Modelling and Stochastic Simulation of Synthetic Biological Boolean Gates.

Daven Sanassy, Harold Fellermann
and Natalio Krasnogor
School of Computing Science
Newcastle University
Newcastle Upon Tyne, UK
Email: natalio.krasnogor@newcastle.ac.uk

Savas Konur
Laurentiu M. Mierla
and Marian Gheorghe
School of Computer Science
University of Sheffield
Sheffield, UK
Email: m.gheorghe@sheffield.ac.uk

Christophe Ladroue
and Sara Kalvala
School of Computer Science
University of Warwick
Coventry, UK
Email: sara.kalvala@warwick.ac.uk

***Abstract*—Synthetic Biology aspires to design and compose biological systems that implement specified behaviour in engineered biological system. When designing such systems, hypothesis testing via computational modelling and simulation is vital in order to reduce the need of costly wet lab experiments. As a case study, we here discuss the use of computational modelling and stochastic simulation of engineered genetic circuits that implement Boolean AND and OR gates that have been reported in the literature. We present performance analysis results for nine different state-of-the-art stochastic simulation algorithms and analyse the dynamic behaviour of the proposed gates. Stochastic simulations verify the desired functioning of the proposed gate designs.**

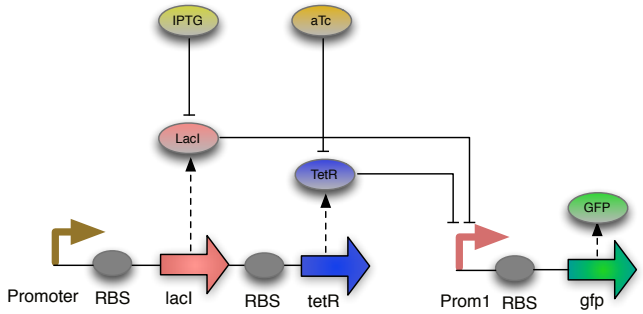
I. Introduction

Synthetic Biology is a relatively new discipline that aspires to design and compose biological systems that implement designer specified behaviour [1]. Synthetic Biology allows scientists to engineer biochemical systems that perform tasks that otherwise would not exist in nature, by harnessing mechanisms that have successfully proliferated all forms of life.

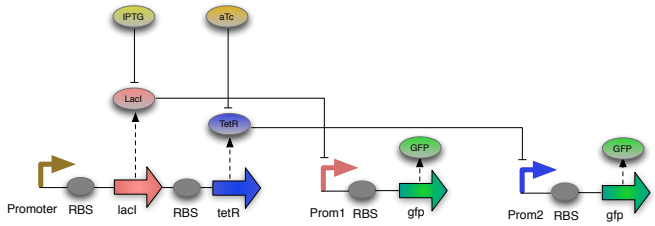
Even small biochemical systems such as bacterial cells possess many robust mechanisms that allow them to maintain homeostasis and increase their probability of survival. Life is regulated by genetic networks that encode complex behaviour. Scientists can re-use, re-configure and re-purpose these genetic mechanisms for practical applications such as natural computation. For example, synthetic cells that have been designed to perform simple computational tasks benefit from massive parallelism via biological reproduction at no extra development cost to the designer. This would result in the biological equivalent of a supercomputing cluster, where a large number of individually unremarkable compute nodes are pooled together to provide tremendous computational power.

The interactions within genetic networks are difficult for humans to comprehensively understand and mentally visualise. It is thus necessary to build models of these networks and analyse their behaviour through simulation. Building these systems in the wet lab is costly particularly with regard to man hours. Therefore, hypothesis testing via modelling and simulation is vital when engineering synthetic systems.

Biochemical systems are traditionally modelled with differential equations, however this continuous deterministic approach is not optimal when considering small genetic systems.



(a) AND gate



(b) OR gate

Fig. 1. The genetic devices functioning as an AND and OR gate. Inputs to the gates are the molecular species `IPTG` and `aTc`. The output of the gate is given by the expressed amount of `GFP` molecules. The input molecules induce expression of `GFP` to perform the desired Boolean computations. (See text for details).

Discrete stochastic modelling and simulation can capture the inherent noise present in small systems and provides a mechanistic account of the system's trajectory. Crucially, differential equations can not model genetic switches and only considers them in a constant partially-on state. On the other hand a discrete model can accurately simulate genetic switches and demonstrate the behaviour of transcription networks.

The Stochastic Simulation Algorithm (SSA) was introduced by Gillespie and produces mathematically exact system trajectories [2], [3]. This algorithm is computationally expensive and multiple runs are required to produce error bounded results. When simulations begin to take days, weeks or months

to complete, performance considerations become important. High performance computing can take advantage of the natural parallel independence of SSA runs whilst different algorithmic variants can provide large performance improvements for individual runs.

The article is structured as follows: we begin with a presentation of the Boolean gates under consideration in Section II, and of our stochastic simulation software in Section III. Results of the performance benchmarking and model behaviour are presented in Section IV, and a conclusion is given in Section V.

II. Modelling Two Synthetic Genetic Logic Gates

Synthetic Boolean logic gates have been addressed in various studies [4], [5], [6] and are of interest as the fundamental building blocks of potential biological computing. The devices discussed in this paper are constructed using the genetic subcomponents of the XOR gate designed by Beal et. al [4]. Here, we consider two important logic gates: AND and OR. Both gates use two inducers, `aTc` and `IPTG`, as inputs. `aTc` and `IPTG` inhibit the activities of `TetR` and `LacI` proteins, respectively. The genetic designs of the gates are presented in Figure 1.

Figure 1a illustrates a genetic AND gate, which receives two input signals: `aTc` and `IPTG`. In this system, the transcription factors `LacI` and `TetR` are expressed by genes controlled by a single promoter. The `aTc` and `IPTG` molecules bind to `TetR` and `LacI`, respectively, to prevent them from inhibiting the production of `GFP` by binding to the corresponding promoter which up-regulates the expression of `GFP`. If both `IPTG` and `aTc` are set to high, then neither `LacI` nor `TetR` can inhibit the `GFP` production and thus `GFP` production will be high.

Figure 1b illustrates a genetic OR gate, comprising two separate mechanisms for inducing `GFP` production. Each mechanism has a unique promoter for each of the two `GFP` genes present in the system, allowing for individual activation of either `GFP` gene. As with the genetic AND gate, `IPTG` and `aTc` are used as inputs for the genetic OR gate. The production of `GFP` in the first mechanism is repressed by `LacI` whilst the second is repressed by `TetR`. As in the AND gate `IPTG` and `aTc` regulate `LacI` and `TetR` respectively. Because there are two separate `GFP` genes present controlled by two unique mechanisms, `GFP` can be produced when `IPTG` is set to high *or* when `aTc` is set to high.

The stochastic model comprises a set of reaction channel rules governing the kinetic and stochastic behaviour of the system. Tables I and II present the rules and the kinetic constants of the devices described above. If we consider the AND gate, Rules r_1 to r_3 describe the expression the `LacI` and `TetR` proteins from `gene_LacI_TetR`, regulated by the same promoter. Rules r_4 and r_5 describe the binding of `LacI` and `IPTG` and `TetR` and `aTc`, respectively. Rules r_{6a} and r_{6b} describe the inhibition activity of `LacI`, i.e. its binding to the promoter that upregulates the `GFP` production. Rules r_{7a} and r_{7b} define the same process for `TetR`. Rule r_8 describes the expression of `GFP`. Rules r_9 to r_{12} define the degradation process of various molecular species. The input molecules `aTc` and `IPTG` are kept constant in our model to stop them being quickly consumed and thus maintain a persistent output state in the model.

TABLE I. Kinetic rules for the Boolean AND gate.

(a) AND gate

Rule		Kinetic constant
r_1 :	gene_LacI_TetR $\xrightarrow{k_1}$ gene_LacI_TetR + mrna_LacI_TetR	$k_1 = 0.12$
r_2 :	mrna_LacI_TetR $\xrightarrow{k_2}$ mrna_LacI_TetR + LacI	$k_2 = 0.1$
r_3 :	mrna_LacI_TetR $\xrightarrow{k_3}$ mrna_LacI_TetR + TetR	$k_3 = 0.1$
r_4 :	LacI + IPTG $\xrightarrow{k_4}$ LacI-IPTG	$k_4 = 1.0$
r_5 :	TetR + aTc $\xrightarrow{k_5}$ TetR-aTc	$k_5 = 1.0$
r_{6a} :	gene_GFP + LacI $\xrightarrow{k_{6a}}$ gene_GFP-LacI	$k_{6a} = 1.0$
r_{6b} :	gene_GFP-LacI $\xrightarrow{k_{6b}}$ gene_GFP + LacI	$k_{6b} = 0.01$
r_{7a} :	gene_GFP + TetR $\xrightarrow{k_{7a}}$ gene_GFP-TetR	$k_{7a} = 1.0$
r_{7b} :	gene_GFP-TetR $\xrightarrow{k_{7b}}$ gene_GFP + TetR	$k_{7b} = 0.01$
r_8 :	gene_GFP $\xrightarrow{k_8}$ gene_GFP + GFP	$k_8 = 1.0$
r_9 :	GFP $\xrightarrow{k_9}$	$k_9 = 0.001$
r_{10} :	LacI $\xrightarrow{k_{10}}$	$k_{10} = 0.01$
r_{11} :	TetR $\xrightarrow{k_{11}}$	$k_{11} = 0.01$
r_{12} :	mrna_LacI_TetR $\xrightarrow{k_{12}}$	$k_{12} = 0.001$

TABLE II. Kinetic rules for the Boolean OR gate.

(b) OR gate

Rule		Kinetic constant
$r_1 - r_5$	same as the rules $r_1 - r_5$ of the AND gate	
r_{6a} :	gene_GFP1 + LacI $\xrightarrow{k_{6a}}$ gene_GFP1-LacI	$k_{6a} = 1.0$
r_{6b} :	gene_GFP1-LacI $\xrightarrow{k_{6b}}$ gene_GFP1 + LacI	$k_{6b} = 0.01$
r_{7a} :	gene_GFP2 + TetR $\xrightarrow{k_{7a}}$ gene_GFP2-TetR	$k_{7a} = 1.0$
r_{7b} :	gene_GFP2-TetR $\xrightarrow{k_{7b}}$ gene_GFP2 + TetR	$k_{7b} = 0.01$
r_8 :	gene_GFP1 $\xrightarrow{k_8}$ gene_GFP1 + GFP	$k_8 = 1.0$
r_9 :	gene_GFP2 $\xrightarrow{k_9}$ gene_GFP2 + GFP	$k_9 = 1.0$
$r_{10} - r_{13}$	same as the rules $r_9 - r_{12}$ of the AND gate	

III. Simulation

Simulation of the stochastic models detailed in Section II is performed using the Gillespie SSA [2], [3]. This Markov Chain Monte Carlo method considers the execution of each and every reaction in the system and is thus computationally expensive. At each reaction execution, the system state vector of molecular species is adjusted and thus a time-series trajectory of the system can be logged. To perform simulations of the models described in Section II we use our *ngss* (next generation stochastic simulator) software. Ngss simulates stochastic models provided in SBML format [7] and generates time-series for all the molecular species present in the system. Time-series are outputted as either plain text comma separated values or in a compressed HDF5 format. The software is written in C++ with attention paid to performance and is compiled natively on Linux, Windows and Mac platforms. Ngss supports parallelisation of runs via OpenMP and for use on clusters with MPI. The Linux version of ngss compiles statically so that no dependencies need to be installed on cluster machines.

Ngss supports nine different variants of the SSA that each employ various optimisations in order to improve computational performance. Eight exact SSA formulations are included. These are Direct Method (DM) [3] and First Reaction Method (FRM) [2], Next Reaction Method (NRM) [8], Optimised Direct Method (ODM) [9], Sorting Direct Method (SDM) [10], Logarithmic Direct Method (LDM) [11], Partial

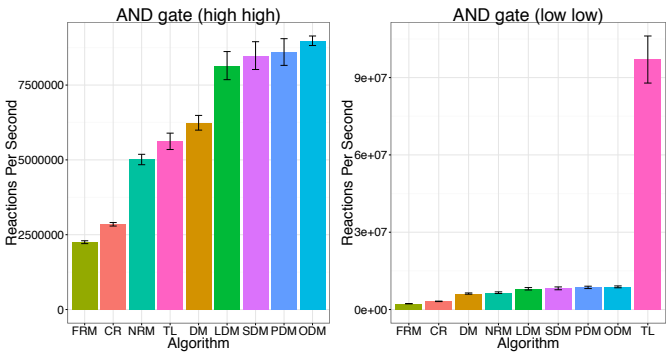


Fig. 2. Algorithm benchmark performance results in rps of each algorithm for the AND gate with `aTc` and `IPTG` in *high-high* (constant 1000 1000) and *low-low* (constant 0 0) input configuration. Each algorithm's performance was evaluated as the mean of a total of 100 runs.

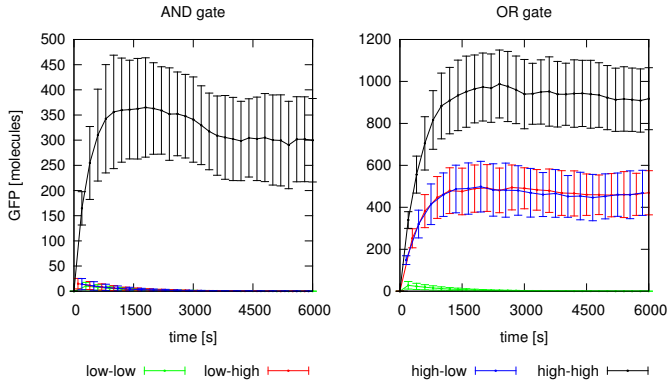


Fig. 3. `GFP` expression in the AND (left) and (OR) gate over time for the `aTc`/`IPTG` input combinations *low-low*, *low-high*, *high-low*, and *high-high*. Error bars denote the standard deviations of 100 statistically independent samples.

Propensities Direct Method (PDM) [12] and Composition Rejection (CR)) [13]. An approximation algorithm, Tau Leaping (TL) [14] is also considered.

As we are concerned with improving the simulation time for a particular model, we benchmarked the performance of each of the mentioned SSA variants for the Boolean gate models. For each algorithm, 100 runs were performed and each simulation completed to 6000 seconds of simulation time. The metric for measuring performance used is *reactions per second (rps)*. Rps is calculated by dividing the number of reactions executed by the amount of computational (process) time required. For each model we tried four different configurations of gate inputs `aTc` and `IPTG` *(high-high, high-low, low-high and low-low)* where low is zero molecules and high is 1000 molecules. Our benchmarking experiments were performed on a single core of an Intel Core i7-3770 CPU @ 3.40GHz with 8GB RAM.

IV. Results

We found that the algorithmic performance profiles of the different input combinations for both the OR gate and AND gate were very similar, with identical algorithm performance rankings per input combination. Figure 2 shows the algorithmic performance for the AND gate model in the *high-high* and

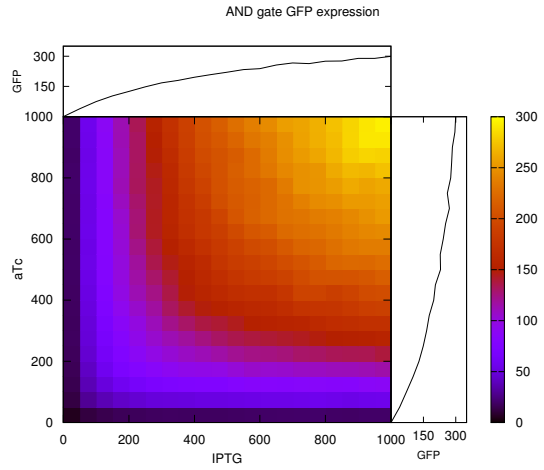


Fig. 4. Heat map visualisation of the AND gate transfer function obtained by stochastic simulation. Colors indicate `GFP` expression for different `aTc`, `IPTG` input values. The top inlay shows the steady-state response of the gate for varying `IPTG` amounts under constant $\mathtt{aTc} = 1000$, the right inlay shows the gate response for varying `aTc` under constant $\mathtt{IPTG} = 1000$.

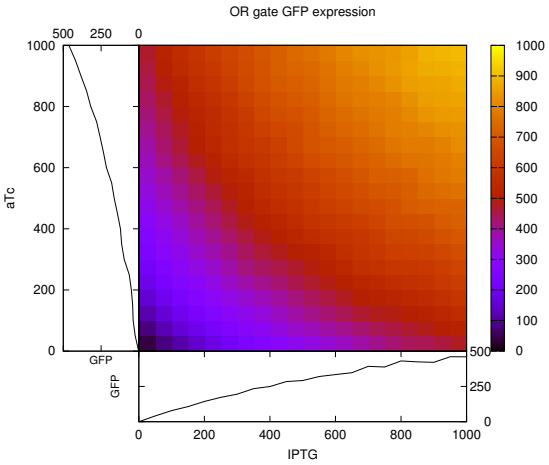


Fig. 5. Heat map visualisation of the OR gate transfer function obtained by stochastic simulation. Colors indicate `GFP` expression for different `aTc`, `IPTG` input values. The bottom inlay shows the steady-state response of the gate for varying `IPTG` amounts under constant $\mathtt{aTc} = 0$, the left inlay shows the gate response for varying `aTc` under constant $\mathtt{IPTG} = 0$.

low-low configurations. These results demonstrate that small differences in a model (in this case, the concentrations of two species) may result in large differences in algorithmic performance profiles. We can see that for the *low low* configuration TL is the fastest performing simulation algorithm, and outperforms others by an order of magnitude. However, in all other configurations ODM is the better algorithmic selection and strongly outperforms TL (see Figure 2). Whilst many SSA runs can be distributed over nodes of a HPC cluster, the minimum possible simulation time is still bounded by the time required for one run. Therefore, the selection of the fastest algorithm for simulating a given model results in a performance increase irrespective of the use of HPC facilities.

Trajectories of both gate dynamics are shown in Figure 3 for the four different input combinations of low and high `aTc` and `IPTG`. The gates quickly approach a steady state

with output concentrations that implement the desired Boolean logics. During the short transient period, GFP is produced in marginal quantities even in the absence of input signals, but this expression is suppressed once LacI and TetR repress the respective promoters and the present GFP degrades.

Figures 4 and 5 show the transfer functions (gate output for varying input values) of the AND and OR gates. In principle, the genetic AND and OR devices closely implement the requested transfer functions and express high GFP amounts under the presence of both (AND gate) or either of the two inputs (OR gate). Yet, the simulations also reveal that the gate outputs follow their inputs more or less linearly and do not implement a clear switching behavior where the output concentration would drastically change around some critical threshold input value. Depending on the application, the observed linear behavior can cause problems by accumulating errors when complicated circuits are composed by feeding the output of one gate into other downstream gates.

V. Conclusion

We have demonstrated the value of computational modelling and simulation for the design of synthetic biological systems by analysing Boolean AND and OR gates. In this case study, simulation results have verified the design of the proposed circuits, but also detected a potential weakness in the design.

We are currently improving our existing software suite, the *Infobiotics Workbench* IBW [15], that assists in the design and rapid prototyping of synthetic biological systems, by providing tools for a) the specification of synthetic biological systems, b) simulation of their behaviour, c) verification of the specification via model checking, and d) eventual compilation into nucleic acid sequences that code for genes implementing the desired behaviour. While this paper focuses on the stochastic simulation capabilities of IBW, model checking and biocompilation aspects of the software applied to the same biological systems are presented in Ref. [16].

We have also shown that even minor differences in a computational model, e.g. its boundary conditions, can result in large changes in algorithmic performance. It is important, therefore, to select the fastest algorithm for a given model. Whilst SSAs are highly parallelisible with respect to the natural parallel independence of simulation runs, minimising an individual run's computational time will result in a proportional improvement to overall simulation time. Therefore, we have developed software *(ssapredict)* [17] that makes an accurate automatic selection of the best algorithm for a given model.

Acknowledgment

This work was supported by the UK Engineering and Physical Sciences Research Council [EP/I031642/1, EP/J004111/1, EP/L001489/1].

References

[1] E. Andrianantoandro, S. Basu, D. K. Karig, and R. Weiss, "Synthetic biology: new engineering rules for an emerging discipline," *Molecular Systems Biology*, vol. 2, no. 1, Jan. 2006.

[2] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of Computational Physics*, vol. 22, no. 4, pp. 403 – 434, 1976.

[3] ——, "Exact stochastic simulation of coupled chemical reactions," *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[4] J. Beal, A. Phillips, D. Densmore, and Y. Cai, "High-level programming languages for biomolecular systems," in *Design and Analysis of Biomolecular Circuits*. Springer New York, 2011, pp. 225–252.

[5] A. Tamsir, J. J. Tabor, and C. A. Voigt, "Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'," *Nature*, vol. 469, no. 7329, pp. 212–215, 2011.

[6] S. Regot, J. Macia, N. Conde, K. Furukawa, J. Kjellen, T. Peeters, S. Hohmann, E. de Nadal, F. Posas, and R. Sole, "Distributed biological computation with multicellular engineered networks," *Nature*, vol. 469, no. 7329, pp. 207–211, 2011.

[7] M. Hucka *et al.*, "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, no. 4, pp. 524–531, 2003.

[8] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *The Journal of Physical Chemistry A*, vol. 104, no. 9, pp. 1876–1889, 2000.

[9] Y. Cao, H. Li, and L. Petzold, "Efficient formulation of the stochastic simulation algorithm for chemically reacting systems," *The Journal of Chemical Physics*, vol. 121, no. 9, pp. 4059–4067, 2004.

[10] J. M. McCollum, G. D. Peterson, C. D. Cox, M. L. Simpson, and N. F. Samatova, "The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior," *Computational Biology and Chemistry*, vol. 30, no. 1, pp. 39 – 49, 2006.

[11] H. Li and L. Petzold, "Logarithmic direct method for discrete stochastic simulation of chemically reacting systems." Department of Computer Science, University of California: Santa Barbara, Tech. Rep., 2006.

[12] R. Ramaswamy, N. Gonzalez-Segredo, and I. F. Sbalzarini, "A new class of highly efficient exact stochastic simulation algorithms for chemical reaction networks," *The Journal of Chemical Physics*, vol. 130, no. 24, p. 244104, 2009.

[13] A. Slepoy, A. P. Thompson, and S. J. Plimpton, "A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction networks," *The Journal of Chemical Physics*, vol. 128, no. 20, p. 205101, 2008.

[14] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *The Journal of Chemical Physics*, vol. 115, no. 4, pp. 1716–1733, 2001.

[15] J. Blakes, J. Twycross, F. J. Romero-Campero, and N. Krasnogor, "The Infobiotics Workbench: an integrated in silico modelling platform for systems and synthetic biology," *Bioinformatics*, 2011.

[16] S. Konur, C. Ladroue, H. Fellermann, D. Sanassy, L. Mierla, F. Ipate, S. Kalvala, M. Gheorghe, and N. Krasnogor, "Modeling and analysis of genetic boolean gates using the Infobiotics Workbench," in *Proceedings of the 2014 Workshop on Verification of Engineered Molecular Devices and Programs*, 2014, submitted.

[17] D. Sanassy, P. Widera, and N. Krasnogor, "Meta-stochastic Simulation of Biochemical Models for Systems & Synthetic Biology," in *Proceedings of The Synthetic Biology: Engineering, Evolution & Design (SEED) conference*, 2014, submitted.