

# ASSA-PBN 3.0: Analysing Context-sensitive Probabilistic Boolean Networks

Andrzej Mizera<sup>3,4</sup>, Jun Pang<sup>1,2</sup>, Hongyang Qu<sup>5</sup>, and Qixia Yuan<sup>1</sup>

<sup>1</sup> Faculty of Science, Technology and Communication, University of Luxembourg

<sup>2</sup> Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg

<sup>3</sup> Luxembourg Centre for Systems Biomedicine, University of Luxembourg

<sup>4</sup> Department of Infection and Immunity, Luxembourg Institute of Health

<sup>5</sup> Department of Automatic Control & Systems Engineering, University of Sheffield

**Abstract.** We present a major new release of ASSA-PBN, a software tool for modelling, simulation, and analysis of probabilistic Boolean networks (PBNs). The new version enables the support for context-sensitive PBNs (CPBNs), which can well balance the uncertainty and stability of the modelled biological systems. It contributes mainly in three aspects. Firstly, it designs a high-level language for specifying CPBNs. Secondly, it implements various simulation-based methods for simulating CPBNs and analysing their long-run dynamics. Last but not least, it provides an efficient method to identify all the attractors of a CPBN. Thanks to its divide and conquer strategy, the implemented detection algorithm can deal with large and realistic biological networks under both synchronous and asynchronous updating schemes.

## 1 Introduction

Probabilistic Boolean networks (PBNs) [1, 2] is a modelling framework widely used to model gene regulatory networks (GRNs). It was introduced as an extension of Boolean networks (BNs) to cope with the uncertainties related to the system under study. Instead of providing a fixed Boolean function for each node, a PBN associates a set of functions for each node and, at any time point, one of the functions in the set is selected in accordance with a probability distribution to determine the value of the corresponding node. This is the original definition of a PBN, which is known as an *instantaneously random* PBN (IPBN). This definition, however, does not consider the perspective that information might come from distinct sources, each representing a “context” of a cell. To capture this perspective, another type of PBNs, called *context-sensitive* PBNs (CPBN), is introduced [3]. A CPBN represents each of the contexts as a BN, and switches between contexts in a stochastic manner. At each step, a CPBN can either remain in a current context, or switch to a new context with a pre-defined probability and evolve accordingly. When a CPBN is within a certain context, it will eventually settle into a set of state cycles called *attractors*. The CPBN can only leave an attractor when the context is switched or perturbations are incorporated. The attractors characterise the long-run behaviours of a CPBN and are very important for understanding the underlying networks. Comparing to IPBNs, CPBNs can better reflect the stability of biological systems. The

study of attractors of a biological system are therefore more performed in terms of CPBNs instead of IPBNs. Example studies of CPBNs can be found in [4–6].

Being able to capture both the attractor information and the uncertainties of the modelled networks, CPBNs is an important and useful framework for analysing and understanding the long-run behaviours of biological networks. A few tools [7–11] have been developed for analysing PBNs or BNs. However, none of them provides intrinsic support for CPBNs. Moreover, the attractor detection functionality provided for BNs is largely restricted by the network size in existing tools. A user-friendly tool supporting attractor detection of large CPBNs is still required.

**Motivations of ASSA-PBN 3.0.** The previous versions of ASSA-PBN [12, 13] have provided rich functionalities for modelling, simulation and analysis of IPBNs. To support efficient analysis of CPBNs, we release this new version of ASSA-PBN. The most important novelty of this version is that it supports the modelling and analysis of CPBNs and includes an important function of *attractor detection*. This functionality is implemented using efficient binary decision diagram (BDD) operations and can identify all the attractors of a given BN. Compared to similar functionalities in other tools, a significant novelty of ASSA-PBN in this respect is that it uses a strongly connected component (SCC)-based decomposition method for dealing with large CPBNs; hence it is able to handle large and realistic PBNs efficiently.

In addition, this new version contributes in another two major aspects. Firstly, it designs a high-level language to specify a CPBN in a straightforward way. Secondly, it extends the existing simulation-based functionalities for analysing the long-run dynamics of CPBNs. This includes computation of steady-state probabilities of CPBNs and parameter estimation for CPBNs. Due to the differences in the semantics of IPBNs and CPBNs, the extension not only adapts the codes with the new semantics, but also applies a few new methods to optimise the performance of the analysis. We give below an overview of the major functionalities of ASSA-PBN and highlight with bold font the new functionalities provided in the newly released version:

- a high-level modelling language for specifying both IPBNs and **CPBNs**;
- random generation of PBNs (both instantaneously random and **context-sensitive**);
- efficient simulation of PBNs (both instantaneously random and **context-sensitive**);
- computation of steady-state probabilities (for both instantaneously random and **context-sensitive** PBNs);
- parameter estimation for both instantaneously random and **context-sensitive** PBNs;
- **attractor detection for constituent BNs (both under the synchronous and the asynchronous update scheme).**

## 2 Preliminaries

We briefly introduce the concept of BNs and PBNs in this section. A BN models the elements of a system with binary-valued nodes. Each node is assigned with a Boolean function to reflect the relationships among the nodes. In the literature, there are various updating schemes for BNs. We introduce here two most frequently used ones, i.e. the

*synchronous* and the *asynchronous* schemes.<sup>6</sup> In a synchronous BN, the values of all the nodes are updated according to their Boolean functions at each time step. In an asynchronous BN, the value of a single, randomly selected node is updated according to its Boolean function at each time step. The dynamics of a BN can be viewed as a discrete-time Markov chain (DTMC). Given an initial state, a BN will eventually evolve into a set of states called an attractor, inside which the BN will stay forever.

A PBN extends a BN by assigning more than one Boolean function to each node. Formally, a PBN  $G(V, \mathcal{F})$  consists of a set of binary-valued nodes (also referred to as genes)  $V = \{v_1, v_2, \dots, v_n\}$  and a list of vector-valued function sets  $\mathcal{F} = (F_0, F_1, \dots, F_k)$ . Each vector-valued function  $F_\ell = (F_\ell^{(1)}, F_\ell^{(2)}, \dots, F_\ell^{(n)})$  determines a constituent Boolean network, or *context*, of the PBN. The function  $F_\ell^{(i)}$  is a predictor of node  $i$  whenever the context  $\ell$  is selected. At each time step, a decision is made whether to switch context based on the switching probability  $q$ . If  $q < 1$ , the PBN is said to be context-sensitive; if  $q = 1$ , the PBN is said to be instantaneously random. If the decision is to switch context, then a context is selected according to the selection probability distribution  $D = (D_0, D_1, \dots, D_k)$ , where  $D_i (i \in [0, k])$  is the probability that context  $i$  is selected. The newly selected context can be the same as the current context. As long as a CPBN remains in one context, it will eventually evolve into an attractor of this context and remain within it. When the context is switched, the CPBN can move out of this attractor. Given a small switch probability, the CPBN can remain in one attractor for a long time. Therefore, it is meaningful to identify the attractors of each context. The *attractors of a CPBN* are defined to be all the attractors of its constituent BNs. We refer to [3] and [1, page 4] for a formal definition and more detailed description of PBNs.

### 3 Usage and New Features

Similar to its previous versions, the newly released ASSA-PBN consists of three main modules, i.e. a modeller, a simulator, and an analyser, which interact with each other to provide a full support for PBN analysis. The modeller provides support to use a high-level language for constructing a PBN model of a real-life biological system, e.g. a GRN, and supports visualisation and interactive editing of PBN predictor function values. The simulator takes the PBN constructed in the modeller and performs simulation to produce trajectory samples. Based on the constructed model and generated samples, the analyser performs basic and in-depth analysis of the PBN. The analysis results can be used either to obtain insights into the original system or to optimise the fitting of the system to experimental measurements. We proceed with describing the three main newly added features.

1	<b>contextDistribution</b> = 0.1, 0.2, 0.7
2	<b>switch</b> = 0.001
3	<b>node</b> $x_0$
4	0,1: $(!x_2) \& (x_1   x_0)$
5	2: $(!x_0) \& x_1 \& x_2$
6	<b>endNode</b>

Fig. 1: Specifying Boolean functions for node  $x_0$ .

<sup>6</sup> For other schemes, we refer to [14].

**A high-level CPBN specification language.** We design a simple high-level language for specifying CPBNs. There are three main parts of a definition of a CPBN in this language: the probability distribution on contexts, the probability for making a switch at each time step, and the Boolean functions in each context. The first line in Figure 1 shows an example for specifying the probability distribution for three contexts, and the second line shows the specification of the switch probability. We describe the Boolean functions in all contexts node by node for a node at one time and specify which context a Boolean function belongs to. The contexts of a CPBN are referred to as numbers starting from 0. Lines 3-6 of Figure 1 show an example for specifying the Boolean functions for node  $x_0$ . In this example, node  $x_0$  has two Boolean functions. The first function, specified in the fourth line, is shared by contexts 0 and 1; and the second function, specified in the fifth line, is only assigned to context 2. To the best of our knowledge, this language for describing context-sensitive PBNs is the first high-level language to be supported by a software tool.

**An efficient simulator and simulation-based analyses for CPBNs.** Statistical approaches are practically the only viable option for the analysis of huge PBNs [15]. Applications of such methods necessitate generation of trajectories of significant length. In order to make the analysis to execute in a reasonable amount of time, ASSA-PBN has applied a number of techniques like multiple CPU/GPU core parallel generation [16, 17]. In this newly released version, the simulation of a CPBN is implemented. To make the simulation as efficient as possible, we combine  $s$ , the switch probability, with  $D = (D_0, D_1, \dots, D_k)$ , the probability distribution over all the contexts, to form a united distribution  $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_k, \mathcal{D}_{k+1})$ , where  $\mathcal{D}_i = D_i * s$  for  $i \in [0, k]$  and  $\mathcal{D}_{k+1} = 1 - s$ . The value of  $\mathcal{D}_i$  ( $i \in [0, k]$ ) represents the probability that a switch of context is made and the new context is the  $i$ th one; and the value of  $\mathcal{D}_{k+1}$  represents the probability that no switch is made. The switching of a context can then be performed by sampling only once from this new distribution  $\mathcal{D}$ , instead of sampling twice when performed based on the original distribution  $D$  and  $s$ . The simulation of a CPBN provides the foundation for further analysis of it with simulation-based methods, e.g., steady-state probability computation, long-run influence analysis, long-run sensitivity analysis, and parameter estimation for CPBNs.

**Decomposition-based methods for attractor detection.** Attractor detection is a completely new functionality in this version. It is specially designed for large CPBNs and BNs. Attractor detection of a CPBN or a BN is non-trivial since attractors are determined based on the network's states, the number of which is exponential in the number of nodes. A lot of efforts have been put in the development of attractor detection algorithms and tools, e.g. [18, 7, 9, 10, 19, 20]. However, these tools are prohibited in large BNs due to the well-known state-space explosion problem, especially for asynchronous networks where there are potentially multiple outgoing transitions in each state.

In order to deal with large BNs, we apply the SCC-based decompositional methods as described in [21] (for synchronous networks) and [22] (for asynchronous networks). The idea of the methods is as follows: a BN is decomposed into sub-networks called blocks according to the SCCs in the *network structure*; then, attractor detection is performed on each of the blocks; finally, the attractors of the original network are recovered with the attractors in the blocks. Since the attractor detection is performed on blocks, the

Networks	References	# nodes	# attractors		Time (second)		Speedup
			singleton	cyclic	HyTarjan	Decomp	
Tumour	[25]	32	9	0	0.514	0.419	<b>1.23</b>
PC12	[26]	33	7	0	0.268	0.125	<b>2.14</b>
MAPK_r3	[27]	53	20	0	5.387	2.599	<b>2.07</b>
MAPK_r4	[27]	53	16	8	11.314	1.866	<b>6.06</b>
HGF	[28]	66	18	0	17.959	3.604	<b>4.98</b>
apoptosis	[29]	97	512	512	1524.09	84.426	<b>18.05</b>
apoptosis*	[29]	97	1536	512	7926.26	183.22	<b>43.26</b>

Table 1: Evaluation results on five real-life biological systems. *Singleton* refers to an attractor with only one state while *cyclic* refers to an attractor with more than one state.

detection speed is much faster than that on the original large network. We have proved the correctness of the decomposition-based methods in [21, 22], and the implementation of these methods is made publicly available through this newly released version.

The decomposition-based attractor detection methods are implemented based on BDDs, which is an efficient compressed symbolic data structure. We encode the transitions of the states in a BN, known as the transition relations, with BDDs, and implement the decomposition-based methods with efficient BDD operations. For synchronous networks, the attractor detection in each block is performed using either the monolithic or enumerative algorithms as detailed in [23]. For asynchronous networks, it is performed with an adapted hybrid Tarjan algorithm, which is the state-of-the-art BDD based SCC detection algorithm described in Algorithm 7 of [24].

## 4 Case Studies

The evaluation focuses on the new functionality: attractor detection. We have evaluated our decomposition-based method for detecting attractors on both randomly generated networks and real-life biological networks in [21, 22]. It has been shown that our tool is much faster than genYsis [7] in attractor detection for both synchronous and asynchronous networks. In this section, we perform an evaluation of our method on five real-life biological networks to further demonstrate its efficiency. The evaluation results are shown in Table 1. The evaluation results shown here are all based on asynchronous networks. We refer to [21] for evaluation on synchronous networks.

Due to the two variants for the MAPK model and apoptosis model, there are in total seven networks. All the seven networks are analysed under the asynchronous updating scheme. We compute the attractors for all the networks with two different methods, i.e., the hybrid Tarjan method described in Algorithm 7 of [24], and our decomposition-based method. All the experiments are conducted on a computer with an Intel(R) Xeon(R) X5675 @ 3.07GHz CPU. We show the results of the two methods (‘HyTarjan’ for the hybrid Tarjan method and ‘Decomp’ for our decomposition-based method) in Table 1. Note that the decomposition-based method needs to perform attractor detection in each sub-network and this is performed using the hybrid Tarjan method. The results in Table 1 clearly demonstrates that the decomposition-based method outperforms the hybrid Tarjan method. In particular, the decomposition-based method is more

likely to gain a large speedup when the network size is big and the number of cyclic attractors increases.

## 5 Future Developments

There are a few research directions that we want to pursue in the future. We plan to apply the technique of the satisfiability (SAT) solver to further improve the performance of attractor detection in large networks, especially for synchronous BNs. We also plan to implement a few optimisations based on network topologies to further improve the performance of attractor detection. Meanwhile, we will apply our tool for the analysis of large realistic biological networks modelled as PBNs or BNs.

**Acknowledgement.** Qixia Yuan was supported by the National Research Fund, Luxembourg (grant 7814267). This work was also partially supported by the research project SEC-PBN funded by the University of Luxembourg and the ANR-FNR project AlgoReCell (INTER/ANR/15/11191283).

## References

1. Trairatphisan, P., Mizera, A., Pang, J., Tantar, A.A., Schneider, J., Sauter, T.: Recent development and biomedical applications of probabilistic Boolean networks. *Cell Communication and Signaling* **11** (2013) 46
2. Shmulevich, I., Dougherty, E., Zhang, W.: From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE* **90**(11) (2002) 1778–1792
3. Shmulevich, I., Dougherty, E.R.: *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks*. SIAM Press (2010)
4. Pal, R., Datta, A., Bittner, M.L., Dougherty, E.R.: Intervention in context-sensitive probabilistic Boolean networks. *Bioinformatics* **21**(7) (2004) 1211–1218
5. Brun, M., Dougherty, E.R., Shmulevich, I.: Steady-state probabilities for attractors in probabilistic Boolean networks. *Signal Processing* **85**(10) (2005) 1993–2013
6. Hashimoto, R.F., Stagni, H., Higa, C.H.A.: Budding yeast cell cycle modeled by context-sensitive probabilistic Boolean network. In: *Proc. 2009 IEEE International Workshop on Genomic Signal Processing and Statistics, IEEE* (2009) 1–4
7. Garg, A., Xenarios, L., Mendoza, L., DeMicheli, G.: An efficient method for dynamic analysis of gene regulatory networks and in silico gene perturbation experiments. In: *Proc. 11th Annual Conference on Research in Computational Molecular Biology. Volume 4453 of LNCS.*, Springer (2007) 62–76
8. Müssel, C., Hopfensitz, M., Kestler, H.A.: Boolnet-an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics* **26**(10) (2010) 1378–1380
9. Dubrova, E., Teslenko, M.: A SAT-based algorithm for finding attractors in synchronous Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **8**(5) (2011) 1393–1399
10. Chaouiya, C., Naldi, A., Thieffry, D.: Logical modelling of gene regulatory networks with ginsim. *Bacterial Molecular Networks: Methods and Protocols* (2012) 463–479
11. Paulevé, L.: Pint: A static analyzer for transient dynamics of qualitative networks with ipython interface. In: *Proc. 15th conference on Computational Methods for Systems Biology. Volume 10545 of Lecture Notes in Computer Science.*, Springer (2017) 370 – 316

12. Mizera, A., Pang, J., Yuan, Q.: ASSA-PBN: a tool for approximate steady-state analysis of large probabilistic Boolean networks. In: Proc. 13th International Symposium on Automated Technology for Verification and Analysis. Volume 9364 of LNCS., Springer (2015) 214–220
13. Mizera, A., Pang, J., Yuan, Q.: ASSA-PBN 2.0: A software tool for probabilistic Boolean networks. In: Proc. 14th International Conference on Computational Methods in Systems Biology. Volume 9859 of LNCS., Springer (2016) 309–315
14. Zhu, P., Han, J.: Asynchronous stochastic Boolean networks as gene network models. *Journal of Computational Biology* **21**(10) (2014) 771–783
15. Mizera, A., Pang, J., Yuan, Q.: Reviving the two-state markov chain approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2018)
16. Mizera, A., Pang, J., Yuan, Q.: GPU-accelerated steady-state computation of large probabilistic Boolean networks. In: Proc. 2nd Symposium on Dependable Software Engineering: Theories, Tools, and Applications. Volume 9984 of LNCS., Springer (2016) 50–66
17. Mizera, A., Pang, J., Yuan, Q.: Fast simulation of probabilistic Boolean networks. In: Proc. 14th International Conference on Computational Methods in Systems Biology. Volume 9859 of LNCS., Springer (2016) 216–231
18. Naldi, A., Thieffry, D., Chaouiya, C.: Decision diagrams for the representation and analysis of logical models of genetic networks. In: Proc. 5th International Conference on Computational Methods in Systems Biology, Springer (2007) 233–247
19. Zañudo, J.G., Albert, R.: An effective network reduction approach to find the dynamical repertoire of discrete dynamic networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **23**(2) (2013) 025111
20. Klarner, H., Bockmayr, A., Siebert, H.: Computing maximal and minimal trap spaces of Boolean networks. *Natural Computing* **14**(4) (2015) 535–544
21. Mizera, A., Qu, H., Pang, J., Yuan, Q.: A new decomposition method for attractor detection in large synchronous Boolean networks. In: Proc. 3rd Symposium on Dependable Software Engineering. Volume 10606 of LNCS., Springer (2017) 232–249
22. Mizera, A., Pang, J., Qu, H., Yuan, Q.: Taming asynchrony for attractor detection in large Boolean networks (technical report) (2017) <https://arxiv.org/abs/1704.06530>.
23. Yuan, Q., Qu, H., Pang, J., Mizera, A.: Improving BDD-based attractor detection for synchronous Boolean networks. *Science China Information Sciences* **59**(8) (2016) 080101
24. Kwiatkowska, M., Parker, D., Qu, H.: Incremental quantitative verification for Markov decision processes. In: Proc. 41st IEEE/IFIP International Conference on Dependable Systems & Networks, IEEE (2011) 359–370
25. Cohen, D.P.A., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., Calzone, L.: Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Computational Biology* **11**(11) (2015) e1004571
26. Offermann, B., Knauer, S., Singh, A., Fernández-Cachón, M.L., Klose, M., Kowar, S., Busch, H., Boerries, M.: Boolean modeling reveals the necessity of transcriptional regulation for bistability in PC12 cell differentiation. *Frontiers in Genetics* **7** (2016)
27. Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perles, B., Thieffry, D.: Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLOS Computational Biology* **9**(10) (2013) e1003286
28. Singh, A., Nascimento, J.M., Kowar, S., Busch, H., Boerries, M.: Boolean approach to signalling pathway modelling in HGF-induced keratinocyte migration. *Bioinformatics* **28**(18) (2012) 495–501
29. Schlatter, R., Schmich, K., Vizcarra, I.A., Scheurich, P., Sauter, T., Borner, C., Ederer, M., Merfort, I., Sawodny, O.: ON/OFF and beyond - a Boolean model of apoptosis. *PLOS Computational Biology* **5**(12) (2009) e1000595