

An Executable Biology Methodology for Systems and Synthetic Biology

J. Blakes, N. Krasnogor*, F. J. Romero-Campero, J. Twycross

¹ Automated Scheduling, Optimisation and Planning Research Group
School of Computer Science, Jubilee Campus, University of Nottingham

² Centre for Plant Integrative Biology
Sutton Bonington Campus, University of Nottingham
*contact author: Natalio.Krasnogor@Nottingham.ac.uk

Abstract. In this paper we introduce a stochastic executable biology methodology that can be used for systems and synthetic biology research. Our methodology is based on a rigorous computational formalism, P systems, that capture the three essential components of living systems: membranes, information molecules (e.g. DNA, RNA, amino acid sequences) and metabolism. The mathematical structure of our formalism allows us to build models in a modular fashion, perform very large simulations efficiently, use model checking to formally prove properties of a model and perform both model parameter and structural optimisation. We briefly mention the application of this methodology to bacterial colonies undergoing quorum sensing and plant root development.

Key words: systems biology, synthetic biology, executable biology, modularity, model generation, parameter estimation, model checking, stochastic P systems

1 Executable biology

The different approaches to modelling biological systems have been classified into *mathematical* and *computational* frameworks, the latter termed **executable biology**[1]. The semantics used in these two approaches constitutes the main difference between them. On one hand, mathematical models follow a *denotational semantics*, and consist of sets of equations which relate the quantities of the different molecular species to each other over time. These equations need to be solved or approximated numerically in order to obtain the time evolution of the system. On the other hand, the semantics of computational models is *operational*, consisting of a list of instructions producing an algorithm whose execution resembles the behaviour of the system under study. Executable biology has many appeals, for example, it not only captures the biological system's dynamics (as, e.g. a differential equations model does) but also it explicitly represents the mechanisms by which these dynamics are achieved. Moreover, executable biology models are usually (e.g. Petri nets, P systems, etc) written in a language that is familiar to biologists (e.g. by representing molecules and biochemical reactions with nodes and arcs or by rules with an antecedent and a consequent). Critically, they - unlike differential equation models that rely on very large numbers of molecules or infinitesimally small entities - can cope well with small numbers of components, spatial heterogeneity, etc.

Many formal computational approaches have been applied to systems biology, such as Petri nets[5, 3] and π -calculus[8, 6], but these tend to ignore the higher level organisation of cells in colonies and tissues, or intracellular compartmentalisation, in favour of modelling only molecular interaction networks. This can make it difficult to study the complex properties of multicellular systems that

depend on cell-cell interaction. The executable biology methodology we propose encompasses both molecular interaction networks and organismal/colony levels. That is, truly multiscale modelling becomes feasible.

Our methodology is based on *P systems*, a formalism derived from the structure and functioning of living cells[9]. P system represent molecular interactions as rewriting rules on multisets of objects and/or strings; cellular organisation is represented through a hierarchy of membranes, while the colony/tissue level is represented as collections of membranes. Our executable biology “programs”, called P system specifications, are executed by a virtual machine running an exact version of Gillespie’s stochastic simulation algorithm (SSA) extended to multiple compartments[10]. This virtual machine guarantees that rules, i.e. biochemical events, with an associated kinetic rate constant are executed at a biologically realistic rate, thus producing realistic trajectories of well-mixed chemical systems[2]. We refer to this variant as **stochastic P systems**.

2 Principles and applications of stochastic P systems

The complexity of the different scales (molecular, cellular and colony/organismal levels) in cellular systems are typically tightly interrelated. Stochastic P systems present an integrative multiscale modelling framework which explicitly specifies the different levels of cellular systems in a relevant and understandable manner. The abstractions used in our approach are closer to the biology than the abstractions of other formal approaches (for example Petri nets and (ordinary) differential equations do not have “membranes”). Within this framework we are developing modelling principles for systems biology, in particular, our chosen application domains of bacterial colonies and plant tissues:

- **Molecular species** of explicit structure such as drugs and metabolites are specified as atomic objects. Strings of symbols are used when you wish to model the internal structure of molecules as, for example, is the case in gene operons with a linear structure consisting of promoters, a linear sequence of genes, repressor binding sites, etc; or proteins with active sites or sites of post-translation modification.
- **Membranes** play a key role in the functioning and structural organisation of living cells. Their role is not limited to defining the boundaries of compartments and cells; instead they actively take part in the regulation of the metabolism and information processing in inter- and intra-cellular systems. The modelling framework based on P systems is one of the few computational approaches which explicitly specifies the role of biological membranes and compartments.
- **Molecular processes consisting of protein-protein interactions and protein translocation** are described in stochastic P systems using rewriting rules on multisets of objects. Our framework provides an intuitive rule-based biological modules library for the most common molecular interactions taking place in models of living cells: transformation and degradation of molecular species, formation and dissociation of complexes, transport of molecules across compartment membranes.
- **Gene expression control** in response to signal transduction is one of the key information processing steps of multicellular life. The specification of the processes involved in gene expression control can be represented in P systems as rewriting rules on multisets of objects or as

rewriting rules on multisets of strings and objects depending on the level of abstraction for the structural organisation of the genes used in the model. Library modules for the most common gene regulation are also included.

- In **multicellular models** the specification of the environment in which cells are located cannot be described by a single membrane since it is normally too big to be considered well-mixed from the point of view of the SSA. Therefore the environment is partitioned into a grid of small regions each of which can be considered well-mixed. Each one of these regions is represented using a compartment and connected in a graph which represents the vicinity of one region of the environment to another. We refer to this structure as a *multienvironment*[12]. A bacterial colony is described using multienvironments by distributing a number of P system models of individual bacterium among the different environments. The behaviour of the colony thus emerges from the environment-mediated interaction of thousands (or more) individual bacterium models.
- For **plant tissues**, with cells composed of a central cytoplasmic region enclosed by continuous cell wall (or apoplastic) region, we model each face of the cell wall as a different, adjacent environment. We are using executable P system models to create multiscale models which combine intracellular signalling and regulatory networks with transcellular processes such as hormone diffusion and transport.

3 Modularity

Biological functions are the results of the evolution of interactions between modules made up of interacting molecular species. By definition, a module is a discrete entity which performs a specific function that is to some extent separable (orthogonal in synthetic biology jargon) from those of other modules. Modules can be isolated or connected to each other. Isolation allows a cell to carry out many diverse processes without (potentially harmful) interference among processes, whereas connectivity allows higher level functions to be built by assembling different modules. Modules quasi-separation originates from either the chemical specificity between the molecular species, from spatial localisation in different compartments, from temporal modulation or from a combination of these factors[4].

Chemical specificity and spatial localisation can be easily specified and analysed in P systems using rewriting rules to describe chemical specificity and membranes to represent spatial localisation. Our stochastic P system modelling framework introduces the idea of *modules of rules* to characterise sets of rules which perform specific biological functions separable from the functioning of the rest of the system, thus for example one can instantiate a positive gene regulation template into two different modules, one dealing with genes A & B, and the other with genes X & Y. A P system module consists of a set of rules of the forms previously introduced representing molecular interactions which occur repetitively in many cell systems. A module is identified with a name and three sets of variables specifying the molecular species, the stochastic constants and the names of the compartments where the modules are active. Complex modules can be constructed from simple modules by applying set union.

4 Model generation and parameter estimation

Models of biological systems can have many unknowns, such as missing rate constants or undiscovered biochemical pathways. Our executable biology methodology is well suited for the automated optimisation of model structure and parameters. It is well suited for these tasks because it is syntactical and semantically rigorous as well as (as explained before) modular. We are using evolutionary algorithms to generate and optimise P system models to fit laboratory observations. Work published in [11] uses a nested genetic algorithm to first evolve the modular structure of the model, and secondly, to fine tune the stochastic rate constants of the rules in the proposed modules. We follow an incremental approach, starting with predefined modules from an elementary library of modules. New modules are generated by combining these elementary modules and those from the fittest models are in turn added to the library of modules so they can be used subsequently to develop more intricate and circuitous modular structures. A particular benefit of this combined approach is that multiple runs of the evolutionary algorithm yield different *fittest* models that vary in their structure and parameters. That is, for a given set of evidential data, our approach can suggest various alternatives to explain the data (multiple biological hypotheses to be empirically, i.e. wet lab, validated). For instance a negative autoregulation module and a positive autoregulation module can produce similar outputs depending on the rate constants involved. Validation or rejection of such hypotheses drives model development and laboratory work.

5 Model checking

Whereas deterministic models are amenable to stability and bifurcation analyses, stochastic models can be formally analysed by other means. The first route to analysis is through simulation, but we may wish to know however, given a particular system, properties of the system such as the likelihood of a gene being actively repressed at a certain time by looking at the *structure* of the model instead. Our executable biology models allows us to perform model checking, a formal and rigorous method for verifying the correctness of systems. Specifically, we are using probabilistic model checking, a variant of classical model checking augmented with quantitative information regarding the likelihood that certain transitions occur and the times at which they do. A stochastic P system model is cast into Probabilistic Computational Tree Logic (PCTL), the language of the model checking software PRISM[7], which generates a discrete time Markov chain (DTMC) of all possible states that is amenable to querying. Because this is an exhaustive approach, that is, all possible behaviours of the system are analysed, the size of the systems that can be analysed are, at the moment, quite small. We are developing ways of model checking modules to incrementally overcome this issue, thus integrating modularity of design with modularity of analysis.

6 Conclusion

In this paper we have presented a multiscale modelling framework for systems and synthetic biology models based on stochastic P systems. We have discussed how to specify the main components of cellular systems and developed a formalism for modularity arising from chemical specificity and spatial localisation. An evolutionary algorithm has been described which can automatically produce and optimise both the modular structure and parameters of stochastic P systems to fit quantitative laboratory data while producing alternative testable hypotheses.

We are currently building a software system which integrates stochastic simulation of multi-compartmental P systems with automated model assembly and model checking. We are using this system to model plant hormone transport in *Arabidopsis thaliana* and the quorum sensing circuitry in *Pseudomonas aeruginosa*. In the future we plan to expand our formalism and software system to include cell movement, growth and division for a range of topological organisations.

7 Acknowledgements

We would like to acknowledge EPSRC grant EP/E017215/1 and BBSRC grants BB/F01855X/1 and BB/D019613/1.

References

1. Fisher, J., Henzinger, T.A. (2007) Executable cell biology, *Nature Biotechnology*, **25** (11), 1239–1249.
2. Gillespie, D.T. (2007) Stochastic Simulation of Chemical Kinetics, *Annu. Rev. Phys. Chem.*, **58**, 35–55.
3. Goss, P. J., Peccoud, J. (1998) Quantitative modelling of stochastic systems in molecular biology by using stochastic Petri nets, *Proc. Natl. Acad. Sci. USA*, **95**, 6750–6755.
4. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W. (1999) From molecular to modular cell biology, *Nature*, **402**, c47–c52.
5. Koch, I., Junker, B., Heiner, M. (2005) Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber, *Bioinformatics* **21**, 1219–1226.
6. Kuttler, C., Niehren, J. (2006) Gene Regulation in the Pi Calculus: Simulating Cooperativity at the Lambda Switch, *Transactions on Computational Systems Biology VII*, **4230**, 24–55.
7. Kwiatkowska, M.Z., Norman, G., Parker, D. (2006) Quantitative Analysis With the Probabilistic Model Checker PRISM, *Electr. Notes Theor. Comput. Sci.*, **153**, 5–31.
8. Lecca, P., Priami, C., Quaglia, P., Rossi, B., Laudanna, C., Constantin, G. (2004) A Stochastic Process Algebra Approach to Simulation of Autoreactive Lymphocyte Recruitment, *Simulation*, **80**, 273–288.
9. Păun, Gh. (2002) *Membrane Computing: An Introduction* (Natural Computing Series), Springer-Verlag Berlin and Heidelberg.
10. Pérez-Jiménez, M.J., Romero-Campero, F.J. (2006) P Systems, a new computational modelling tool for systems biology, *Transactions on Computational Systems Biology VI, LNBI*, **4220**, 176–197.
11. Romero-Campero, F.J., Cao, H., Cámara, M., Krasnogor, N. (2008) Structure and Parameter Estimation for Cell Systems Biology Models, *Proc. of the Genetic and Evolutionary Computation Conference GECCO 2008*, 331–338 (awarded best paper medal).
12. Romero-Campero, F.J., Twycross, J., Bennett, M., Camara, M., and Krasnogor, N. (2008) Modular assembly of cell systems biology models using P systems. In *Proceedings of the Prague International Workshop on Membrane Computing*, Lecture Notes in Computer Science. Springer (to appear).