D-VASim: Dynamic Virtual Analyzer and Simulator for Genetic Circuits

Hasan Baig and Jan Madsen
Department of Applied Mathematics and Computer Science
Technical University of Denmark
{haba, jama}@dtu.dk

1. INTRODUCTION

A genetic circuit represents a gene regulator network that is triggered by a combination of external signals, such as chemicals, proteins, light or temperature, to emit signals to control gene expression or metabolic pathways accordingly. In order to match the intended behaviour, genetic circuits are either assembled from a standard library of well-defined genetic gates or from parts of an available library, for instance, BioBricks. The obtained behavior can be validated through in-silico analysis, solving reaction kinetics using ordinary differential equations (ODEs) or by stochastic simulation, with the aim to reduce the number of required in-vitro experiments.

We present a behavioural simulation and analysis tool that allows the biologist to carry out virtual lab experiments as an *interactive* process during simulation of the genetic circuit, rather than a batch process, which is current practice. We believe that this increases the insights gained from the analysis and allows for exploring more parameters in an intuitive manner.

2. GENETIC CIRCUIT ANALYSIS

The Systems Biology Mark-up Language (SBML) is a standard way of representing computational biological models [1]. It is a machine-readable format, which enable models to be shared and published in a form that can be used by different software tools.

Beside the functional behavioural of the biological systems, SBML allows the user to model a sequence of input patterns in order to capture a more elaborate experiment. This is done through *events*, which describe the instantaneous, discontinuous state changes in the model [1]. For example, in genetic circuits, *events* are used to trigger the concentration of any input species to a certain level, at a specific point in time, and to observe the effects on the concentration of output species. Since *events* are predefined, they cannot be changed during runtime, which means that the output of a genetic circuit can be observed only for defined events. In order to observe the output, the different set of input conditions, i.e., when to change what input to which level, must be defined in each event. Even for moderate sized genetic circuits, capturing all combinations of inputs and concentration levels may require a very large number of events to be defined and simulated.

The ability to interact with the model, during runtime, makes it more convenient to observe the behaviour and directly make changes of input species as a reaction to the observed changes. This not only helps the user to analyse the model appropriately by triggering the concentration of input species to any level and at any instant of time, but it also makes the user free of defining long list of events for all the possible combinations of inputs in the SBML description.

There are more than 260 systems biology tools [2], which assist users in model construction and analysis. Some of these tools serve as a toolbox for commercial platforms including MATLAB, Mathematica, and Oracle; some are developed as APIs or plugins to specific software systems, while the rest are independent tools for design and simulation. A vast majority of these tools supports reading and/or writing SBML files. To the best of our knowledge, there exist no tools that allow users to trigger/change input species on the fly during the simulation, effectively creating a *virtual lab*.

3. VIRTUAL LAB SIMULATION

In the wetlab, the biologists are either provided with ready-made biological models available in test tubes or are given a specification/recipe from which to prepare it in the lab. Their duty is to analyse the model and verify its functional behaviour. This analysis is done interactively by among other things, increasing the molar concentration of input species at any instant of time and observing the effects.

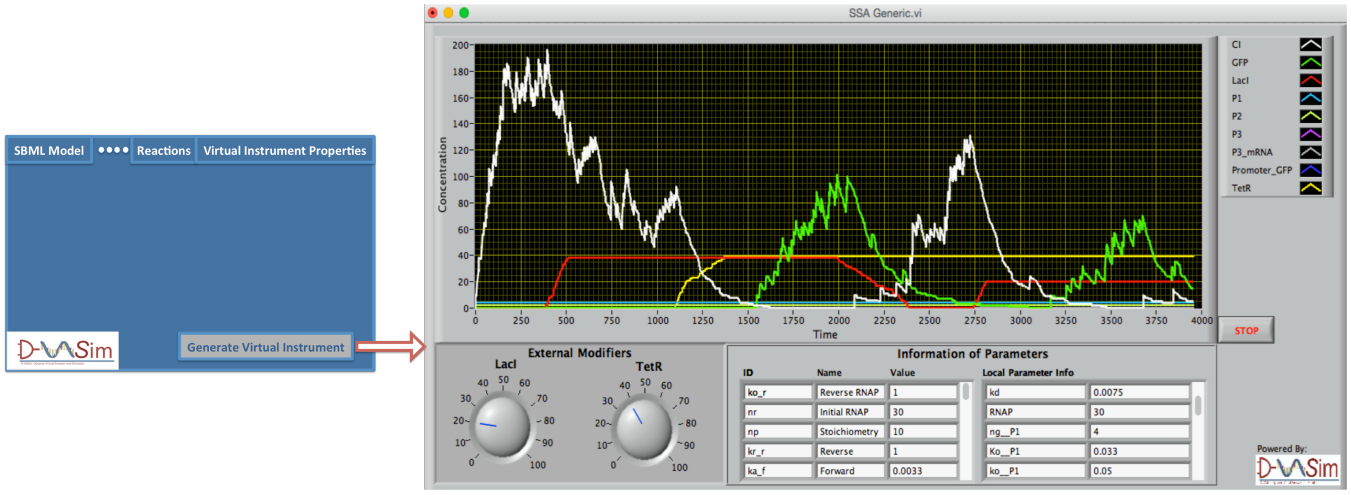
This motivated us to develop a virtual laboratory environment where users can perform interactive experiments by varying the molar concentrations during run time. This inspiration lead us to develop D-VASim (Dynamic Virtual Analyzer and Simulator), a user-friendly environment to simulate and analyse the behaviour of genetic circuit models written in SBML. D-VASim takes as input a SBML file and generates an interactive virtual instrument (VI) to simulate the behaviour of the biological model. This virtual instrument works as a standalone simulation tool for the particular SBML model. Currently D-VASim offers two types of virtual instruments, one based on solving reaction kinetics using ODEs, and the other based on stochastic simulation.

Both D-VASim and the generated virtual instrument are developed on National Instruments LabVIEW^{TM}[1] platform, which is a graphical programming platform commonly used to rapidly develop instrumentation systems for data acquisition, instrument control, and industrial automation [3].

Besides giving the biologist the feeling of being in the lab, D-VASim has also proved useful to help early-stage researchers or students, with little experience in biology, to get an intuitive feeling of the underlying biological processes and their interactions. A virtual laboratory environment is desired for such inexperienced users to observe the live biological phenomenon by varying the species concentrations without being afraid of crossing the threshold values.

D-VASim also allows user to analyse the SBML model components. Depending on the parameter settings, D-VASim generates a VI for deterministic or stochastic analysis separately. Once the instrument is generated, the user can analyse the model by varying those species' concentrations, which acts as external modifiers. This makes the VI more analogous to the real-life experimentation where the operator can increase the molar

[1] LABoratory Virtual Instrument Engineering Workbench



(a) (b)

Figure 1. D-VASim (a) Top-level diagram of D-VASim showing different tabs (b) Generated virtual instrument for stochastic simulation of genetic AND gate model.

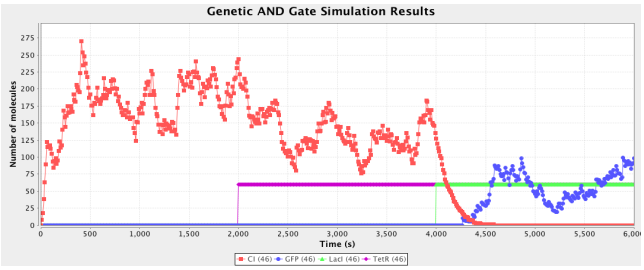


Figure 2. Static stochastic simulation plot of genetic AND gate generated by iBioSim [5].

concentration of external inputs only. Unlike the real-life experimentation, where the reaction takes place at specific rates, users can speed-up or slow-down the reaction by varying the parameter values during runtime.

4. EXPERIMENTAL RESULTS

D-VASim is tested with different example models imported from existing tools including CellDesigner [4] and iBioSim [5]. We have also tested some of the genetic gates modelled by Myers [6]. Due to the space limitations, only the results of a genetic AND gate [6] is included here.

Figure 1 shows a screen captures of D-VASim running the genetic AND gate model. Figure 1(a) shows the basic top-level diagram of D-VASim containing different tabs. For example, *Reactions tab* helps the user to analyse the reaction kinetics of the model in a user-friendly manner. Similarly, *Virtual Instrument Properties* tab allow users to set up different properties of the VI including the VI window-bounds, sizes of VI objects (knobs, graph-window etc.), deterministic or stochastic simulation, timing bounds for ODE simulation, type of continuous solver for ODE simulation etc. After setting up these properties, the VI can be generated by pressing the button *Generate Virtual Instrument* depicted in Figure 1(a). Figure 1(b) shows the virtual instrument generated for stochastic simulation of the genetic AND gate model [6]. The screen of the VI is captured during the simulation, which clearly shows the interactive stochastic simulation results. In comparison, Figure 2 illustrates the static stochastic simulation results of the same genetic AND gate generated by iBioSim [5]. As shown in Figure 2, the events, to trigger the number of molecules of TetR and LacI to 60, are predefined to be activated at time 2000 and 4000 units respectively. Also, the simulation runs for a predefined interval, 6000 time units in this example. From Figure 2, it can be observed that the production of GFP (blue curve) starts when the number of molecules of both the inputs, TetR and LacI, reaches at the same level i.e. 60. It is, however, more evident in Figure 1(b) that the production of GFP (green curve) starts even when the concentration of both the inputs are not same (at time unit 3200). Therefore, to observe the behaviour of combinatorial genetic logic circuits more clearly, either all the possible combinations of inputs with all possible concentration levels should be defined in the list of SBML *events* – a tedious task, or the model's behaviour should be examined in the interactive environment. It may also be possible to generate the list of SBML *events* by running pre-written scripts with minimal efforts, but the idea of interacting with the model during runtime gives the insight of performing live virtual lab experiments. Hence the significance of a run-time interactive simulation environment, like D-VASim, is more obvious as it helps the user to analyse the model more easily and explore its parameter space intuitively.

5. SUMMARY

We are currently working on an algorithm to make the D-VASim capable of extracting the Boolean expression from the interactive simulation. It will help students and scientists to validate if the genetic circuit model behaves as expected. In future, we plan to incorporate a Boolean logic minimization tool for genetic cost reduction, which will specifically be helpful when building cascaded genetic circuits.

6. REFERENCES

[1] The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 1 Core, October 06, 2010.

[2] Systems Biology Mark-up Language Software Matrix, http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix.

[3] NI LabVIEW, http://www.ni.com/labview/.

[4] Funahashi, A.; Matsuoka, Y.; Jouraku, A.; Morohashi, M.; Kikuchi, N.; Kitano, H. "CellDesigner 3.5: A Versatile Modeling Tool for Biochemical Networks" Proceedings of the IEEE Volume 96, Issue 8, pp. 1254 – 1265 Aug. 2008.

[5] C. Madsen, C. Myers, T. Patterson, N. Roehner, J. Stevens, and C.Winstead, "Design and test of genetic circuits using iBioSim," IEEE Design and Test, 29(3): pp. 32-39, May/June 2012.

[6] Chris J. Myers, "Engineering Genetic Circuits", Chapman & Hall/CRC Press, July 2009.