BIOINFORMATICS APPLICATIONS NOTE Vol. 26 no. 10 2010, pages 1378–1380
doi:10.1093/bioinformatics/btq124

Systems biology

Advance Access publication April 7, 2010

BoolNet—an R package for generation, reconstruction and analysis of Boolean networks

Christoph Müssel[1,†], Martin Hopfensitz[2,†] and Hans A. Kestler[1,2,*]

[1]Institute of Neural Information Processing, University of Ulm and [2]Department of Internal Medicine I, University Hospital Ulm, 89081 Ulm, Germany

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: As the study of information processing in living cells moves from individual pathways to complex regulatory networks, mathematical models and simulation become indispensable tools for analyzing the complex behavior of such networks and can provide deep insights into the functioning of cells. The dynamics of gene expression, for example, can be modeled with Boolean networks (BNs). These are mathematical models of low complexity, but have the advantage of being able to capture essential properties of gene-regulatory networks. However, current implementations of BNs only focus on different sub-aspects of this model and do not allow for a seamless integration into existing preprocessing pipelines.

Results: *BoolNet* efficiently integrates methods for synchronous, asynchronous and probabilistic BNs. This includes reconstructing networks from time series, generating random networks, robustness analysis via perturbation, Markov chain simulations, and identification and visualization of attractors.

Availability: The package *BoolNet* is freely available from the R project at http://cran.r-project.org/ or http://www.informatik.uni-ulm.de/ni/mitarbeiter/HKestler/boolnet/ under Artistic License 2.0.

Contact: hans.kestler@uni-ulm.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on November 25, 2009; revised on March 3, 2010; accepted on March 17, 2010

1 BACKGROUND

A popular class of models for describing gene regulation are Boolean networks (BNs; Kauffman, 1969, 1993). Here, genes are modeled as Boolean variables that exhibit a simple bistable 'ON/OFF' behavior, i.e. transcribed or not, encoded as 1 and 0. This qualitative approach constitutes an abstract, but intuitive representation of interactions. BNs can approximate the main dynamic properties of gene-regulatory networks, while being of simple structure (Bornholdt, 2005): it is assumed that concentration levels in gene-regulatory networks behave according to a Hill function (de Jong, 2002). Boolean functions approximate the sigmoidal behavior of this function by the step function. *BoolNet* supports three kinds of BNs: *synchronous BNs* consist of a set of Boolean variables (genes) and a set of transition functions, one for each variable (Kauffman, 1969, 1993). The next state of the network is calculated by applying *all* transition functions synchronously. In *asynchronous BNs*, only *one* of the transition functions is chosen at random at each point of time, and the corresponding Boolean variable is updated (Harvey and Bossomaier, 1997). *Probabilistic BNs* allow for specifying more than one transition function per variable (Shmulevich *et al.*, 2002). Each of these functions has a probability to be chosen, where the probabilities of all functions for one variable sum up to 1.

Simulations of BNs can provide insight into the dynamics of gene-regulatory networks (Bornholdt, 2005). Several genetic networks have been successfully modeled and analyzed using BNs, such as the mammalian cell cycle (Fauré *et al.*, 2006), or the yeast cell cycle (Li *et al.*, 2004).

Existing software tools in this field often specialize on certain aspects of BN research, or do not support all three types of networks (e.g. Albert *et al.*, 2008; Klamt *et al.*, 2007; Wuensche, 2009, or the BN/PBN toolbox by Lähdesmäki and Shmulevich at http://personal.systemsbiology.net/ilya/PBN/PBN.htm). The R package *BoolNet* provides methods for all major uses of synchronous, asynchronous and probabilistic BNs and includes novel functions for attractor search, robustness analysis and binarization. The application supports reconstruction of networks from time series, parsing networks specified by human experts, generation of random networks, perturbation of networks and identification of attractors. Genes can be temporarily knocked out and overexpressed. In addition, convenient visualization methods are provided. *BoolNet* integrates well with existing modeling tools, such as BioTapestry (Longabaugh *et al.*, 2005) and Pajek (Batagelij and Mrvar, 1998).

2 FUNCTIONALITY

There are several ways of assembling BNs: one alternative is to reconstruct a network from time series of gene measurements. Our package includes the reconstruction algorithms *Best-Fit Extension* (Lähdesmäki *et al.*, 2003) and *REVEAL* (Liang *et al.*, 1998). If necessary, methods for binarization of real-valued measurements are available for preprocessing the time series. Furthermore, BNs can be constructed from natural-language statements, e.g. from literature knowledge on the dependencies of genes. Our software supports reading in networks as collections of Boolean formulae from files. *BoolNet* also imports networks from BioTapestry (Longabaugh *et al.*, 2005), a popular application for visual modeling of gene-regulatory networks.

The package includes an innovative facility to generate various kinds of random BNs (Aldana, 2003; Kauffman, 1969, 1993).

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

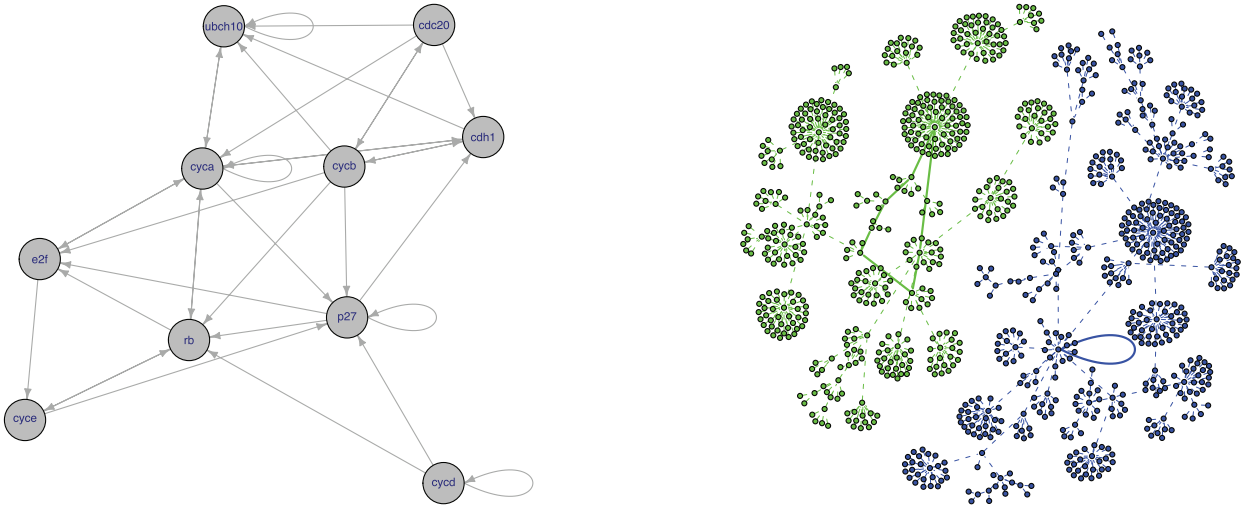

Fig. 1. (**A**) Dependencies among the genes in the mammalian cell-cycle network (Fauré *et al.*, 2006). (**B**) Visualization of the basins of attraction in the same network. Each node represents one state, and each line represents a state transition. Different colors mark different basins of attraction, and bold lines highlight the state transitions that belong to the attractors. Here, the green states belong to the basin of attraction of a dynamical cycle of seven states characterized by the presence of CycD, and the blue states lie in the basin of attraction of a single stable state, which is attained when CycD is absent.

Different techniques such as generating scale-free networks, networks with fixed in-degree, biased Boolean functions and a combination of those are offered. Furthermore, existing networks can be perturbed to test the robustness of structural properties to noise. The package supports commonly used methods, such as flipping single bits in the input functions, but also introduces non-standard approaches, such as perturbing states in the transition table of a network and constructing a new network from this table. Based on such randomly generated and perturbed networks, *BoolNet* offers a generic and extensible interface for computer-intensive tests to identify specific properties of biologically meaningful networks (see the Supplementary Methods).

Moreover, knock-out and overexpression experiments can be simulated by setting genes in a network to fixed values without touching their transition functions.

In the context of BNs, the identification and analysis of attractors is a major task. Attractors are stable cycles of states. As they comprise the states in which a gene-regulatory network resides most of the time, they carry strong biological implications and often can be linked to phenotypes (Kauffman, 1993; Li *et al.*, 2004). Our software supports several methods to identify attractors: synchronous attractors can be identified by exhaustive search of all 2^n states (for n genes), or by a heuristic search starting from a number of predefined or randomly chosen states. In addition, a new random walk algorithm for the identification of complex asynchronous attractors is provided. For synchronous and probabilistic networks, potential attractor states and probabilities of reaching certain states can also be calculated using Markov chain simulations (Shmulevich *et al.*, 2002).

Gene dependencies can be visualized as a graph. Additionally, the package provides methods to visualize the basins of attraction and the transitions of attractors. Some of the plots are shown in Figure 1. For further visualizations, the state graph can also be exported to the Pajek file format (Batagelij and Mrvar, 1998).

Time-critical algorithms, such as the reconstruction algorithms, the identification of attractors and the Markov chain simulations, were implemented in ANSI C, ensuring a high performance through the use of bit vectors. For all functions, R interfaces are supplied. The integrated manual can be accessed using the R command `help(package=BoolNet)`. Moreover, an extensive step-by-step tutorial of all important aspects of the package can be accessed using the command `vignette("BoolNet_package_vignette")`.

Exemplary uses of BoolNet: the package includes the mammalian cell cycle network (Fauré *et al.*, 2006) as an example. The network can be loaded via

```
> data(cellcycle)
```

and is now stored in a variable called `cellcycle`. The gene dependencies of the network can be visualized using

```
> plotNetworkWiring(cellcycle)
```

This graph is depicted on Figure 1A. To identify all synchronous attractors in this network, we call

```
> attr <- getAttractors(cellcycle)
```

The resulting structure contains information on the attractors as well as the transition table of the network. The transition table can now be visualized as depicted on Figure 1B:

```
> plotStateGraph(attr)
```

ACKNOWLEDGEMENTS

We thank Paul Spellman for the permission to include a preprocessed version of the yeast cell cycle time series in the package (Spellman *et al.*, 1998).

Funding: German Science Foundation (SFB 518, Project C5); Stifterverband für die Deutsche Wissenschaft (to H.A.K.); Graduate School of Mathematical Analysis of Evolution, Information, and Complexity, University of Ulm (to C.M. and H.A.K.).

Conflict of Interest: none declared.

REFERENCES

Albert,I. *et al*. (2008) Boolean network simulations for life scientists. *Source Code Biol. Med.*, **3**, 16.

Aldana,M. (2003) Boolean dynamics of networks with scale-free topology. *Physica D*, **185**, 45–66.

Batagelij,V. and Mrvar,A. (1998) Pajek – program for large network analysis. *Connections*, **21**, 47–57.

Bornholdt,S. (2005) Systems biology. Less is more in modeling large genetic networks. *Science*, **310**, 449–451.

de Jong,H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comp. Biol.*, **9**, 67–103.

Fauré,A. *et al*. (2006) Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, **22**, e124–e131.

Harvey,I. and Bossomaier,T. (1997) Time out of joint: attractors in asynchronous random Boolean networks. In Husbands,P. and Harvey,I. (eds) *Proceedings of the Fourth European Conference on Artificial Life*. MIT Press, Cambridge, MA, USA, pp. 67–75.

Kauffman,S.A. (1969) Metabolic stability and epigensis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.

Kauffman,S.A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, NewYork.

Klamt,S. *et al*. (2007) Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Syst. Biol.*, **1**, 2.

Lähdesmäki,H. *et al*. (2003) On learning gene regulatory networks under the Boolean network model. *Mach. Learn.*, **52**, 147–167.

Li,F. *et al*. (2004) The yeast cell-cycle network is robustly designed. *Proc. Natl Acad. Sci. USA*, **101**, 4781–4786.

Liang,S. *et al*. (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Pac. Symp. Biocomp.*, **3**, 18–29.

Longabaugh,W.J.R. *et al*. (2005) Computational representation of developmental genetic regulatory networks. *Dev. Biol.*, **283**, 1–16.

Shmulevich,I. *et al*. (2002) Probabilistic Boolean networks: a rule-based uncertainty model for gene-regulatory networks. *Bioinformatics*, **18**, 261–274.

Spellman,P.T. *et al*. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, **9**, 3273–3297.

Wuensche,A. (2009) Discrete dynamics lab: tools for investigating cellular automata and discrete dynamical networks. In Adamatzky,A. and Komosinski,M. (eds) *Artificial Life Models in Software*. Springer, London, pp. 215–258.