Funded by EPSRC & BBSRC grants
EP/E017215/1& BB/F01855X/1

Infobiotics

An Integrated Development Environment for *Synthetic Biology* Models

Jonathan Blakes[1], Francisco J. Romero-Campero[1], Jamie Twycross[2], Hongqing Cao[1], Natalio Krasnogor[1]

[1]*ASAP Research Group, School of Computer Science and IT, University of Nottingham, Jubilee Campus, Nottingham, NG8 1BB, UK. {fxc, jvb, hxc, nxk}@cs.nott.ac.uk*
[2] *Centre for Plant Integrative Biology (CPIB), Sutton Bonnington Campus, University of Nottingham, UK. jpt@cs.nott.ac.uk*

Introduction and Methodology

Synthetic Biology aims to construct biological circuits from well-characterised parts that can be integrated, orthogonally, into existing organisms to produce novel devices. It is the task of computational synthetic biologists to deliver appropriate and easy-to-use software which can enable their biological counterparts to design and test prototype such devices by modelling and simulation *in silico.*

We are developing an integrated development environment for synthetic biology models that has at its core a formal computational model of cellular processes based on an extension of *P systems*[1]. P systems are a natural computing paradigm, inspired by the structure and functioning of eukaryotic cells, in which sets of rewriting rules (reactions) operate on multisets of objects (molecular populations) within a nested membrane structure (cells and intracellular compartments). In our methodology the execution model for the application of rules is a version of Gillespie's widely-used stochastic simulation algorithm extended to multiple compartments[2]. Additionally, we have extended the P system formalism to accommodate the arrangement of cells on a regular lattice (described by a vector pair; figure 1 below) so that multicellular structures - tissues and colonies - can be modelled in space as well as time. This is formalised in our software through an XML standard (P system XML).

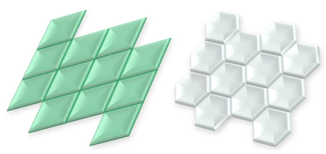


Figure 1. potential lattices

$$PostTransc(\{X\},\{c_1,\ldots,c_5\}) = \begin{cases} [\,rnaX{\cdot}RNAP\,]_b \xrightarrow{c_1} [\,rnaX\,]_b \\ [\,rnaX\,]_b \xrightarrow{c_2} [\,]_b \\ [\,rnaX\,]_b \xrightarrow{c_3} [\,rnaX + proteinX{\cdot}Rib\,]_b \\ [\,proteinX{\cdot}Rib\,]_b \xrightarrow{c_4} [\,proteinX\,]_b \\ [\,proteinX\,]_b \xrightarrow{c_5} [\,]_b \end{cases}$$

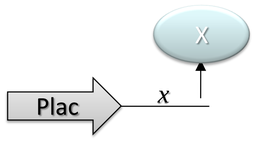
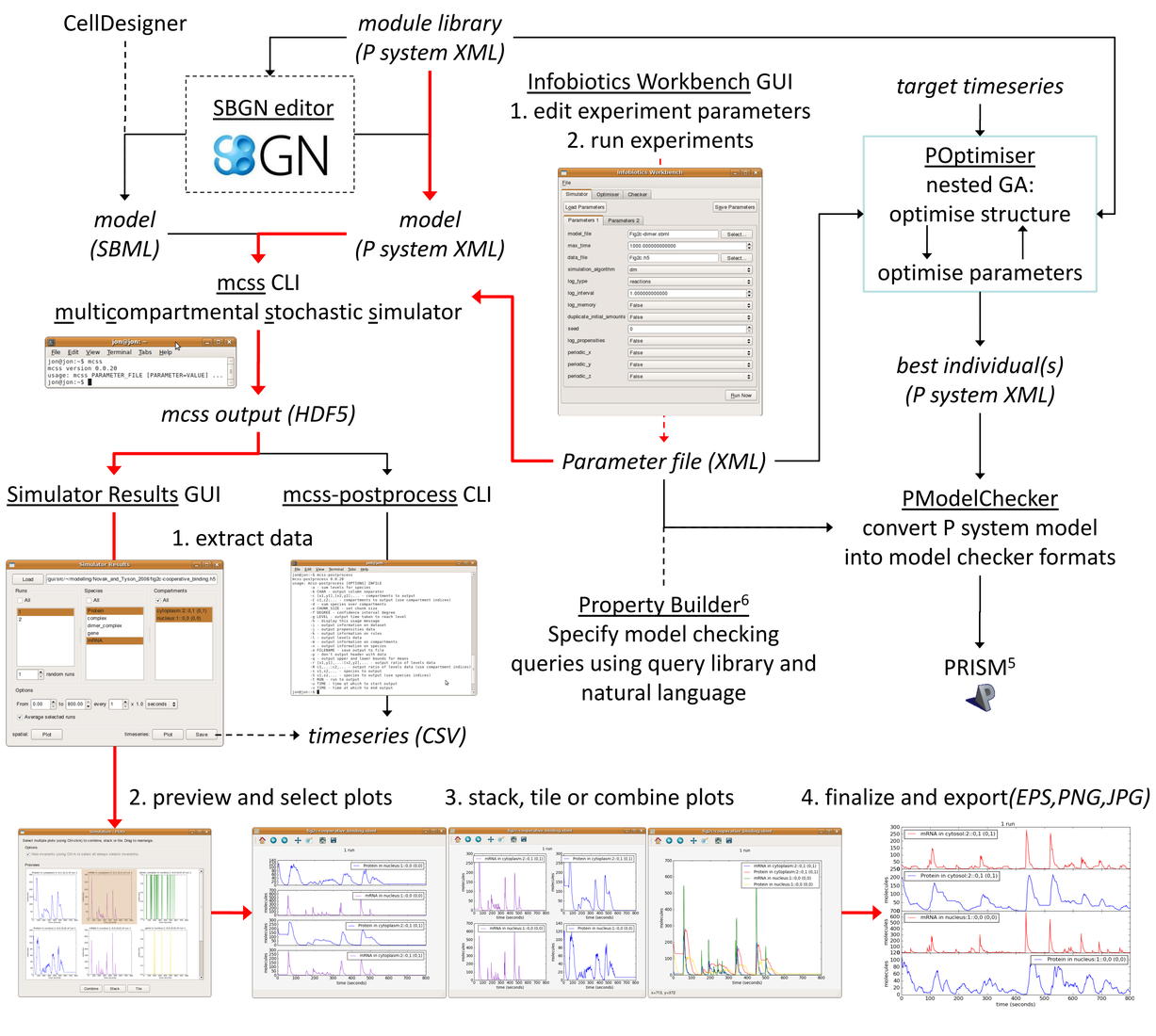


Figure 2. an example module

We also provide a means of specifying our models in a compositional and reusable manner through a modular library of rules defined in the same standard. Figure 2 shows an example module in which the production of the protein X from gene x is controlled by the promoter Plac for which a constitutive rate of expression is given. This module takes as input the identities of x and its products, and the rate constants of the products' transcription, translation and degradation. Other examples of modules include diffusion where the rules for the movement of molecules between cells and compartments are generated based on the current lattice. Modelling in this modular way not only speeds up and improves the consistency of models, it encourages the abstraction of motifs and the characterisation of their concrete components for maximum reusability.

In addition to stochastic simulation P system models can be discovered from target timeseries using structure and parameter optimisation[3] and model checked using PRISM after a conversion step. Each of these 'experiments' is run from the unified Infobiotics Workbench GUI. The Simulation Results GUI assists in extracting useful information from the simulator output and producing high quality plots.

Current Pipeline



Example: a Pulse Generating bacterium

To illustrate the use of our software we present a synthetic biology model taken from the literature: the Weiss lab's *pulse generating cells* on solid phase media[4]. The path taken through the pipeline used in these experiments is shown in red (bottom left). In this model (shown in figure 3) *sender cells* produce large quantities of N-acyl-homoserinelactone (AHL), a freely-diffusing small molecule, activating the LuxR receptor protein in the pulse generating cells where it acts as a transription factor for the Plux and PluxPR promoters. At low concentrations the LuxR-AHL complex is predominantly bound to the PluxPR promoter fused to *gfp* and the cells fluoresce. At higher concentrations it is also bound to the Plux promoter fused to *cI* and the resulting CI protein effectively represses the PluxPR promoter, switching off fluorescence.

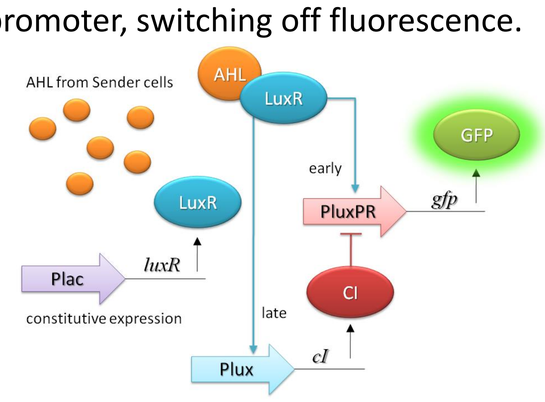


Figure 3. the pulse-generating network

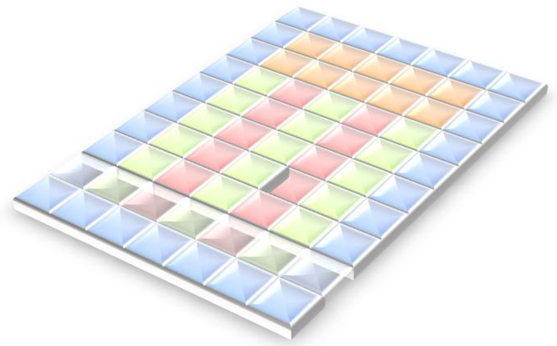


Figure 4. arrangement of the P system model

Figure 4 shows the arrangement of cells on a square lattice. Sender cells are fixed in the first rows, pulse-generator cells are arranged in columns separated by environments through which AHL diffuses freely. The system is surrounded by boundary cells that prevent AHL escaping. (The transparent row second from bottom is intended to show that system extends in this manner for many cells.)

Three-dimensional plots of quantities over the lattice were produced using the 'spatial' plotting function of the Simulation Results GUI. Figure 5 shows the large-scale production of AHL by the sender cells in the early stages of the simulation; the peaks at the front of the 'wave' are AHL in pulse-generating cells. Figures 6 and 7 show GFP levels in the pulse-generating cells over the simulation. As expected, production peaks and then declines as more CI is produced and GFP production is repressed. It is clear from figure 7 that very little GFP is produced by pulse-generating cells furthest from the sender cells. This is due partially to the sequestration and degradation of AHL in previous pulse-generating cells, but also because the system is responsive to the rate of arrival of AHL and so exhibits local effects due to the concentration gradient[4].

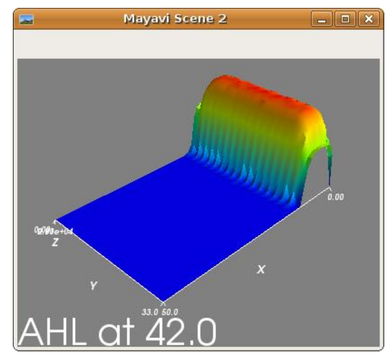


Figure 5. AHL spatial plot

Figure 6. GFP production

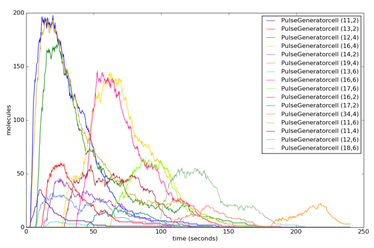
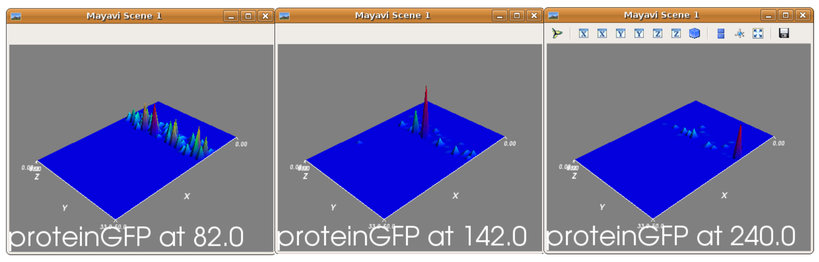


Figure 7. GFP spatial plots



Conclusions and Future Work

We have outlined our integrated approach to model development, simulation, optimisation and analysis, and demonstrated its utility with a spatial example.

Future work will focus on the development of an visual front-end for module and model specification using the SBGN standard for representation of biochemical reaction networks. This will be enhanced with facilities to overlay simulation results on the network, flag parameters for optimisation, link to bioinformatics databases and launch model checking queries on sub-models.

Proposed extensions to our underlying formalism include cell grow, division, differentiation and death to enable the modelling of development in plants and coordinated behaviour such as swarming and biofilm formation in bacteria.

References

1. Romero-Campero F. J., Twycross J., Cao H., Blakes J., Krasnogor, N. A Multiscale Modelling Framework Based On P Systems. Workshop on Membrane Computing, July 28-31, Edinburgh, United Kingdom, Lecture Notes in Computer Science, in press.
2. Romero-Campero F. J., Twycross J., Camara M., Bennett M., Gheorghe M., and Krasnogor N. Modular Assembly of Cell Systems Biology Models Using P Systems. International Journal of Foundations of Computer Science (2009) to appear.
3. Romero-Campero F., H.Cao , Camara M., and Krasnogor N. Structure and Parameter Estimation for Cell Systems Biology Models. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2008), pages 331-338. ACM Publisher
4. Basu, Mehreja, Thiberge, Chen, Weiss. Spatiotemporal control of gene expression with pulse-generating networks. PNAS (2004) 101(17): 6355-6360
5. PRISM: http://www.prismmodelchecker.org/
6. Property Builder: http://www.dcs.shef.ac.uk/~marian/teaching/projects/3rdy2008-09.html

Infobiotics website: www.infobiotic.org
SynBioNT (A Synthetic Biology Network for Modelling and Programming Cell-Chell Interactions): www.synbiont.org