

Simulating genetic networks made easy: Network construction with simple building blocks

Steven Vercruyse¹ and Martin Kuiper¹

¹Department of Plant Systems Biology, Flanders Interuniversity Institute for Biotechnology (VIB), Ghent University, Technologiepark 927, B-9052, Gent, Belgium

ABSTRACT

Summary: We present SIM-plex, a genetic network simulator with a very intuitive interface in which a user can easily specify interactions as simple “if-then” statements. The simulator is based on the mathematical model of *Piecewise Linear Differential Equations*. With PLDEs, genetic interactions are approximated as acting in a switch-like manner.

Availability: The Java program, examples and a tutorial are available at <http://www.psb.ugent.be/cbd/>

Contact: {stcru,makui}@psb.ugent.be

INTRODUCTION

As gathering data on interactions in genetic networks today is almost trivial, the need for efficient software tools to model and simulate the network behaviour is becoming increasingly important. For example in the study of the regulatory interaction network of cell cycle control, yeast two-hybrid experiments, protein-DNA interaction array experiments, and transcript microarrays yield data that may supplement well-described interactions between core components. In a systems biology approach, the objective is to integrate all these pieces of knowledge into a functional model of the regulatory process. This typically involves mathematical modeling of a (piece of the) genetic network to simulate its behaviour, and then using the simulation results to design the next experiments. However, because of a lack of precise parameter information, the use of exact differential equations, as used in Novak *et al.* (2001), in practice is rarely realistic or feasible. Moreover, such complex equations also put exact mathematic modeling beyond the reach of researchers primarily trained in biology.

We present SIM-plex: a simulator based on PLDEs, that are a simplified mathematical model where genes ‘turn each other on or off’ like switches (de Jong *et al.*, 2004). We focused on building an intuitive interface in which the user can declare each interaction as a simple “if-then” statement (Figs. 1 and 2). The result of a quantitative simulation can be a graph of product amounts varying over time. The overall shape of the plots and the relative product amounts can be much more important than the

exact values, although they can be tuned too when more precise measurements become available. In this way, the simulator can serve as a bridge between qualitative (states, relative values only) and quantitative (numerical values) modeling, whereas its intuitive interface may bridge a gap between computational modeling and wet-lab biologists.

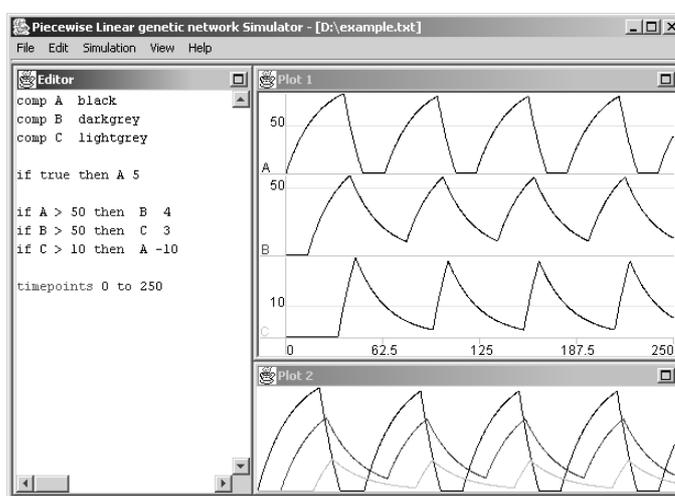


Fig. 1. The simulator’s graphical interface. The definition of a network, shown left, results in product amount profiles in the top-right window. The bottom-right window shows a combined view.

METHOD AND IMPLEMENTATION

Network definition

Activation of a gene can be the net result of many underlying factors. Several activating or attenuating proteins may bind to its promoter, ultimately determining the rate of transcription. Such activation or attenuation can be approximated to occur almost stepwise (Glass & Kauffman, 1973). An easy way to formulate this is a statement like: “if transcription factor A rises above a certain threshold, and repressor B is below another threshold, then the transcription of C is switched on.” This statement is one contribution to the set of “if-then” statements that together define the total synthesis rate of the product C. The simulator that we present offers an editing window where one can

define a genetic network by simply stacking such statements. The above statement will look like:

```
if A>thr1 and B<thr2 then C rate1
```

The exact values of these thresholds and creation rate can usually merely be estimated (as in Novak *et al.*, 2001), until more exact experiments are carried out. The determination of a set of parameters that are able to make a network behave according to biology already constitutes a major accomplishment, and can directly be used to devise new hypotheses about biological process components.

In case a transcription factor ‘activates’ a gene, and an additional one enhances this transcription, one can define two statements that work additively, such as:

```
if A>thr1 then C 5
```

```
if A>thr1 and B>thr2 then C 2
```

Their effects will be summed and yield one combined differential equation for C. When both A and B increase beyond their thresholds, this results in the additive production of C at a rate of 7 units per time unit. Conversely, if B strongly inhibits the transcription of C, one could define:

```
if A>thr1 then C 5
```

```
if B>thr2 then C block 0.10
```

which will decrease the production of C to 10%. A statement like `... then C -3`, can be used to consume already produced C (also see website). Note that PLDEs can also include a constant degradation rate to mimic natural breakdown by proteases, nucleases, etc.

A number of additional statements are provided to further facilitate network assembly (see website). One example is: `if A>thr1 then transform X to Y 5`. This entry is sufficient to specify a kinase that transforms a protein to its activated form. Before simulation starts, this statement is actually translated into two normal “if-then” statements: `... then X -5` and `... then Y 5`. As a consequence, the approach with stepwise activation is less precise, but initial applications of the tool indicate that it is very useful as a first approximation.

A last feature was built to address the problem that two gene products that are subject to two different regulation backgrounds can have a similar, additional regulatory effect on a third gene. We have implemented a small extension on PLDEs, by enabling the definition of virtual, intermediate products in the simulator, such as `virtprod C = 1*A + 0.5*B`. Here, A will have twice the regulation power as B, on genes that are regulated by the virtual C.

Graphical interface

The syntax-highlighting editor offers a programming environment-like ease of use where upon adding network components one keystroke will start simulation. Thus, at any time in the network architecture building, one can view the intermediate performance. Effects of mutations or deletions can be immediately validated, offering a potentially very powerful tool to experimental biologists.

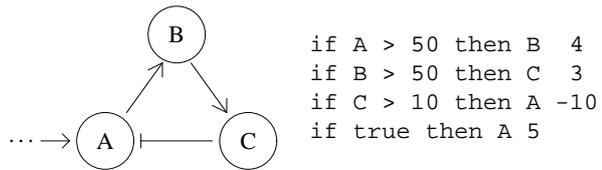


Fig. 2. Network definition example. The figure shows the graphical and (quantitative) textual representation of an example network. The final statement gives rise to a constant production of A.

DISCUSSION

We have developed a simulator that allows genetic network construction by intuitively stacking components defined in simplified building blocks. By choosing to only approximate interaction parameters and to simplify the interactions, we realise that the network outcome will only be an approximation of real life biology. However, we could successfully reproduce the results of Novak *et al.* (2001) on the cell cycle control of fission yeast, replacing complex differential equations by simplified building blocks (manuscript in preparation). Furthermore, in many cases a basic understanding of network topology may already be sufficient to test biological hypotheses, making this a potentially powerful tool for systems biology. With the current graphical interface, the user can also easily explore and ‘play with’ different network configurations.

Future prospects

We are currently applying the software as a validation tool for the *Arabidopsis* cell cycle control network, to exploit its use as a hypothesis-generating engine and to identify novel components in this network. Our longer-term objective is to implement this tool in an automated network validation pipeline. As a preparation, the functionality was built in to allow command-line control of the simulator, or to call Java methods directly.

S.V. is indebted to the IWT for a predoctoral fellowship. We thank Hidde de Jong for valuable comments.

REFERENCES

- de Jong,H., Gouzé,J.-L., Hernandez,C., Page,M., Sari,T. and Geiselman,J. (2004) Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol.*, **66**, 301–340.
- Glass,L. and Kauffman,S. (1973) The logical analysis of continuous non-linear biochemical control networks. *J. Theor. Biol.*, **39**, 103–129.
- Novak,B., Pataki,Z., Ciliberto,A. and Tyson,J. (2001) Mathematical model of the cell division cycle of fission yeast. *Chaos*, **11**, 277–286.