

# BooleSim: an interactive Boolean network simulator

Matthias Bock<sup>1</sup>, Till Scharp<sup>1</sup>, Chaitanya Talnikar<sup>2</sup> and Edda Klipp<sup>1,\*</sup><sup>1</sup>Department of Biology, Theoretical Biophysics, Humboldt-Universität zu Berlin, 10115 Berlin, Germany and <sup>2</sup>Chemical Engineering Department, Indian Institute of Technology Bombay, Mumbai 400 076, India

Associate Editor: Igor Jurisica

## ABSTRACT

**Summary:** BooleSim (Boolean network simulator) is an open-source in-browser tool for simulation and manipulation of Boolean networks. It was developed mainly during Google's Summer of Code 2012 and uses the biographer project for network visualization. It can be used specifically for the modeling of gene regulatory or signal transduction networks.

**Availability and implementation:** BooleSim is free software and can be downloaded from GitHub (<https://github.com/matthiasbock/BooleSim>). Online version available at <http://rumo.biologie.hu-berlin.de/boolesim/>.

**Contact:** edda.klipp@rz.hu-berlin.de

Received on March 4, 2013; revised on September 13, 2013; accepted on September 24, 2013

## 1 INTRODUCTION

Since Kauffman (1969) introduced Boolean networks for biological simulations, they are extensively used to describe dynamic network behavior, especially when networks are big, experimental data scarce or only qualitatively available [i.e. cell cycle (Waltermann *et al.*, 2010), signaling (Saez-Rodriguez *et al.*, 2007) or stem cell differentiation (Flöttmann *et al.*, 2012)].

Boolean networks can be visualized as a graph consisting of a set of nodes representing the species of the network plus a set of edges representing the interactions between these species. In Boolean networks, nodes can only take the discrete values 0 and 1. Biological entities are thus exclusively viewed as active or inactive, phosphorylated or unphosphorylated, expressed or not expressed and nothing in between. This is a simplification of reality, as biochemical species can be present in a continuous range of concentrations. However, signals are often transmitted via threshold crossings. Thus it is often appropriate to discriminate between two states—below and above the threshold—which have different qualitative characteristics. Transitions between these states are defined by Boolean logical operations that can account for activation, inhibition and more complicated cooperative interactions. Thus the use of Boolean networks in mathematical modeling of biological processes is justifiable by its practicality (Fauré *et al.*, 2006; Waltermann *et al.*, 2010).

Although software packages such as BoolNet for R (Müssel *et al.*, 2010) or BooleanNet for Python (Albert *et al.*, 2008) are available to simulate and analyze Boolean networks, these packages require downloading, installation and a certain basic knowledge of programming. Moreover, manipulating dynamic

simulations, i.e. changing initial conditions or perturbing a steady state, often requires advanced scripting abilities.

Therefore, we wrote a simulator that runs visually in a browser and does not require downloading or installation. BooleSim, the Boolean network simulator, supports import of common model file formats, model simulation, easy manipulation through on-click functionality and visualization of a network's dynamical properties as well as export of the Boolean model, graphical network view and the time series.

## 2 METHODS AND FUNCTIONALITY

BooleSim supports importing Boolean networks from Python BooleanNet, R BoolNet and jsBGN (Krause *et al.*, 2013) formats. These are text based and define the nodes of the network and the Boolean update rules. During model import, a node is created for every variable in the model. After that, edges are created between every two nodes, which influence each other. This interaction can be positive, which biologically corresponds to stimulation or activation, or negative, which corresponds to inhibition.

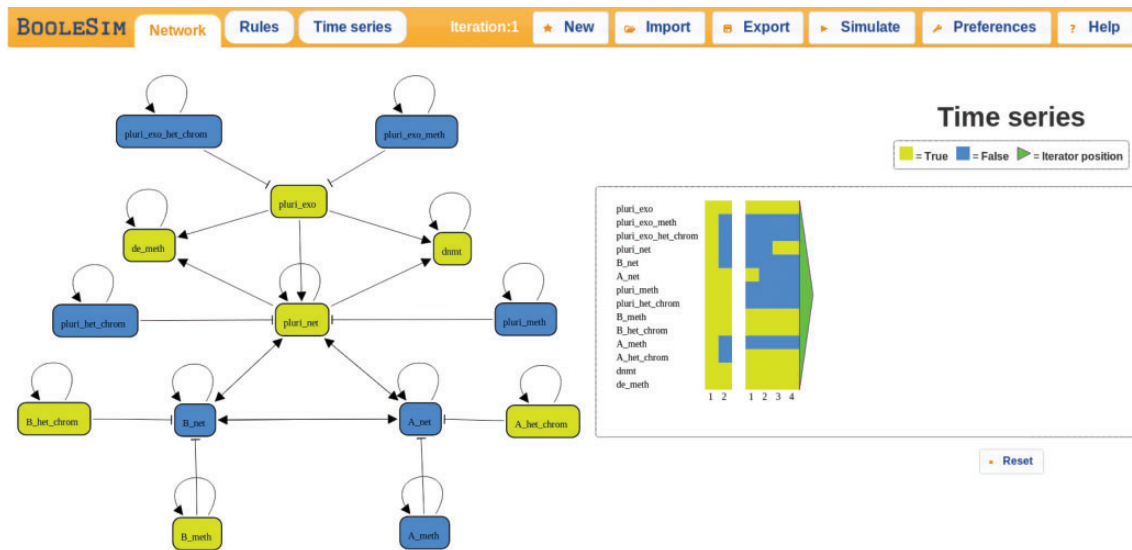
BooleSim uses modern web technologies, such as HTML5 and JavaScript to deliver a highly interactive BooleSim. The network is rendered as a dynamic scalable vector graphics (SVG), generated during runtime using the biographer-user interface (UI) interface (Krause *et al.*, 2013). A force-directed graph layout algorithm, derived from the d3 gravity/repulsion-based algorithm (<http://d3js.org/>), arranges the nodes based on the size of the labels and length of the edges.

The color of each node in the network graph denotes its respective state. Yellow refers to active and blue to inactive states. A node's state can be switched by clicking it, nodes can be added or deleted, whereas Boolean update rules can be edited in a separate text box tab. This manipulation is internally evaluated, translated into a new network layout and accounted for in the next simulation steps.

As simulation progresses, nodes change color in a smooth transition effect, which is making use of jQuery features (<http://jquery.com/>). The simulation stops once a steady state is reached. In case a system converges into a cyclic attractor, the simulation continues until the user stops it by clicking the simulate/pause button.

The evolution of the node's states can be monitored in the time series tab: On the left side of the graph, network nodes are listed in rows. Each column represents the state of the corresponding node in one iteration, where time is progressing to the right. In this heatmap representation, yellow represents active and blue

\*To whom correspondence should be addressed.



**Fig. 1.** BooleSim functionality: shown are the demo network [submodel from Flöttmann *et al.* (2012) as explained in the text] and the heatmap of the simulation obtained by clicking the time series tab and introducing a minimal set of perturbations to 'reprogram' the network to a pluripotent state

inactive node states. Beneath its graphical representation of the time course, the time series heatmap functionality is also helpful e.g. to identify cyclic attractors.

To enhance interoperability, BooleSim supports exporting networks to Python BooleanNet or R BoolNet formats, export to and reimport from biographer's internal exchange format jSBGN allowing storing the graph alongside its update rules together in one file. The network graph and time series can also be saved as SVG vector image.

### 3 RESULTS

To demonstrate our tool, we imported a submodel of a published probabilistic Boolean network (PBN), describing epigenetic and transcriptional dynamics in somatic cell reprogramming (Flöttmann *et al.*, 2012). To transform the PBN into a deterministic Boolean network, we combined the update rules with the highest probability for every species. Results are shown in Figure 1.

The time series represents a reprogramming process, i.e. the transition from a differentiated cell lineage A to the pluripotent cell state and the minimal set of manipulations that were effected to enable this transition with the simplified synchronous Boolean network (see Fig. 1).

### 4 CONCLUSION

We presented a Boolean modeling tool that is easy to use and provides users with a qualitative insight into dynamic behavior of biological networks. As the first tool for that purpose, it has in-browser functionality and works cross-platform. It enables users—without the need for downloading but simply by opening their browser (Chromium recommended) and visiting the Web site <http://rumo.biologie.hu-berlin.de/boolesim/>—to load their network and perform simulations and manipulations through clicking on the colored nodes. BooleSim is completely open

source and licensed under the free software license GNU Affero General Public License (GPL) version 3. Source code and off-line version are available on the GitHub repository (<https://github.com/matthiasbock/BooleSim>).

In the future, BooleSim could be extended to include stochasticity through asynchronous or PBNs, to find attractors or to represent state spaces.

### ACKNOWLEDGEMENTS

This tool was mainly developed in Google's Summer of Code 2012 (GSoC'12). The authors would also like to thank the developers of biographer for helpful discussions and Ivo Maintz for maintaining an online version of our software.

**Funding:** Google, Bundesministerium für Bildung und Forschung (BMBF) in the context of GerontoMitoSys (FKZ0315584B) and DrugIPS (FKZ0315398F).

**Conflict of Interest:** none declared.

### REFERENCES

- Albert, I. *et al.* (2008) Boolean network simulations for life scientists. *Source Code Biol. Med.*, **3**, 16.
- Fauré, A. *et al.* (2006) Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, **22**, e124–e131.
- Flöttmann, M. *et al.* (2012) A stochastic model of epigenetic dynamics in somatic cell reprogramming. *Front. Physiol.*, **3**, 216.
- Kauffman, S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.
- Krause, F. *et al.* (2013) Biographer: web-based editing and rendering of SBGN compliant biochemical networks. *Bioinformatics*, **29**, 1467–1468.
- Müssel, C. *et al.* (2010) Boolnet - an R package for generation, reconstruction, and analysis of Boolean networks. *Bioinformatics*, **26**, 1378–1380.
- Saez-Rodriguez, J. *et al.* (2007) A logical model provides insights into T cell receptor signaling. *PLoS Comput. Biol.*, **3**, e163.
- Waltermann, C. *et al.* (2010) G1 and g2 arrests in response to osmotic shock are robust properties of the budding yeast cell cycle. *Genome Inform.*, **24**, 204–217.