses on the molecular level, the input/output coordination model provided by the process algebra approach is (at least potentially) amenable to the description of higher-level interactions.

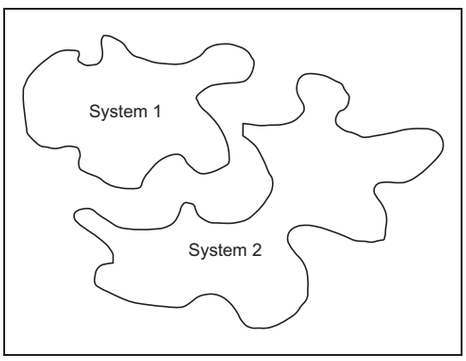

Figure 2: Example of biological entities

Finally, biological models are driven by a lot of quantitative information concerning for instance energy, time, affinity, distance, electrostatic charge, number of components. Therefore on the computer science side we must resort to stochastic variants of mobile process algebras (examples are works by Priami[55,56]) thus having the formal tools to include numbers in system specifications. The basic idea is to replace actions α with pairs (α, f), where f is a probabilistic distribution function driving the selection of the action to fire among all the ones enabled. The mechanism is a race condition: all the activities enabled attempt to proceed, but only the fastest one succeeds. Relying on continuous time distributions we ensure that two

activities cannot end simultaneously, thus ruling out any non-determinism from our models.

Computer science and systems biology

We now have the computer science counterpart of systems biology: models are specified by using stochastic process algebras, experiments are carried out *in silico* (rather than wet biology) relying on analysis, verification and simulation techniques developed in the last decades in the field of concurrency theory and refined for the new applicative domain.

Equivalences

The neat result is that we can try hundreds of experiments without using reactants or animals – of course we must tune the techniques adopted by comparing the *in silico* results with real wet experiments. Once the framework is tuned we can use it in a predictive fashion. *In silico* experiments can help biologists selecting real experiment strategies among a plethora of possible ones. For instance it is possible to investigate how a new drug can break a signalling pathway leading to a disease by leaving unaffected as many functionalities of a complete signalling network as possible (reduction of side effects of drugs), or even how transduction mechanisms leading to DNA damage are activated.

ANALYSIS AND SIMULATION

The added value of modelling biosystems with process algebras for mobility is given by the already available techniques for analysis and simulation.

Concurrency, causality, locality

Useful tools have been defined during the last years to extract causality, locality, concurrency information from the specification of systems (some examples are reported in works by Degano, Boreale, Castellani and coworkers[57–59]). The notion of compartments (or location where reactions occur) is essential in biology. There are tools that handle this localisation of actions and that were developed to take care of the location of mobile appliances or to take care of administrative domains barrier for security reasons. The very same problems, and also the need to better understand the behaviour of complex concurrent systems, posed the challenge of defining and automatically computing a relation of causality between events. This notion could help also in systems biology as it introduces a notion of distributed flow of time. Any set of reactions has its own clock that must be synchronised with the clock of another set only when there is a flow of information between the two. The set of clock synchronisations provides an implicit notion of causality between different flows of information. Furthermore, causality can be used to track the activation factors of diseases.

Methods and tools that compare the behaviour of different systems by abstracting from their precise structure and based on the notion of bisimulation have been developed.[60–62] Indeed it turns out that systems with different structure may share the same behaviour with respect to some observational property. It could be the same for biological systems and this sharing could be investigated with a fine tuning of the same bisimulation checker used in the process algebra field. Furthermore, behavioural equivalences can be used to inspect whether a complex specification in a process algebra exposes a desirable high-level biological behaviour. The idea is to specify the high-level behaviour using the same process algebra and abstracting from implementation details. If the two specifications are equivalent with respect to the observable property, we can state that the complex specification actually encodes the high-level biological behaviour.

All the tools and methods described above can be used both qualitatively and quantitatively. The quantitative measures can be available only for some pieces of the models and for the others they can be left unspecified so that the tools can do inferences (always notifying the user) or can ask the user for values. This facility can be used for parameter checking or for parameter discovery. One can run many analyses simply changing some parameters

to see how the outcome changes, and then infer which could be a suitable experiment to discover the actual value of the parameters.

The integration of quantitative measures in analysis and verification tools is also useful to check consistency of quantitative data available in the literature. Different sets of data may not be compatible with each other or may not be consistent with the qualitative description of the phenomenon. An immediate application of literature mining will help this validation of experimental data.

Linguistic features of modelling languages

Equipping mathematical description languages with quantitative parameters permits the implementation of stochastic (or deterministic) simulators of the behaviour of the systems. Therefore the user can study the variation of concentration of substances as time passes. Again this feature is strongly integrated with the analysis tools so that the user can also compare the outputs of different tools on the same phenomenon. The first relevant example of this approach is the BioSpi system.[63] The majority of tools developed for the analysis of process algebra specifications are implemented in C, C++ or some functional language (see the website[64] for details). Differently from them, the current implementation of BioSpi is based on Flat Concurrent Prolog which is usually considered to be computationally less efficient than the aforementioned languages. The BioSpi prototype, however, has been shown to run huge systems (order of hundreds of parallel processes) relatively fast (order of seconds). Furthermore, the main positive aspect of the tool is the evidence of the feasibility of the approach, and this is a mandatory step towards efficiency.

Computer science added value

CONCLUSIONS

Systems biology added value

The main added value for systems biology in joining the process algebra approach is given by the abstraction mechanisms that computer scientists have been developing for concurrent systems over the last 30 years,[65] so that a considerable speed-up in life science research could be possible especially on the fields of predictive, preventive and personalised medicine, drug design and gene therapy, toxicological research and environmental research. Besides abstraction, the main feature to be exploited in systems biology is compositionality. It allows us to fix the building blocks of systems and to enlarge models by composition without changing the description of the subsystems already available as it would be the case for ODE, Petri nets or Statecharts. There is a general understanding in the scientific community that computer science will be as indispensable for biology as mathematics has been for physics.

On the other hand it should be made clear that computer science cannot be only a service for biology otherwise the model cannot work, owing to the different expectations of the two research communities. The best way to proceed is to create a new interdisciplinary community in which all the components have the same weight. A new definition of bioinformatics by NCBI and available at the website[66] catches exactly this point: *bioinformatics is the science in which biology and computer science join together to ease new biological discovery and to define new computational paradigms inspired by living systems*.

The added value for computer science in joining systems biology is well expressed in the last part of the definition above. It is hoped that the identification of abstraction mechanisms to model living systems may provide us with new computational and information processing paradigms that could successfully be applied in programming the net at various level of coordination: from grid computing to global computing. Another obvious application of reusing biological information in the computer science domain concerns security: once there is a complete model of the immune systems the same conditions can be recreated in artificial systems such as the net (also this strategy begins to be investigated, see, for example, recent research by Chao and Forrest[67]).

Further steps

A first step towards the definition of new computational models is the understanding of the living systems that can be accomplished if we provide biologists with integrated environments in which they can model and simulate systems. This poses two technological challenges. First, we should provide them with environments that hide from the user as many technical details as possible. Second, we should be able to fully integrate the new tools with the ones that are already in use, such as databases and data/literature mining frameworks.

Acknowledgments

The authors have been partially supported by the FIRB project '*Modelli formali per Sistemi Biochimici*'.

References

1. Milner, R. (1993), 'Elements of interaction (Turing Award Lecture)', *Commun. ACM*, Vol. 36(1), pp. 78–89.
2. Sauro, H. M. (1993), 'SCAMP: A general-purpose simulator and metabolic control analysis program', *Computer Appl. Biosci.*, Vol. 9(4), pp. 441–450.
3. McAdams, H. H. and Shapiro, L. (1995), 'Circuit simulation of genetic networks', *Science*, Vol. 269(5224), pp. 650–656.
4. McAdams, H. H. and Arkin, A. (1997), 'Stochastic mechanisms in gene expression', *Proc. Natl Acad. Sci. USA*, Vol. 94(3), pp. 814–819.
5. Arkin, A., Ross, J. and McAdams, H. (1998), 'Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli* cells', *Genetics*, Vol. 149, pp. 1633–1648.
6. Voit, E. O. (2000), 'Computational Analysis of Biochemical Systems', Cambridge University Press, Cambridge.
7. Kauffman, S. A. (1969), 'Metabolic stability and epigenesis in randomly constructed genetic nets', *J. Theoret. Biol.*, Vol. 22, pp. 437–467.
8. Sanchez, L., van Helden, J. and Thieffry, D. (1997), 'Establishment of the dorso-ventral pattern during embryonic development of *Drosophila melanogaster*. A logical analysis', *J. Theoret. Biol.*, Vol. 189(4), pp. 377–389.
9. Thieffry, D. and Thomas, R. (1998), 'Qualitative analysis of gene networks', in 'Pacific Symposium on Biocomputing', Altman, R. B., Dunker, A. K., Hunter, L. and Klein, T. E., Eds, Vol. 3, World Scientific Press, Singapore, pp. 77–88.
10. Mendoza, L., Thieffry, D. and Alvarez-Buylla, E. R. (1999), 'Genetic control of flower morphogenesis in *Arabidopsis thaliana*: A logical analysis', *Bioinformatics*, Vol. 15(7–8), pp. 593–606.
11. Thieffry, D. and Romero, D. (1999), 'The modularity of biological regulatory networks', *Biosystems*, Vol. 50(1), pp. 49–59.
12. Akutsu, T., Miyano, S. and Kuhara, S. (2000), 'Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function', *J. Comp. Biol.*, Vol. 7(3), pp. 331–343.
13. Akutsu, T., Miyano, S. and Kuhara, S. (2000), 'Algorithms for inferring qualitative models of biological networks', in 'Pacific Symposium on Biocomputing', Altman, R. B., Dunker, A. K., Hunter, L. and Klein, T. E., Eds, Vol. 5, World Scientific Press, Singapore, pp. 293–304.
14. Karp, P. D., Krummenacker, M., Paley, S. and Wagg, J. (1999), 'Integrated pathway/genome databases and their role in drug discovery', *Trends Biotechnol.*, Vol. 17(7), pp. 275–281.
15. Selkov, E., Grechkin, Y., Mikhailova, N. *et al.* (1998), 'MPW: The Metabolic Pathways Database', *Nucleic Acids Res.*, Vol. 26(1), pp. 43–45.
16. Ogata, H., Goto, S., Sato, K. *et al.* (2000), 'KEGG: Kyoto Encyclopedia of Genes and Genomes', *Nucleic Acids Res.*, Vol. 27(1), pp. 29–34.
17. Igarashi, T. and Kaminuma, T. (1997), 'Development of a cell signalling networks database', in 'Pacific Symposium on Biocomputing', Altman, R. B., Dunker, A. K., Hunter, L. and Klein, T. E., Eds, World Scientific Press, Singapore, pp. 187–197.
18. van Helden, J., Naim, A., Mancuso, R. *et al.* (2000), 'Representing and analysing molecular and cellular function using the computer', *Biol. Chem.*, Vol. 381(9–10), pp. 921–935.
19. Kolpakov, F. A., Ananko, E. A., Kolesov, G. B. and Kolchanov, N. A. (1998), 'GeneNet: A gene network database and its automated visualization', *Bioinformatics*, Vol. 14(8), pp. 529–537.
20. Wingender, E., Chen, X., Fricke, E. *et al.* (2001), 'The TRANSFAC system on gene expression regulation', *Nucleic Acids Res.*, Vol. 29(1), pp. 281–283.
21. Eilbeck, K., Brass, A., Paton, N. and Hodgman, C. (1999), 'INTERACT: An object oriented protein–protein interaction database', in 'Intelligent Systems for Molecular Biology', Vol. 7, AAAI Press, Palo Alto, pp. 87–94.
22. Xenarios, I. E. E. F., Salwinski, L., Duan, X. J. *et al.* (2001), 'DIP: The database of interacting proteins: 2001 update', *Nucleic Acids Res.*, Vol. 29(1), pp. 239–241.

23. Bader, G. D., Donaldson, I., Wolting, C. *et al.* (2001), 'BIND – the Biomolecular Interaction Network Database', *Nucleic Acids Res.*, Vol. 29(1), pp. 242–245.
24. URL: http://www.grt.kyushu-u.ac.jp/spad/
25. Sanchez, C., Lachaize, C., Janody, F. *et al.* (1999), 'Grasping at molecular interactions and genetic networks in *Drosophila melanogaster* using FlyNets, an Internet database', *Nucleic Acids Res.*, Vol. 27(1), pp. 89–94.
26. Kazic, T. (2000), 'Semiotes: A semantics for sharing', *Bioinformatics*, Vol. 16(12), pp. 1129–1144.
27. Finney, A., Sauro, H., Hucka, M. *et al.* (2000), 'An XML-based Model Description Language for Systems Biology Simulations', Technical Report, California Institute of Technology.
28. Goss, P. J. E. and Peccoud, J. (1998), 'Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets', *Proc. Natl Acad. Sci. USA*, Vol. 95(12), pp. 6750–6755.
29. Goss, P. J. E. and Peccoud, J. (1999), 'Analysis of the stabilizing effect of Rom on the genetic network controlling ColE1 plasmid replication', in 'Pacific Symposium on Biocomputing', Altman, R. B., Dunker, A. K., Hunter, L. and Klein, T. E., Eds, Vol. 4, World Scientific Press, Singapore, pp. 65–76.
30. Matsuno, H., Doi, A., Nagasaki, M. *et al.* (2000), 'Hybrid Petri net representation of gene regulatory network', in 'Pacific Symposium on Biocomputing', Altman, R. B., Dunker, A. K., Hunter, L. and Klein, T. E., Eds, Vol. 5, World Scientific Press, Singapore, pp. 341–352.
31. Kuffner, R., Zimmer, R. and Lengauer, T. (2000), 'Pathway analysis in metabolic databases via differential metabolic display', *Bioinformatics*, Vol. 16(9), pp. 825–836.
32. Hofestadt, R. and Thelen, S. (1998), 'Quantitative modeling of biochemical networks', *In Silico Biology*, Vol. 1(1), pp. 39–53.
33. Eker, Z., Knapp, M., Laderoute, K. *et al.* (2002), 'Pathway logic: Symbolic analysis of biological signaling', in 'Pacific Symposium on Biocomputing', Altman, R. B., Dunker, A. K., Hunter, L. and Klein, T. E., Eds, Vol. 7, World Scientific Press, Singapore.
34. Holcombe, M. and Bell, A. (1998), 'Computational models of immunological pathways', in 'Information Processing in Cells and Tissues: Proceeding of IPCAT '97', M. Holcombe and R. Paton, Eds, Plenum Press, New York, pp. 213–226.
35. Kam, N., Cohen, I. and Harel, D. (2001), 'The immune system as a reactive system: Modeling T cell activation with Statecharts', in 'Proc. Symposia on Human-Centric Computing Languages and Environments', IEEE Computer Society Press, Oxford, pp. 15–22.
36. Kam, N., Harel, D. and Cohen, I. (2001), 'Modeling biological reactivity: Statecharts vs. boolean logic', in 'Proc. 2nd International Conference on Systems Biology', Pasadena, CA.
37. Brockett, R. W. (1994), 'Dynamical systems and their associated automata', in 'Systems and Networks: Mathematical Theory and Applications – Proceedings of the 1993 MTNS', Helmke, U., Mennicken, R. and Saurer, J., Eds, Vol. 77, Akademie-Verlag, Berlin, pp. 49–69.
38. Alur, R., Belta, C., Ivancic, F. *et al.* (2001), 'Hybrid modeling and simulation of biological systems', In 'Proc. of the Fourth International Workshop on Hybrid Systems: Computation and Control', LNCS 2034, Springer-Verlag, Berlin, pp. 19–32.
39. URL: http://www.sbw-sbml.org/the_project.html
40. Tomita, M., Hashimoto, K., Takahashi, K. *et al.* (1999), 'E-CELL: Software environment for whole-cell simulation', *Bioinformatics*, Vol. 15(1), pp. 72–84.
41. Bhalla, U. S. and Iyengar, R. (1999), 'Emergent properties of networks of biological signaling pathways', *Science*, Vol. 283(5400), pp. 381–387.
42. Endy, D. and Brent, R. (2001), 'Modeling cellular behavior', *Nature*, Vol. 409(18), pp. 391–395.
43. Milner, R., Parrow, J. and Walker, D. (1992), 'A calculus of mobile processes (I and II)', *Inf. Comp.*, Vol. 100(1), pp. 1–77.
44. Plotkin, G. D. (1981), 'A Structural Approach to Operational Semantics', Technical Report DAIMIFN-19, Aarhus University, Denmark.
45. DeJong, H. (2002), 'Modeling and simulation of genetic regulatory systems: A literature review', *J. Comp. Biol.*, Vol. 9(1), pp. 67–103.
46. Regev, A., Silverman, W. and Shapiro, E. (2000), 'Representing biomolecular processes with computer process algebra: π-calculus programs of signal transduction pathways', in 'Proceedings of the Pacific Symposium of Biocomputing 2000', World Scientific Press, Singapore.
47. Priami, C., Regev, A., Silverman, W. *et al.* (2001), 'Application of a stochastic name-passing calculus to representation and simulation of molecular processes', *Inf. Proc. Lett.*, Vol. 80, p. 1.
48. Regev, A. (2001), 'Representation and simulation of molecular pathways in the stochastic π-calculus', in 'Proceedings of the 2nd Workshop on Computation of Biochemical Pathways and Genetic Networks', Logos, Berlin.

49. Regev, A., Silverman, W. and Shapiro, E. (2001), 'Representation and simulation of biochemical processes using the pi-calculus process algebra', in 'Proceedings of the Pacific Symposium of Biocomputing 2001', Vol. 6, pp. 459–470, World Scientific Press, Singapore.
50. Regev, A., Panina, E., Silverman, W. *et al.* (2004), 'BioAmbients: An abstraction for biological compartments', *Theoret. Comp. Sci.* (in press).
51. Danos, V. and Laneve, C. (2003), 'Core formal molecular biology', in 'European Symposium on Programming (ESOP03)', Springer, Berlin.
52. Danos, V. and Krivine, J. (2003), 'Formal molecular biology done in CCS-R', in 'BioConcur '03, Workshop on Concurrent Models in Molecular Biology', Elsevier, Amsterdam.
53. Priami, C. and Quaglia, P. (2004), 'Beta binders for biological interactions', in 'Proc. 2nd Int. Workshop on Computational Methods in Systems Biology, CMSB '04', Springer, Berlin.
54. Cardelli, L. (2003), 'Membrane interactions', in 'BioConcur '03, Workshop on Concurrent Models in Molecular Biology', Elsevier, Amsterdam.
55. Priami, C. (1995), 'Stochastic π-calculus', *Comp. J.*, Vol. 38(6), pp. 578–589.
56. Priami, C. (2002), 'Language-based performance prediction for distributed and mobile systems', *Inf. Comp.*, Vol. 175, pp. 119–145.
57. Degano, P. and Priami, C. (1999), 'Non-interleaving semantics for mobile processes', *Theoret. Comp. Sci.*, Vol. 216, pp. 237–270.
58. Boreale, M. and Sangiorgi, D. (1998), 'A fully abstract semantics for causality in the pi-calculus', *Acta Inf.*, Vol. 35, pp. 353–400.
59. Castellani, I. (2001), 'Process algebras with localities', in 'Handbook of Process Algebra', Bergstra, J., Ponse, A. and Smolka, S., Eds, Elsevier, Amsterdam, pp. 945–1045.
60. Cleaveland, R., Parrow, J. and Steffen, B. (1993), 'The concurrency workbench: A semantics-based tool for the verification of concurrent systems', *ACM Trans Prog. Lang. Systems*, Vol. 15, pp. 36–72.
61. Victor, B. and Moller, F. (1994), 'The mobility workbench – A tool for the π-calculus', in 'Proc. CAV'94', Lecture Notes in Computer Science, Springer, Berlin, vol. 818, pp. 428–440.
62. Gilmore, S. and Hillston, J. (1994), 'The PEPA Workbench: A tool to support a process algebra-based approach to performance modelling', in 'Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation', No. 794 in Lecture Notes in Computer Science. Springer-Verlag, Vienna, pp. 353–368.
63. URL: http://wisdom.weizmann.ac.il/~biospi/
64. URL: http://www-i2.informatik.rwth-aachen.de/Research/AG/MCS/Tools/
65. Regev, A. and Shapiro, E. (2002), 'Cells as computations', *Nature*, Vol. 419, pp. 343.
66. URL: http://www.ncbi.nlm.gov/Education/
67. Chao, D. and Forrest, S. (2002), 'Information immune systems', in 'Proceedings of the First International Conference on Artificial Immune Systems (ICARIS)', pp. 132–140, (URL: http://www.aber.ac.uk/~icawww/proceedings/toc-icaris.htm).