Oles, V and Kukushkin, A 2020 BoolSi: A Tool for Distributed Simulations and Analysis of Boolean Networks. *Journal of Open Research Software,* 8: 26. DOI: https://doi.org/10.5334/jors.308

SOFTWARE METAPAPER

BoolSi: A Tool for Distributed Simulations and Analysis of Boolean Networks

Vladyslav Oles* and Anton Kukushkin†

* Department of Mathematics and Statistics, Washington State University, US

† Unaffiliated

Corresponding author: Vladyslav Oles (vlad.oles@protonmail.com)

We present BoolSi, an open-source cross-platform command line tool for distributed simulations of synchronous Boolean networks. It uses MPI standard to support execution on computational clusters, as well as parallel processing on a single computer. BoolSi can be used to model the behavior of complex dynamic networks, such as gene regulatory networks. In particular, it allows for identification and statistical analysis of network attractors. We perform a case study of the activity of a cambium cell to demonstrate the capabilities of the tool.

Keywords: Boolean networks; MPI; Python; bioinformatics; gene regulatory networks

(1) Overview

Introduction

A Boolean network is a discrete dynamical system consisting of nodes that represent objects from application domain, and their update rules, that represent interactions between the objects. Update rule of a node is a Boolean function that defines the next state of the node based on current states of nodes. Therefore, each node can take one of two possible states, denoted 0 or 1. In synchronous Boolean networks, the next state of the network is calculated by applying all update rules simultaneously. In asynchronous Boolean networks, the states of only a subset of network nodes get updated at a time. The exact nodes to update at each time step can be either determined by some pre-selected rule, or chosen at random, which introduces non-determinism into system dynamics.

State space of a Boolean network is finite and consists of 2^n network states, where n is the number of nodes in the network. Simulating a Boolean network with deterministic dynamics will eventually cause the network to reach a previously visited state and thus fall into a cycle called an attractor. In the case of a cycle of length 1, the attractor is called a steady state. States of the network that lead to a particular attractor constitute its basin of attraction. Attractors represent long-term behavior of the modeled processes and may therefore carry important information about them [10, 11].

Using Boolean networks to model continuous real-world processes requires binary interpretation of their states, which is usually achieved through thresholding. Despite such simplification, Boolean network models can capture important properties of the dynamics of complex systems, which makes them a practical choice in a range of domains, including systems biology [6], robotics [16], epidemiology [13], and economics [3, 15].

Due to the practicality of the model, a number of software tools for Boolean network simulations have been made available. BoolNet [12] is an R package for construction, simulation and analysis of synchronous, asynchronous, and random Boolean networks, that supports identification of synchronous attractors by both exhaustive and heuristic search. BooleanNet [1] is a Python software library for simulation and visualization of synchronous and asynchronous Boolean networks, capable of detecting their attractors. Unlike most other tools from the field, BooleanNet allows for the simulation of simulate discrete update rules as piecewise linear differential equations by introducing continuous variables to the model. RaBooNet [7] is a Matlab toolbox for generation and simulation of random Boolean networks, with an emphasis on their application to supervised machine learning. RaBooNet does not support user-defined update rules of a network, only allowing rules to be randomly generated based on network structure. BioNetGen [4], GINsim [8], and PlantSimLab [9] are software packages with graphical user interface, aimed at rule-based modeling of biochemical systems. All three packages can operate with both binary-state and multi-state models.

Here, we present BoolSi, a command line tool for simulation and analysis of synchronous Boolean networks. The main distinguishing characteristic of BoolSi is the support for MPI standard to run simulations in parallel, e.g. on a computational cluster. To the best of our knowledge, this is the first Boolean network modeling tool capable of high-performance computing. Another unique feature of BoolSi is the analysis that it performs on the identified attractors. This approach, inspired by statistical mechanics, was first introduced in [14] and used therein to better understand the role of hormonal signals in regulating cambium proliferation.

Implementation and architecture

BoolSi requires user to define nodes, update rules, and initial states of the network via simple text input. It supports simulating the network from a range of initial states within a single run, and aggregating the results of simulations. In particular, BoolSi can use exhaustive search to identify all network attractors, as well as their trajectory length distributions and the sizes of their basins of attraction. User can limit the length of attractors to search for — for example, setting the limit to 1 will identify the steady states only. Another mode of operation aims to find conditions that lead to specific states of the network.

It is possible to override the simulation process, which can be used to model external influence on the network. User can fix the state of any node for the entire simulation (e.g. modeling gene knockout in regulatory networks), or perturb it at any time step (e.g. modeling sensory input in robotics).

To speed up simulations, BoolSi stores pre-evaluated update rules as truth tables. This requires $O(n \cdot 2^d)$ memory, where n is the number of nodes in the network and d is its maximum in-degree. Such an approach imposes a practical limitation on the in-degree of a node, with the majority of modern computers being able to handle the nodes of in-degree up to 20. This however is sufficient for many Boolean network models from the literature, where the in-degrees rarely exceed 10 (see e.g. [2, 5] and the references therein).

Attractor analysis

BoolSi performs statistical analysis of the found attractors to extract information about the interplay between the nodes. Let n denote the number of nodes in a Boolean network. Simulating the network from each of its possible initial states $s^k \in \{0, 1\}^n$, $k = 1, \ldots, 2^n$, leads to an attractor, comprised of one or multiple network states. We define $\alpha_k^i \in [0, 1]$, the activity coefficient of node i associated with network state s^k, as the average of 0 and 1 states of the node in the attractor reachable from s^k. **Figure 1** shows an example of the activity coefficient calculation. All activity coefficients of node i form vector $\alpha^i \in [0, 1]^{2^n}$, which represents long-term activity of the node with regard to the entire state space of the network. For every pair of nodes i and j, BoolSi computes Spearman's correlation coefficient $\rho_{ij} \in [-1, 1]$ between α^i and α^j, interpreted as the degree of mutual amplification of the two nodes. We note that the impact of each attractor on ρ_{ij} is proportional to the size of its basin of attraction. Spearman's correlation measures the strength and direction of a monotonic association between two variables, and was chosen for its ability to capture non-linear relationships.

User interface

BoolSi provides a command line interface. It relies on YAML input file to define nodes and update rules of a Boolean network and to provide simulation details, such as initial states. Update rules are described using common syntax for logical functions, including operators "and", "or", "not".

BoolSi supports a number of output formats: PDF documents, vector or raster image files (SVG, PNG, TIFF), or CSV text files to allow for machine processing.

Quality control

BoolSi code is covered by unit and functional tests. To demonstrate performance and functionality of our software, we perform a case study on a Boolean network model of cambium cell. The results can be reproduced by executing `$ mpiexec -np 168 boolsi attract cambium.yaml` (the contents of `cambium.yaml` are shown in Appendix A), assuming that BoolSi, as well as an MPI implementation and `mpi4py` library, are installed.

Case study

Cambium is a plant tissue layer that produces cells for plant growth. It is concealed under multiple layers of other cells, which hinders identification of mutants with

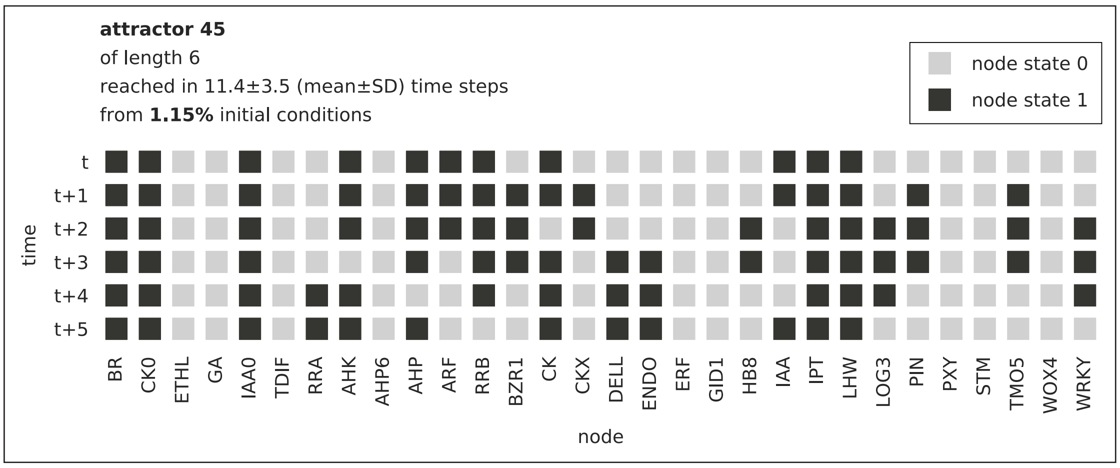


Figure 1: BoolSi output showing an attractor of a 30-node network (model of cambium regulation from [14], used in Section Case study). The activity coefficient of the node CK associated with each of roughly $0.0115 \cdot 2^{30} \approx 12{,}348{,}031$ network states s^k leading to this attractor is $\alpha_k^{\text{CK}} = \frac{5}{6}$.

altered secondary growth and isolation of live cambium cells. As a consequence, understanding the biology of cambium remains incomplete. Mathematical modeling has the potential to help with predicting the outcome of interactions controlling cambium activity [14].

We based the case study off of the 30-node BN model of cambium regulation (see **Figure 2**) described in [14], formatted as a YAML input file for BoolSi (see Appendix A).

We used BoolSi to find network attractors by simulating from all $2^{30} \approx 1.07 \cdot 10^9$ initial states of the network, and to analyze the found attractors. The exhaustive search took about 45 hours on a cluster of 21 computers with 8 CPU cores each. Of the total 168 CPU cores, 112 were 4 GHz, 17 were 3.4 GHz, 11 were 2.1 GHz, and 28 were 1.4 GHz.

The results of attractor analysis (**Figure 3**) agree with the findings of [14] and experimental data. In particular, each

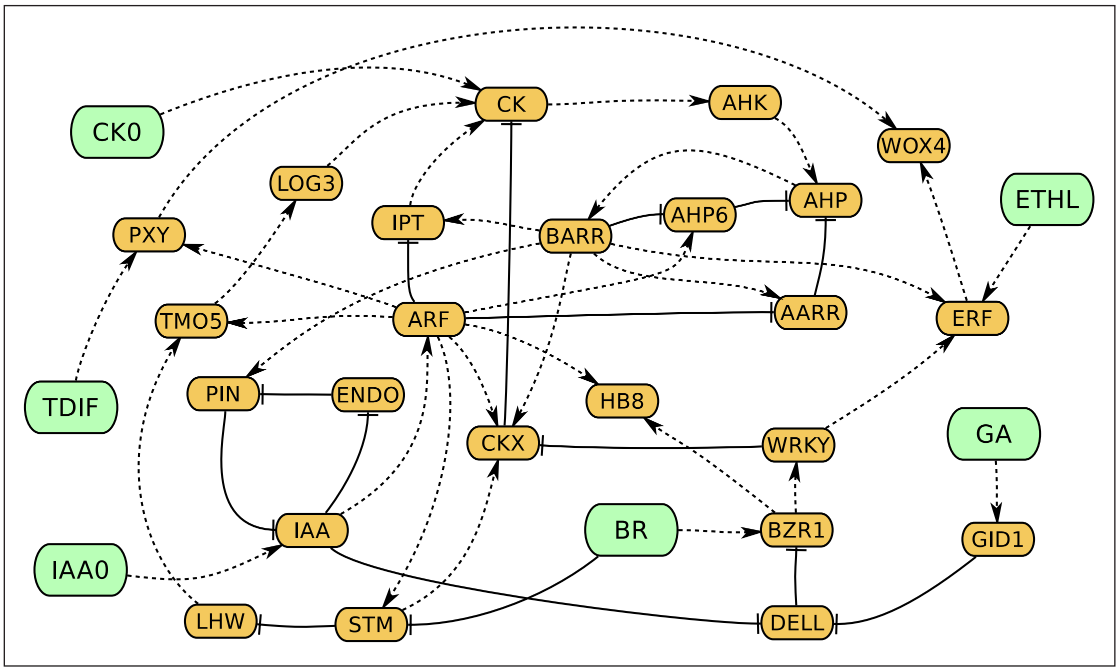
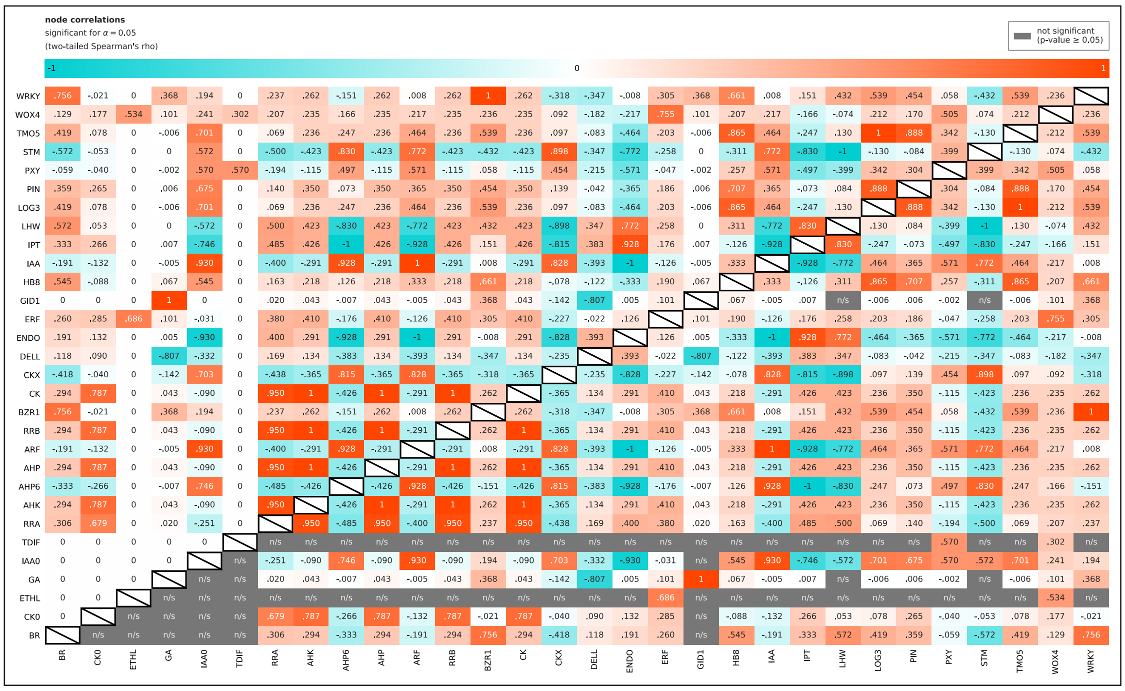


Figure 2: Cambium regulation network from [14]. Bigger, green-colored nodes represent hormonal signals produced outside cambium; their states do not change throughout each simulation run. Dashed and solid lines represent activation and inhibition respectively.



node correlations
significant for $\alpha = 0.05$
(two-tailed Spearman's rho)

▬ not significant (p-value ≥ 0.05)

-1 … 0 … 1

	BR	CK0	ETHL	GA	IAA0	TDIF	RRA	AHK	AHP6	AHP	ARF	RRB	BZR1	CK	CKX	DELL	ENDO	ERF	GID1	HB8	IAA	IPT	LHW	LOG3	PIN	PXY	STM	TMO5	WOX4	WRKY
WRKY	.756	-.021	0	.368	.194	0	.237	.262	-.151	.262	.008	.262	1	.262	-.318	-.347	-.008	.305	.368	.661	.008	.151	.432	.539	.454	.058	-.432	.539	.236	
WOX4	.129	.177	.534	.101	.241	.302	.207	.235	.166	.235	.217	.235	.236	.235	.092	-.182	-.217	.755	.101	.207	.217	-.166	-.074	.212	.170	.505	.074	.212		.236
TMO5	.419	.078	0	-.006	.701	0	.069	.236	.247	.236	.464	.236	.539	.236	.097	-.083	-.464	.203	-.006	.865	.464	-.247	.130	1	.888	.342	-.130		.212	.539
STM	-.572	-.053	0	0	.572	0	-.500	-.423	.830	-.423	.772	-.423	-.432	-.423	.898	-.347	-.772	-.258	0	-.311	.772	-.830	-1	-.130	-.084	.399		-.130	.074	-.432
PXY	-.059	-.040	0	-.002	.570	.570	-.194	-.115	.497	-.115	.571	-.115	.058	-.115	.454	-.215	-.571	-.047	-.002	.257	.571	-.497	-.399	.342	.304		.399	.342	.505	.058
PIN	.359	.265	0	.006	.675	0	.140	.350	.073	.350	.365	.350	.454	.350	.139	-.042	-.365	.186	.006	.707	.365	-.073	.084	.888		.304	-.084	.888	.170	.454
LOG3	.419	.078	0	-.006	.701	0	.069	.236	.247	.236	.464	.236	.539	.236	.097	-.083	-.464	.203	-.006	.865	.464	-.247	.130		.888	.342	-.130	1	.212	.539
LHW	.572	.053	0	0	-.572	0	.500	.423	-.830	.423	-.772	.423	.432	.423	-.898	.347	.772	.258	0	.311	-.772	.830		.130	.084	-.399	-1	.130	-.074	.432
IPT	.333	.266	0	.007	-.746	0	.485	.426	-1	.426	-.928	.426	.151	.426	-.815	.383	.928	.176	.007	-.126	-.928		.830	-.247	-.073	-.497	-.830	-.247	-.166	.151
IAA	-.191	-.132	0	-.005	.930	0	-.400	-.291	.928	-.291	1	-.291	.008	-.291	.828	-.393	-1	-.126	-.005	.333		-.928	-.772	.464	.365	.571	.772	.464	.217	.008
HB8	.545	-.088	0	.067	.545	0	.163	.218	.126	.218	.333	.218	.661	.218	-.078	-.122	-.333	.190	.067		.333	-.126	.311	.865	.707	.257	-.311	.865	.207	.661
GID1	0	0	0	1	0	0	.020	.043	-.007	.043	-.005	.043	.368	.043	-.142	-.807	.005	.101		.067	-.005	.007	n/s	-.006	.006	-.002	n/s	-.006	.101	.368
ERF	.260	.285	.686	.101	-.031	0	.380	.410	-.176	.410	-.126	.410	.305	.410	-.227	-.022	.126		.101	.190	-.126	.176	.258	.203	.186	-.047	-.258	.203	.755	.305
ENDO	.191	.132	0	.005	-.930	0	.400	.291	-.928	.291	-1	.291	-.008	.291	-.828	.393		.126	.005	-.333	-1	.928	.772	-.464	-.365	-.571	-.772	-.464	-.217	-.008
DELL	.118	.090	0	-.807	-.332	0	.169	.134	-.383	.134	-.393	.134	-.347	.134	-.235		.393	-.022	-.807	-.122	-.393	.383	.347	-.083	-.042	-.215	-.347	-.083	-.182	-.347
CKX	-.418	-.040	0	-.142	.703	0	-.438	-.365	.815	-.365	.828	-.365	-.318	-.365		-.235	-.828	-.227	-.142	-.078	.828	-.815	-.898	.097	.139	.454	.898	.097	.092	-.318
CK	.294	.787	0	.043	-.090	0	.950	1	-.426	1	-.291	1	.262		-.365	.134	.291	.410	.043	.218	-.291	.426	.423	.236	.350	-.115	-.423	.236	.235	.262
BZR1	.756	-.021	0	.368	.194	0	.237	.262	-.151	.262	.008	.262		.262	-.318	-.347	-.008	.305	.368	.661	.008	.151	.432	.539	.454	.058	-.432	.539	.236	1
RRB	.294	.787	0	.043	-.090	0	.950	1	-.426	1	-.291		.262	1	-.365	.134	.291	.410	.043	.218	-.291	.426	.423	.236	.350	-.115	-.423	.236	.235	.262
ARF	-.191	-.132	0	-.005	.930	0	-.400	-.291	.928	-.291		-.291	.008	-.291	.828	-.393	-1	-.126	-.005	.333	1	-.928	-.772	.464	.365	.571	.772	.464	.217	.008
AHP	.294	.787	0	.043	-.090	0	.950	1	-.426		-.291	1	.262	1	-.365	.134	.291	.410	.043	.218	-.291	.426	.423	.236	.350	-.115	-.423	.236	.235	.262
AHP6	-.333	-.266	0	-.007	.746	0	-.485	-.426		-.426	.928	-.426	-.151	-.426	.815	-.383	-.928	-.176	-.007	.126	.928	-1	-.830	.247	.073	.497	.830	.247	.166	-.151
AHK	.294	.787	0	.043	-.090	0	.950		-.426	1	-.291	1	.262	1	-.365	.134	.291	.410	.043	.218	-.291	.426	.423	.236	.350	-.115	-.423	.236	.235	.262
RRA	.306	.679	0	.020	-.251	0		.950	-.485	.950	-.400	.950	.237	.950	-.438	.169	.400	.380	.020	.163	-.400	.485	.500	.069	.140	-.194	-.500	.069	.207	.237
TDIF	0	0	0	0	0		n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	.570	n/s	n/s	.302	n/s
IAA0	0	0	0	0		n/s	-.251	-.090	.746	-.090	.930	-.090	.194	-.090	.703	-.332	-.930	-.031	n/s	.545	.930	-.746	-.572	.701	.675	.570	.572	.701	.241	.194
GA	0	0	0		n/s	n/s	.020	.043	-.007	.043	-.005	.043	.368	.043	-.142	-.807	.005	.101	1	.067	-.005	.007	n/s	-.006	.006	-.002	n/s	-.006	.101	.368
ETHL	0	0		n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	.686	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	n/s	.534	n/s
CK0	0		n/s	n/s	n/s	n/s	.679	.787	-.266	.787	-.132	.787	-.021	.787	-.040	.090	.132	.285	n/s	-.088	-.132	.266	.053	.078	.265	-.040	-.053	.078	.177	-.021
BR		n/s	n/s	n/s	n/s	n/s	.306	.294	-.333	.294	-.191	.294	.756	.294	-.418	.118	.191	.260	n/s	.545	-.191	.333	.572	.419	.359	-.059	-.572	.419	.129	.756

Figure 3: BoolSi output showing the results of attractor analysis. Cell in *i*-th row and *j*-th column shows ρ_{ij}, Spearman's correlation coefficient between activity of the nodes *i* and *j* in the attractors. In the lower-right triangle, the values corresponding to p-values ≥ 0.05 are hidden.

of the nodes IAA and BR showed positive correlation with both WOX4 and HB8, whose combined activity represents cambium proliferation. Node ETHL exhibited strong positive correlation (>0.5) with WOX4, but no significant correlation with HB8. The analysis also reproduced the mutually inhibitory relationship between cytokinin and auxin signaling [17], manifested as the negative correlation between the nodes IAA and CK. Unlike in [14], our analysis found a positive effect of the nodes CK and TDIF on the expression of WOX4, as well as a positive relationship between CK and HB8 (the correlation between TDIF and HB8 was 0). This is consistent with published experimental data on the key role of both CK and TDIF in regulation of cambium activity. We note that BoolSi automatically extracts information about the interplay between each pair of nodes, stepping up the approach of manually inferring the relationships between the selected nodes.

To test if BoolSi can accurately model the mutations of a cambium cell, we simulated the effect of mutations known to reduce cambium proliferation: double knockout of ERFs *erf108erf109*, a quadruple knockout of IPT which was deficient in cytokinin synthesis, and mutants in cytokinin receptor AHK4 [14]. Mutations were mimicked by fixing the states of ERF, IPT, or AHK to be 0 in the simulations (see Appendix B for the corresponding BoolSi input). In agreement with the experimental data, all mutations resulted in lower proliferation activity, measured as $\sqrt{\frac{||\alpha^{\text{WOX4}}||_1^2+||\alpha^{\text{HB8}}||_1^2}{2}}$ (see Section Attractor analysis). The proliferation activity in *erf*, *ipt*, and *ahk* was, respectively, 1.91, 1.16, and 1.02 times lower than in the model of cambium unaffected by mutations. We note that the insignificance of the decrease of proliferation activity in *ahk* suggests the direction for improvement of the model.

(2) Availability

BoolSi can be installed via Python package manager pip (e.g. `pip install boolsi`), or from source.

Operating system

*nix (tested on Ubuntu 18.04 LTS and macOS 10.14), Windows (tested on Windows 8).

Programming language

Python 3.4 or higher.

Dependencies

BoolSi requires the following Python libraries: `numpy>=1.12.0`, `scipy>=0.14.0`, `Click`, `PyYAML`, `matplotlib`, `seaborn`, `pillow`, `ZODB`, `BTrees`, `persistent`, `transaction`. Note that the combination of `seaborn=0.9.0` and `matplotlib=3.1.1` is known to have issues with displaying the plots correctly.

Running BoolSi with MPI also requires installing an MPI implementation and `mpi4py` library. BoolSi has been tested with OpenMPI and MPICH, and we expect it to work with other MPI implementations as well.

List of contributors

Vladyslav Oles, Anton Kukushkin.

Software location

Archive

Name: Zenodo
Persistent identifier: 10.5281/zenodo.3766974
Licence: MIT
Publisher: Vladyslav Oles
Version published: v1.0.0
Date published: 26 Apr 2020

Code repository

Name: GitHub
Persistent identifier: https://github.com/boolsi/boolsi
Licence: MIT
Date published: 7 Oct 2019

Language

English.

(3) Reuse potential

BoolSi is an open-source tool for simulations and analysis of Boolean networks, written in Python 3. It is the first BN modeling tool capable of high-performance computing. In addition, BoolSi incorporates a novel method for analyzing network attractors, capable of capturing emergent properties of a Boolean network. The software can be used for studying network dynamics, and in particular biological networks.

The latest version of BoolSi and user documentation is hosted on GitHub repository at http://github.com/boolsi/boolsi. The documentation contains simple examples that can be easily modified for different user scenarios. BoolSi is released under the MIT license and welcomes any contributions. We encourage users to submit feedback using GitHub issue tracker, or by emailing any of the authors.

Appendices

A. Input file for cambium cell model

```
nodes:
  - BR
  - CK0
  - ETHL
  - GA
  - IAA0
  - TDIF
  - RRA
  - AHK
  - AHP6
  - AHP
  - ARF
  - RRB
  - BZR1
  - CK
  - CKX
  - DELL
  - ENDO
  - ERF
  - GID1
  - HB8
  - IAA
  - IPT
```

```
 - LHW
 - LOG3
 - PIN
 - PXY
 - STM
 - TMO5
 - WOX4
 - WRKY

update rules:
 BR: BR
 CK0: CK0
 ETHL: ETHL
 GA: GA
 IAA0: IAA0
 TDIF: TDIF
 RRA: RRB and not ARF
 AHK: CK
 AHP6: ARF and not RRB
 AHP: AHK and not (RRA and AHP6)
 ARF: IAA
 RRB: AHP
 BZR1: BR and not DELL
 CK: majority(CK0, IPT, LOG3, not CKX, 0)
 CKX: majority(ARF, RRB, not WRKY, STM, 0)
 DELL: not (IAA or GID1)
 ENDO: not IAA
 ERF: ETHL and (RRB or WRKY)
 GID1: GA
 HB8: ARF and BZR1
 IAA: IAA0 and not PIN
 IPT: RRB or not ARF
 LHW: not STM
 LOG3: TMO5
 PIN: RRB and not ENDO
 PXY: TDIF and ARF
 STM: ARF and not BR
 TMO5: LHW and ARF
 WOX4: PXY or ERF
 WRKY: BZR1

initial state:
 BR: any
 CK0: any
 ETHL: any
 GA: any
 IAA0: any
 TDIF: any
 RRA: any
 AHK: any
 AHP6: any
 AHP: any
 ARF: any
 RRB: any
 BZR1: any
 CK: any
 CKX: any
 DELL: any
 ENDO: any
 ERF: any
 GID1: any
 HB8: any
 IAA: any
 IPT: any
 LHW: any
 LOG3: any
 PIN: any
 PXY: any
 STM: any
 TMO5: any
 WOX4: any
 WRKY: any
```

B. Input file modifications for modeling the effects of mutations

1. Modifications of the input file (see Appendix A) for modeling ERF mutant background:

```
83c83
<    ERF: any
---
ERF: 0
97c97,98
<
---
> fixed nodes:
>    ERF: 0
```

2. Modifications of the input file for modeling IPT mutant background:

```
83c83
<    IPT: any
---
IPT: 0
97c97,98
<
---
> fixed nodes:
>    IPT: 0
```

3. Modifications of the input file for modeling ERF mutant background:

```
83c83
<    AHK: any
---
AHK: 0
97c97,98
<
---
> fixed nodes:
>    AHK: 0
```

Acknowledgements
The authors are grateful to the Department of Mathematics and Statistics of Washington State University for providing access to the computer cluster used in the case study. In addition, the authors would like to thank Kevin Cooper for his guidance on MPI standard and Oracle Grid Engine software, Alexander Panchenko for suggestions on the functionality of BoolSi, Taras Nazarov for their assistance with the biology part of the paper, Iana Khotsianivska and

Aleksey Lapenko for helping with the design of BoolSi output, and Grace Grimm for reviewing and editing.

Competing Interests

The authors have no competing interests to declare.

References

1. **Albert, I, Thakar, J, Li, S, Zhang, R** and **Albert, R** 2008 Boolean network simulations for life scientists. *Source code for biology and medicine*, 3(1): 16. DOI: https://doi.org/10.1186/1751-0473-3-16
2. **Aldana, M, Coppersmith, S** and **Kadanoff, L P** 2003 Boolean dynamics with random couplings. In: *Perspectives and Problems in Nolinear Science*, 23–89. New York, NY: Springer. DOI: https://doi.org/10.1007/978-0-387-21789-5_2
3. **Alexander, J M** 2003 Random Boolean networks and evolutionary game theory. *Philosophy of Science*, 70(5): 1289–1304. DOI: https://doi.org/10.1086/377408
4. **Blinov, M L, Faeder, J R, Goldstein, B** and **Hlavacek, W S** 2004 BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17): 3289–3291. DOI: https://doi.org/10.1093/bioinformatics/bth378
5. **Bornholdt, S** 2008 Boolean network models of cellular regulation: prospects and limitations. *Journal of the Royal Society Interface*, 5(suppl_1): S85–S94. DOI: https://doi.org/10.1098/rsif.2008.0132.focus
6. **Davidich, M I** and **Bornholdt, S** 2008 Boolean network model predicts cell cycle sequence of fission yeast. *PloS one*, 3(2). DOI: https://doi.org/10.1371/journal.pone.0001672
7. **Ferranti, D, Krane, D** and **Craft, D** 2017 The value of prior knowledge in machine learning of complex network systems. *Bioinformatics*, 33(22): 3610–3618. DOI: https://doi.org/10.1093/bioinformatics/btx438
8. **Gonzalez, A G, Naldi, A, Sanchez, L, Thieffry, D** and **Chaouiya, C** 2006 GINsim: A software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems*, 84(2): 91–100. DOI: https://doi.org/10.1016/j.biosystems.2005.10.003
9. **Ha, S, Dimitrova, E, Hoops, S, Altarawy, D, Ansariola, M, Deb, D, Glazebrook, J, Hillmer, R, Shahin, H, Katagiri, F** and **McDowell, J** 2019 PlantSimLab-a modeling and simulation web tool for plant biologists. *BMC bioinformatics*, 20(1): 1–11. DOI: https://doi.org/10.1186/s12859-019-3094-9
10. **Kauffman, S A** 1993 *The origins of order: Self-organization and selection in evolution*. USA: Oxford University Press. DOI: https://doi.org/10.1007/978-94-015-8054-0_8
11. **Li, F, Long, T, Lu, Y, Ouyang, Q** and **Tang, C** 2004 The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences*, 101(14): 4781–4786. DOI: https://doi.org/10.1073/pnas.0305937101
12. **Müssel, C, Hopfensitz, M** and **Kestler, H A** 2010 BoolNet — an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, 26(10): 1378–1380. DOI: https://doi.org/10.1093/bioinformatics/btq124
13. **Newman, M E** 2002 Spread of epidemic disease on networks. *Physical review E*, 66(1): 016128. DOI: https://doi.org/10.1103/PhysRevE.66.016128
14. **Oles, V, Panchenko, A** and **Smertenko, A** 2017 Modeling hormonal control of cambium proliferation. *PloS one*, 12(2). DOI: https://doi.org/10.1371/journal.pone.0171927
15. **Paczuski, M, Bassler, K E** and **Corral, Á** 2000 Self-organized networks of competing boolean agents. *Physical Review Letters*, 84(14): 3185. DOI: https://doi.org/10.1103/PhysRevLett.84.3185
16. **Roli, A, Manfroni, M, Pinciroli, C** and **Birattari, M** 2011, April. On the design of Boolean network robots. In: *European Conference on the Applications of Evolutionary Computation*, 43–52. Berlin, Heidelberg: Springer. DOI: https://doi.org/10.1007/978-3-642-20525-5_5
17. **Schaller, G E, Bishopp, A** and **Kieber, J J** 2015 The yin-yang of hormones: cytokinin and auxin interactions in plant development. *The Plant Cell*, 27(1): 44–63. DOI: https://doi.org/10.1105/tpc.114.133595

How to cite this article: Oles, V and Kukushkin, A 2020 BoolSi: A Tool for Distributed Simulations and Analysis of Boolean Networks. *Journal of Open Research Software,* 8: 26. DOI: https://doi.org/10.5334/jors.308

Submitted: 27 October 2019 **Accepted:** 17 September 2020 **Published:** 07 October 2020