Reviving the Two-state Markov Chain Approach

Andrzej Mizera, Jun Pang, and Qixia Yuan

Abstract—Probabilistic Boolean networks (PBNs) is a well-established computational framework for modelling biological systems. The steady-state dynamics of PBNs is of crucial importance in the study of such systems. However, for large PBNs, which often arise in systems biology, obtaining the steady-state distribution poses a significant challenge. In this paper, we revive the two-state Markov chain approach to solve this problem. This paper contributes in three aspects. First, we identify a problem of generating biased results with the approach and we propose a few heuristics to avoid such a pitfall. Secondly, we conduct an extensive experimental comparison of the extended two-state Markov chain approach and another approach based on the Skart method. We analyse the results with machine learning techniques and we show that statistically the two-state Markov chain approach has a better performance. Finally, we demonstrate the potential of the extended two-state Markov chain approach on a case study of a large PBN model of apoptosis in hepatocytes.

Index Terms—Probabilistic Boolean networks, Markov chains, steady-state analysis, approximation.

✦

1 INTRODUCTION

COMPUTATIONAL modelling plays a prominent role in providing a system-level understanding of processes that take place in a living cell. However, it faces significant challenges when modelling realistic biological systems due to the size of the state-space that needs to be considered. Hence, profound understanding of biological processes asks for the development of new methods that would provide means for formal analysis and reasoning about large systems.

We focus on the steady-state dynamics of biological processes modelled as discrete-time Markov chains (DTMCs). This is the case, for example, when the biological system under study is cast into the mathematical/computational framework of probabilistic Boolean networks (PBNs) [1], [2], [3]. In these or other discrete-time models, e.g., dynamic Bayesian networks, the real (considered as continuous) time is not modelled. Instead, the evolution of the system is abstracted as a sequence of consecutive events. These coarse-grained models have been successfully applied in many systems biology studies and proved their predictive power [4]. In fact, for the study of large regulatory systems they remain the only reasonable solution. Extrapolating the ordinary differential equations model of a single elementary building block of the network (e.g., a gene) to the whole large system would result in a prohibitively complex model. Moving towards a higher-level description by ignoring the molecular details allows to grasp the system-level behaviour of the network [5]. In consequence, these coarse-grained formalisms are broadly applied in systems biology. One of the key aspects in the analysis of such dynamic systems is the comprehension of their steady-state (long-run) behaviour. For example, cellular phenotypes were hypothesised to be characterised by a collection of sets of states, referred to as *attractors*, that the system reaches and in which it remains forever in a long run. Another complementary conjecture is that attractors correspond to functional cellular states such as *proliferation*, *apoptosis*, or *differentiation* [6]. These interpretations may cast new light on the understanding of cellular homeostasis and cancer progression [1]. In this work, we focus on the computation of

steady-state probabilities which are crucial for the determination of long-run influences and sensitivities. These are measures that quantify the impact of genes on other genes and that enable the identification of elements with highest impact.

So far the huge-state space, which often characterises dynamical models of biological systems, tempers the application of the above mentioned techniques in the analysis of realistic biological systems. In fact, approximations with the use of Markov chain Monte Carlo (MCMC) techniques are the only viable solution to this problem [7]. However, due to the difficulties with the assessment of the convergence rate to the steady-state distribution (see, e.g., [8]), certain care is required when applying these methods in practice. A number of statistical methods exist which allow to empirically determine when the simulation can be stopped, with the Skart method [9] considered to be the state-of-the-art technique. In this study, we focus on another method, i.e., the two-state Markov chain approach. It was introduced in 1992 by Raftery and Lewis, and Shmulevich et al. [7] proposed its application to the analysis of PBNs in 2003. However, to the best of our knowledge, since then it has not been widely applied for the analysis of large PBNs. In this paper, we aim to revive it by demonstrating its usefulness for approximating steady-state probabilities of large PBNs, which often arise in systems biology as computational models of cellular gene regulatory networks.

This paper contributes in three aspects. First, we identify a problem of generating biased results by the original two-state Markov chain approach, due to the size of the initial sample with which the approach needs to start. We propose three heuristics for the approach to avoid some unfortunate initialisations. Secondly, we perform an extensive evaluation and comparison of the extended two-state Markov chain approach with the state-of-the-art Skart method on a large number of randomly generated PBNs. We apply techniques originating from machine learning to analyse the obtained results. We show that in the context of gene regulatory networks modelled with PBNs, actually the two-state Markov chain approach seems to perform better than the Skart method in most cases in terms of computational cost. Thirdly, we demonstrate the potential of the two-state Markov chain approach on a study of a large, 91-node PBN model of apoptosis in hepatocytes. In particular, we show that the two-state

• A. Mizera, J. Pang, and Q. Yuan are with the Computer Science and Communications Research Unit, University of Luxembourg.
E-mail: firstname.lastname@uni.lu

Markov chain approach facilitates the quantification of both the steady-state influences of genes on their target genes and long-run sensitivities in large networks.

2 PRELIMINARIES

2.1 Finite discrete-time Markov chains (DTMCs)

Let S be a finite set of states. A (first-order) DTMC is an S-valued stochastic process $\{X_t\}_{t\in\mathbb{N}}$ with the property that the next state is independent of the past states given the present state. Formally, $\mathbb{P}(X_{t+1} = s_{t+1} \,|\, X_t = s_t, X_{t-1} = s_{t-1}, \ldots, X_0 = s_0) = \mathbb{P}(X_{t+1} = s_{t+1} \,|\, X_t = s_t)$ for all $s_{t+1}, s_t, \ldots, s_0 \in S$. Here, we consider *time-homogenous* Markov chains, i.e., chains where $\mathbb{P}(X_{t+1} = s' \,|\, X_t = s)$, denoted $P_{s,s'}$, is independent of t for any states $s, s' \in S$. The transition matrix $P = (P_{s,s'})_{s,s'\in S}$ satisfies $P_{s,s'} \geqslant 0$ and $\sum_{s'\in S} P_{s,s'} = 1$ for all $s \in S$. We denote by π a probability distribution on S. If $\pi = \pi P$, then π is a *stationary distribution* of the DTMC (also referred to as the *invariant distribution*). A path of length n is a sequence $s_1 \to s_2 \to \cdots \to s_n$ such that $P_{s_i, s_{i+1}} > 0$ and $s_i \in S$ for $i \in \{1, 2, \ldots, n\}$. State $q \in S$ is *reachable* from state $p \in S$ if there exists a path such that $s_1 = p$ and $s_n = q$. A DTMC is *irreducible* if any two states are reachable from each other. The period of a state is defined as the greatest common divisor of the lengths of all paths that start and end in the state. A DTMC is *aperiodic* if all states in S are of period 1. A finite state DTMC is called *ergodic* if it is irreducible and aperiodic. By the famous ergodic theorem for DTMCs [10], an ergodic chain has a unique stationary distribution being its *limiting distribution* (also referred to as the *steady-state distribution*) given by $\lim_{n\to\infty} \pi_0 P^n$, where π_0 is any initial probability distribution on S. In consequence, the limiting distribution for an ergodic chain is independent of the choice of π_0. The steady-state distribution can be estimated from any initial distribution by iteratively multiplying it by P.

The evolution of a first-order DTMC can be described by a stochastic recurrence sequence $X_{t+1} = \phi(X_t, U_{t+1})$, where $\{U_t\}_{t\in\mathbb{N}}$ is an independent sequence of uniformly distributed real random variables over $[0,1]$ and the transition function $\phi : S \times [0,1] \to S$ satisfies the property that $\mathbb{P}(\phi(s, U) = s') = P_{s,s'}$ for any states $s, s' \in S$ and for any U, a real random variable uniformly distributed over $[0,1]$. When S is partially ordered and the transition function $\phi(\cdot, u)$ is monotonic, then the chain is said to be *monotone* ([11]).

2.2 Probabilistic Boolean networks (PBNs)

A PBN $G(V, \mathscr{F})$ consists of a set of binary-valued nodes (also referred to as genes) $V = \{v_1, v_2, \ldots, v_n\}$ and a list of sets $\mathscr{F} = (F_1, F_2, \ldots, F_n)$. For each $i \in \{1, 2, \ldots, n\}$ the set $F_i = \{f_1^{(i)}, f_2^{(i)}, \ldots, f_{l(i)}^{(i)}\}$ is a collection of predictor functions for node v_i, where $l(i)$ is the number of predictor functions for v_i. Each $f_j^{(i)} \in F_i$ is a Boolean function defined with respect to a subset of nodes referred to as parent nodes of v_i. There is a probability distribution on each $F_i \in \mathscr{F}$: $c_j^{(i)}$ is the probability of selecting $f_j^{(i)} \in F_i$ as the next predictor for v_i and it holds that $\sum_{j=1}^{l(i)} c_j^{(i)} = 1$. We denote by $v_i(t)$ the value of node v_i at time point $t \in \mathbb{N}$. The state space of the PBN is $S = \{0,1\}^n$ and it is of size 2^n. The state of the PBN at time t is determined by $s(t) = (v_1(t), v_2(t), \ldots, v_n(t))$. The dynamics of the PBN is given by the sequence $(s(t))_{t=0}^{\infty}$. We consider here *independent* PBNs where predictor functions for different nodes are selected

independently of each other. The transition from $s(t)$ to $s(t+1)$ is conducted by randomly selecting a predictor function for each node v_i from F_i and by synchronously updating the node values in accordance with the selected functions. There are $N = \prod_{i=1}^n l(i)$ different ways in which the predictors can be selected for all n nodes. These combinations are referred to as *realisations* of the PBN and are represented as n-dimensional function vectors $\boldsymbol{f}_k = (f_{k_1}^{(1)}, f_{k_2}^{(2)}, \ldots, f_{k_n}^{(n)}) \in F_1 \times F_2 \times \ldots \times F_n$, where $k \in \{1, 2, \ldots, N\}$ and $k_i \in \{1, 2, \ldots, l(i)\}$. A realization selected at time t is referred to as $\boldsymbol{F}_t$. Due to independence, $\mathbb{P}(\boldsymbol{f}_k) = \mathbb{P}(\boldsymbol{F}_t = \boldsymbol{f}_k) = \prod_{i=1}^n c_{k_i}^{(i)}$.

In PBNs *with perturbations*, a perturbation parameter $p \in (0,1)$ is introduced to sample the perturbation vector $\boldsymbol{\gamma}(t) = (\gamma_1(t), \gamma_2(t), \ldots, \gamma_n(t))$, where $\gamma_i(t) \in \{0,1\}$ and $\mathbb{P}(\gamma_i(t) = 1) = p$ for all t and $i \in \{1, 2, \ldots, n\}$. Perturbations provide an alternative way to regulate the dynamics of a PBN: the next state is determined as $s(t+1) = \boldsymbol{F}_t(s(t))$ if $\boldsymbol{\gamma}(t) = \boldsymbol{0}$ and as $s(t+1) = s(t) \oplus \boldsymbol{\gamma}(t)$ otherwise, where $\oplus$ is the exclusive or operator for vectors. The perturbations, by the latter update formula, allow the system to move from any state to any other state in one single transition, hence render the underlying Markov chain irreducible and aperiodic. Therefore, the dynamics of a PBN with perturbations can be viewed as an ergodic DTMC [2]. The transition matrix is given by $P_{s,s'} = (1-p)^n \sum_{k=1}^N \mathbb{1}_{[\boldsymbol{f}_k(s)=s']} \mathbb{P}(\boldsymbol{f}_k) + (1 - (1-p)^n) p^{\eta(s,s')} (1-p)^{n-\eta(s,s')}$, where $\mathbb{1}$ is the indicator function and $\eta(s, s')$ is the Hamming distance between states $s, s' \in S$. According to the ergodic theory, adding perturbations to any PBN assures the long-run dynamics of the resulting PBN is governed by a unique limiting distribution, convergence to which is independent of the choice of the initial state. However, the perturbation probability value should be chosen carefully, not to dilute the behaviour of the original PBN. In this way the 'mathematical trick', although introduces some noise to the original system, allows to significantly simplify the analysis of the steady-state behaviour.

The density of a PBN is measured with its function number and parent nodes number. For a PBN G, its density is defined as $\mathscr{D}(G) = \frac{1}{n} \sum_{i=1}^{N_F} \omega(i)$, where n is the number of nodes in G, N_F is the total number of predictor functions in G, and $\omega(i)$ is the number of parent nodes for the ith predictor function.

Within the framework of PBNs the concept of influences is defined; it formalizes the impact of parents nodes on a target node and enables its quantification ([12]). The concept is based on the notion of a partial derivative of a Boolean function f with respect to variable x_j ($1 \leq j \leq n$):

$$\frac{\partial f(x)}{\partial x_j} = f(x^{(j,0)}) \oplus f(x^{(j,1)}),$$

where $\oplus$ is addition modulo 2 (exclusive OR) and for $l \in \{0,1\}$

$$x^{(j,l)} = (x_1, x_2, \ldots, x_{j-1}, l, x_{j+1}, \ldots, x_n).$$

The *influence of node x_j on function f* is the expected value of the partial derivative with respect to the probability distribution $D(x)$:

$$I_j(f) = \mathbb{E}_D\left[\frac{\partial f(x)}{\partial x_j}\right] = \mathbb{P}\left\{\frac{\partial f(x)}{\partial x_j} = 1\right\} = \mathbb{P}\{f(x^{(j,0)}) \neq f(x^{(j,1)})\}.$$

Let now F_i be the set of predictors for x_i with corresponding probabilities $c_j^{(i)}$ for $j = 1, \ldots, l(i)$ and let $I_k(f_j^{(i)})$ be the influence

of node x_k on the predictor function $f_j^{(i)}$. Then, the *influence of node x_k on node x_i* is defined as:

$$I_k(x_i) = \sum_{j=1}^{l(i)} I_k(f_j^{(i)}) \cdot c_j^{(i)}.$$

The *long-term influences* are the influences computed when the distribution $D(x)$ is the stead-state distribution of the PBN.

We define and consider in this study two types of long-run sensitivities.

Definition 2.1. The *long-run sensitivity with respect to selection probability perturbation* is defined as

$$s_c[c_j^{(i)} = p] = \|\tilde{\pi}[c_j^{(i)} = p] - \pi\|_l,$$

where $\|\cdot\|_l$ denotes the l-norm, π is the steady-state distribution of the original PBN, $p \in [0,1]$ is the new value for $c_j^{(i)}$, and $\tilde{\pi}[c_j^{(i)} = p]$ is the steady-state probability distribution of the PBN perturbed as follows. The jth selection probability for node x_i is replaced with $\tilde{c}_j^{(i)} = p$ and all $c_k^{(i)}$ selection probabilities for $k \in I_{-j} = \{1, 2, \dots, j-1, j+1, \dots, l(i)\}$ are replaced with

$$\tilde{c}_k^{(i)} = c_k^{(i)} + (c_j^{(i)} - p) \cdot \frac{c_k^{(i)}}{\sum_{l \in I_{-j}} c_l^{(i)}},$$

The remaining selection probabilities of the original PBN are unchanged.

Definition 2.2. The *long-run sensitivity with respect to permanent on/off perturbations of a node x_i* as

$$s_g[x_i] = \max\{\|\tilde{\pi}[x_i \equiv 0] - \pi\|_l, \|\tilde{\pi}[x_i \equiv 1] - \pi\|_l\},$$

where π, $\tilde{\pi}[x_i \equiv 0]$, and $\tilde{\pi}[x_i \equiv 1]$ are the steady-state probability distributions of the original PBN, of the original PBN with all $f^{(i)} \in F_i$ replaced by $\tilde{f}^{(i)} \equiv 0$, and all $f^{(i)} \in F_i$ replaced by $\tilde{f}^{(i)} \equiv 1$, respectively.

Notice that the definition of long-run sensitivity with respect to permanent on/off perturbations is similar but not equivalent to the definition of long-run sensitivity with respect to 1-gene function perturbation of [12].

2.3 The Two-state Markov Chain Approach

The two-state Markov chain approach [13] is a method for estimating the steady-state probability of a subset of states of a DTMC. In this approach the state space of an arbitrary DTMC is split into two disjoint sets, referred to as meta states. One of the meta states, numbered 1, is the subset of interest and the other, numbered 0, is its complement. The steady-state probability of meta state 1, denoted π_1, can be estimated by performing simulations of the original Markov chain. For this purpose a two-state Markov chain abstraction of the original DTMC is considered. Let $\{Z_t\}_{t \geq 0}$ be a family of binary random variables, where Z_t is the number of the meta state the original Markov chain is in at time t. $\{Z_t\}_{t \geq 0}$ is a binary (0-1) stochastic process, but in general it is not a Markov chain. However, as argued in [13], a reasonable assumption is that the dependency in $\{Z_t\}_{t \geq 0}$ falls off rapidly with lag. Therefore, a new process $\{Z_t^{(k)}\}_{t \geq 0}$, where $Z_t^{(k)} = Z_{1+(t-1)k}$, will approximately a first-order Markov chain for k large enough. A procedure for determining appropriate k is given in [13]. The first-order chain consists of the two meta states with transition probabilities
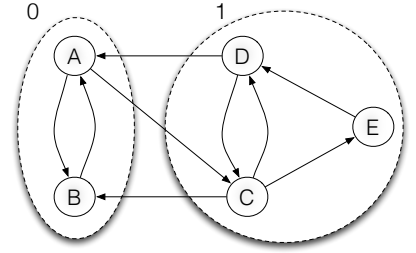
(a) Original DTMC
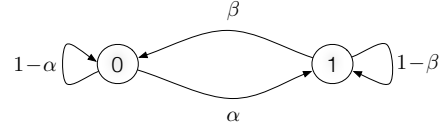
(b) Two-state DTMC

Fig. 1: Conceptual illustration of the idea of the two-state Markov chain construction. (a) The state space of the original discrete-time Markov chain is split into two meta states: states A and B form meta state 0, while states D, C, and E form meta state 1. The split of the state space into meta states is marked with dashed ellipses. (b) Projecting the behaviour of the original chain on the two meta states results in a binary (0-1) stochastic process. After potential subsampling, it can be approximated as a first-order, two-state Markov chain with the transition probabilities α and β set appropriately.

α and β between them. See Figure 1 for an illustration of the construction of this abstraction.

The steady-state probability estimate $\hat{\pi}_1$ is computed from a simulated trajectory of the original DTMC. The key point is to determine the optimal length of the trajectory. Two requirements are imposed. First, the abstraction of the DTMC, i.e., the two-state Markov chain, should converge close to its steady-state distribution $\pi = [\pi_0 \ \pi_1]$. Formally, t satisfying $|\mathbb{P}[Z_t^{(k)} = i \,|\, Z_0^{(k)} = j] - \pi_i| < \varepsilon$ for a given $\varepsilon > 0$ and all $i, j \in \{0, 1\}$ needs to be determined. t is the so-called 'burn-in' period and determines the part of the trajectory of the two-state Markov chain that needs to be discarded. Second, the estimate $\hat{\pi}_1$ is required to satisfy $\mathbb{P}[\pi_1 - r \leq \hat{\pi}_1 \leq \pi_1 + r] \geq s$, where r is the required precision and s is a specified confidence level. This condition is used to determine the length of the second part of the trajectory used to compute $\hat{\pi}_1$, i.e., the sample size. Now, the total required trajectory length of the original DTMC is then given by $M + N$, where $M = 1 + (t-1)k$ and $N = 1 + (\lceil n(\alpha, \beta)\rceil - 1)k$, where $t = \lceil m(\alpha, \beta)\rceil$. The functions m and n depend on the transitions probabilities α and β and are given by

$$m(\alpha, \beta) = \frac{\log\left(\frac{\varepsilon(\alpha+\beta)}{\max(\alpha,\beta)}\right)}{\log(|1 - \alpha - \beta|)}$$

and

$$n(\alpha, \beta) = \frac{\alpha\beta(2 - \alpha - \beta)}{(\alpha + \beta)^3} \frac{\left(\Phi^{-1}(\frac{1}{2}(1+s))\right)^2}{r^2},$$

where Φ^{-1} is the inverse of the standard normal cumulative distribution function. For the completeness of the presentation, the detailed derivations of the expressions for m and n are given in the Appendices A and B.

Since α and β are unknown, they need to be estimated. This is achieved iteratively in the two-state Markov chain approach

of [13]. It starts with sampling an arbitrary initial length trajectory, which is then used for estimating the values of α and β. M and N are calculated based on these estimates. Next, the trajectory is extended to reach the required length, and α and β values are re-estimated. The new estimates are used to re-calculate M and N. This process is iterated until $M + N$ is smaller than the current trajectory length. Finally, the resulting trajectory is used to estimate the steady-state probability of meta state 1. For more details, see [13]. Notice however the small oversights in the formulas for m (absolute value missing in the denominator) and n (the inverse of Φ should be used) therein.

3 TWO-STATE MARKOV CHAIN APPROACH: THE INITIALISATION PROBLEM

Shmulevich et al. [7] proposed to use the two-state Markov chain approach for analysing the steady-state dynamics of PBNs in 2003. However, since then it has not been widely applied. We aim to revive the two-state Markov chain approach by demonstrating its usefulness for approximating steady-state probabilities of large PBNs. In this section, we first identify an initialisation problem of the original approach due to the size of the initial sample, this particular problem can lead to biased results. We then propose three heuristics to extend the approach for avoiding unfortunate initialisations.

Given good estimates of α and β, the theory of the two-state Markov chain approach presented above guarantees that the obtained value satisfies the imposed precision requirements. However, the method starts with generating a trajectory of the original DTMC of an arbitrarily chosen initial length, i.e., $M_0 + N_0 = 1 + (m_0 - 1)k + 1 + (n_0 - 1)k$, where m_0 is the 'burn-in' period and n_0 is the sample size of the two-state Markov chain abstraction. An unfortunate choice may lead to initial estimates of α and β that are biased and result in the new values of M and N such that $M + N$ is either smaller or not much larger than the initial $M_0 + N_0$. In the former case the algorithm stops immediately with the biased values for α, β and, more importantly, with an estimate for the steady-state probability that does not satisfy the precision requirements. The second case may lead to the same problem. As an illustration we considered a two-state Markov chain with $\alpha = \frac{24}{11873}$ (0.0020214) and $\beta = \frac{24}{25}$ (0.96). The steady-state probability distribution was [0.997899 0.002101]. With $k = 1$, $\varepsilon = 10^{-6}$, $r = 10^{-3}$, $s = 0.95$, $m_0 = 5$, and $n_0 = 1,920$ the first estimated values for α and β were $\frac{1}{1918}$ (0.0005214) and 1, respectively. This subsequently led to $M = 2$ and $N = 1,999$, resulting in a request for the extension of the trajectory by 76. After the extension, the new estimates for α and β were $\frac{1}{1997}$ and 1, respectively. These estimates gave $M = 2$, $N = 1,920$, and the algorithm stopped. The estimated steady-state probability distribution was [0.99950 0.00050], which was outside the pre-specified precision interval given by r. Independent 10^4 runs resulted in estimates of the steady-state probabilities that were outside the pre-specified precision interval 10% of times. Given the rather large number of repetitions, it can be concluded that the specified 95% confidence interval was not reached.

The reason for the biased result is the unfortunate initial value for n_0 and the fact that the real value of α is small. In the initialisation phase the value of α is underestimated and $\lceil n(\alpha, \beta) \rceil$ calculated based on the estimated values of α and β is almost the same as n_0. Hence, subsequent extension of the trajectory does not

provide any improvement to the underestimated value of α since the elongation is too short.

To identify and avoid some of such pitfalls, we consider a number of cases and formulate some of the conditions in which the algorithm may fail to achieve the specified precision. To start, let n_0 be the initial size of the sample used for initial estimation of α and β. Neither α nor β is zero. It might be the case that the initial sample size is not big enough to provide non-zero estimates for both α and β. If this is the case, n_0 is doubled and the trajectory is elongated to collect a sample of required size. This is repeated iteratively until non-zero estimates for α and β are obtained. In the continuation we assume that n_0 provides non-zero estimates for both α and β. Then, the smallest possible estimates for both α and β are greater than $\frac{1}{n_0}$.

For a moment, let us set an upper bound value for n_0 to be 10^4. For most cases this boundary value is reasonable. Notice however that this is the case only if the real values of α and β are larger than 10^{-4}. In general, the selection of a proper value for n_0 heavily depends on the real values of α and β, which are unknown *a priori*. From what was stated above, it follows that both first estimates for α and β are greater than 10^{-4}. The following cases are possible.

(1) If both α and β are small, e.g., less than 0.1, then we have that $10^{-4} < \alpha, \beta < 0.1$ and $n(\alpha, \beta) > 72,765$ as can be seen by investigating the $n(\cdot, \cdot)$ function. In this case the sample size is increased more than 7-fold which is reasonable since the two-state Markov chain seems to be bad-mixing by the first estimates of the values for α and β and the algorithm asks for a significant increase of the sample size. We therefore conclude that the bad-mixing case is properly handled by the algorithm.

(2) Both first estimates of α and β are close to 1. If $\alpha, \beta \in [0.7, 0.98]$, the value of $n(\alpha, \beta)$ is larger than 19,000. If both $\alpha, \beta > 0.98$, then the size of the sample drops, but in this case the Markov chain is highly well-mixing and short trajectories are expected to provide good estimates.

(3) The situation is somewhat different if one of the parameters is estimated to be small and the other is close to 1 as in the example described above. The extension to the trajectory is too small to significantly change the estimated value of the small parameter and the algorithm halts.

Considering the above cases leads us to the observation that the following situation needs to be treated with care: *The estimated value for one of the parameters is close to $\frac{1}{n_0}$, the value of the second parameter is close to 1, and $n(\alpha, \beta)$ is either smaller or not significantly larger than n_0.*

First approach: pitfall avoidance. To avoid this situation, we determine n_0 which in principle could lead to inaccurate initial estimates of α or β and such that the next sample size given by $\lceil n(\alpha, \beta) \rceil$ would practically not allow for an improvement of the estimates. As stated above, the 'critical' situation may take place when one of the parameters is estimated to be very small, i.e., close to $\frac{1}{n_0}$, and the increase in the sample size is not significant enough to improve the estimate. If the initial estimate is very small, the real value is most probably also small, but the estimate is not accurate. If the value is underestimated to the lowest possible value, i.e., $\frac{1}{n_0}$, on average the improvement can take place only if the sample size is increased at least by n_0. Therefore, with the trade-off between the accuracy and efficiency of the method in mind, we propose the sample size to be increased at least by n_0. Then the 'critical' situation condition is $n(\alpha, \beta) < 2n_0$. By

analysing the function $n(\cdot, \cdot)$ as described in details in Appendix D, we can determine the values of n_0 that are 'safe', i.e., which do not satisfy the 'critical' condition. We present them in Table 1 for a number of values for r and s.

Second approach: controlled initial estimation of α and β. The formula for n is asymptotically valid provided that the values for α and β are known. However, these values are not known *a priori* and they need to be estimated. Unfortunately, the original approach does not provide any control over the quality of the initial estimate of the values of these parameters. In certain situation, e.g., as in the case discussed above, the lack of such control mechanism may lead to results with worse statistical confidence level than the specified one given by s. In the discussed example $s = 95\%$, but this value was not reached in the performed experiment. In order to address this problem, we propose to extend the initial phase of the two-state approach algorithm in the following way. The algorithm samples a trajectory of the original DTMC and estimates the values of α and β. We denote the estimates as $\hat{\alpha}$ and $\hat{\beta}$, respectively. Next, the algorithm computes the sample size required to reach the s confidence level that the true value of $\min(\alpha, \beta)$ is within a certain interval. For definiteness, we assume from now on that $\hat{\alpha} < \hat{\beta}$, which suggests that $\min(\alpha, \beta) = \alpha$. During the execution of the procedure outlined in the following the inequality may be inverted. If this is the case, the algorithm makes corresponding change in the consideration of α and β.

The aim is to have a good estimate for α. Notice that the smallest possible initial value of $\hat{\alpha}$ is greater than $\frac{1}{n_0}$. We refer to $\frac{1}{n_0}$ as the *resolution of estimation*. Given the resolution, one cannot distinguish between values of α in the interval $(\hat{\alpha} - \frac{1}{n_0}, \hat{\alpha} + \frac{1}{n_0})$. In consequence, if $\alpha \in (\hat{\alpha} - \frac{1}{n_0}, \hat{\alpha} + \frac{1}{n_0})$, then the estimated value $\hat{\alpha}$ should be considered as optimal. Hence, one could use this interval as the one which should contain the real value with specified confidence level. Nevertheless, although the choice of this interval usually leads to very good results, as experimentally verified, the results are obtained at the cost of large samples which make the algorithm stop immediately after the initialisation phase. Consequently, the computational burden is larger than would be required by the original algorithm to reach the desired precision specified by r and s parameters in most cases. In order to reduce this unnecessary overhead, we consider the interval $(\hat{\alpha} - \frac{\hat{\alpha}}{2}, \hat{\alpha} + \frac{\hat{\alpha}}{2})$, which is wider than the previous one whenever $\hat{\alpha} > \frac{1}{n_0}$ and leads to smaller sample sizes.

The two-state Markov chain consists of two states 0 and 1, i.e., the two meta states of the original DTMC. We set α as the probability of making the transition from state 0 to state 1 (denoted as $0 \to 1$). The estimate $\hat{\alpha}$ is computed as the ratio of the number of transitions from state 0 to state 1 to the number of transition from state 0. Let $n_{0,\alpha}$ be the number of transitions in the sample starting from state 0. Let X_i, $i = 1, 2, \ldots, n_{0,\alpha}$, be a random variable defined as follows: X_i is 1 if ith transition from meta-state 0 is $0 \to 1$ and 0 otherwise.

Notice that state 0 is an accessible atom in the terminology of the theory of Markov chains, i.e., the Markov chain regenerates after entering state 0, and hence the random variables X_i, $i = 1, 2, \ldots, n_{0,\alpha}$, are independent. They are Bernoulli distributed with parameter α. The unbiased estimate of the population variance from the sample, denoted $\hat{\sigma}^2$, is given by $\hat{\sigma}^2 = \hat{\alpha} \cdot (1 - \hat{\alpha}) \cdot \frac{n_{0,\alpha}}{n_{0,\alpha} - 1}$. Due to independence, $\hat{\sigma}^2$ is also the asymptotic variance and, in consequence, the sample size that provides the specified confidence level for the estimate of the value of α is given by

$n_{\alpha,s}(\hat{\alpha}, n_{0,\alpha}) = \hat{\alpha} \cdot (1 - \hat{\alpha}) \cdot \frac{n_{0,\alpha}}{n_{0,\alpha} - 1} \cdot \left(\frac{\Phi^{-1}(\frac{1}{2}(1+s))}{\hat{\alpha}/2} \right)^2$. The Markov chain is in state 0 with steady-state probability $\frac{\beta}{\alpha+\beta}$. Then, given that the chain reached the steady-state distribution, the expected number of regenerations in a sample of size n is given by $\frac{n \cdot \beta}{\alpha+\beta}$. Therefore, the sample size used to estimate the value of α with the specified confidence level s is given by $n_\alpha = \frac{\alpha+\beta}{\beta} \cdot n_{\alpha,s}(\hat{\alpha}, n_{0,\alpha})$. As the real values of α and β are unknown, the estimated values $\hat{\alpha}$ and $\hat{\beta}$ can be used in the above formula. If the computed n_α is bigger than the current number of transitions $n_{0,\alpha}$, we extend the trajectory to reach n_α transitions from 0 to 1 and re-estimate the values for α and β using the extended trajectory. We repeat this process until the computed n_α value is smaller than the number of transitions used to estimate α. In this way, good initial estimates for α and β are obtained and the original two-state Markov chain approach using the formula for $n(\alpha, \beta)$ is run.

Third approach: simple heuristics. When performing the initial estimation of α and β, we require both the count of transitions from state 0 to state 1 and the count of transitions from meta-state 1 to state 0 be at least 3. If this condition is not satisfied, we proceed by doubling the length of the trajectory. In this way the problem of reaching the resolution boundary is avoided. Our experiments showed that this simple approach in many cases led to good initial estimates of the α and β probabilities.

Discussions. The first approach provides us with safe initial starting points. As can be seen in Table 1, there might however be no safe starting point in certain conditions. Nevertheless, the first approach can be used in the initialisation phase of the other two approaches. The second approach introduces a new iteration process to provide a good estimate of α or β. The third one modifies the two-state Markov chain approach by adding only one extra restriction and therefore is the most simple one. We have verified with experiments that the last two approaches have the potential to make the two-state Markov chain approach meet the predefined precision requirement even in the case of an unlucky initial sample size. As a small example, we show in Table 2 the results for verifying two PBNs each of eight nodes. For each of the PBNs, we compute the steady-state probability for one subset of states using three different approaches: 1) the original two-state Markov chain approach (columns 'Original'), the proposed second approach (columns '2nd') and the proposed third approach (columns '3rd'). The precision and confidence level are set to 0.001 and 0.95 respectively. We repeat the computation for 1000 times and count the percentage of times that the calculated result is within the precision requirement (shown in columns labelled 'Computed confidence level'). As can be seen in Table 2, the original two-state Markov chain approach fails to meet the confidence level requirement while the both proposed approaches can meet the requirement. Due to its simplicity, we use the third approach in the remaining of the paper.

4 EVALUATION

In this section, we focus on verifying the performance of the two-state Markov chain approach with another related method called the Skart method [9]. We use the tool ASSA-PBN [14], [15] as the platform for this verification. ASSA-PBN is a tool specially designed for steady-state analysis of large PBNs; it includes the two-state Markov chain approach with the simple heuristics presented in Section 3 and the Skart method. For the steady-state analysis of large PBNs, applications of these two

TABLE 1: Ranges of integer values for n_0 that do not satisfy the 'critical' condition $n(\alpha,\beta) < 2n_0$ for the given values of r and s.

r	0.01			0.001			0.0001		
s	0.9	0.95	0.975	0.9	0.95	0.975	0.9	0.95	0.975
$n_0 \in$	$\emptyset$	$[2,136]$	$\emptyset$	$[2,1161]$	$[2,1383]$	$[2,1582]$	$[2,11628]$	$[2,13857]$	$[2,15847]$

TABLE 2: Performance of the second and third approaches.

Model	Computed confidence level			Average sample size		
	Original	2nd	3rd	Original	2nd	3rd
PBN 1	88.3	96.7	95.5	9265	9040	9418
PBN 2	87.8	99.3	96.5	7731	13635	8201

methods necessitate generation of trajectories of significant length. To make this efficient, we applied the alias method [16] to sample the consecutive trajectory state. This enables ASSA-PBN, e.g, to simulate 12,580 steps within 1s for a 2,000 nodes PBN, which is hundreds of times faster than the related tool optPBN [17].

In Section 4.1, we briefly describe the Skart method. We present an empirical comparison of the performance of these two methods in Section 4.2.

4.1 The Skart Method

We choose the Skart method [9] as a reference for the evaluation of the performance of the two-state Markov chain approach. The Skart method is a successor of ASAP3, WASSP, and SBatch methods, which are all based on the idea of batch means [9]. It is a procedure for on-the-fly statistical analysis of the simulation output, asymptotically generated in accordance with a steady-state distribution. Usually it requires an initial sample of size smaller than other established simulation analysis procedures [9]. Briefly, the algorithm partitions a long simulation trajectory into batches, for each batch computes a mean and constructs an interval estimate using the batch means. Further, the interval estimate is used by Skart to decide whether a steady state distribution is reached or more samples are required. For a more detailed description of this method, see [9].

The Skart method differs in three key points with the two-state Markov chain approach. First, it specifies the initial trajectory length to be at least 1,280, while for the two-state Markov chain approach this information is not provided. This difference, however, does not change the fact that the two methods can provide the same accuracy guarantee providing that the unlucky choice of the initial trajectory length of the two-state Markov chain approach is fixed as mentioned in the previous section. Second, the Skart method applies the student distribution for skewness adjustment while the two-state approach makes use of the normal distribution for confidence interval calculations. Thirdly, the two-state Markov chain approach does not require to keep track of the simulated trajectories; instead, the statistics (e.g., the α and β as in Figure 1) of the trajectories are enough. the Skart method, however, requires to keep track of the simulated trajectories, which consumes a large memory in the cases of large size trajectories.

4.2 Performance evaluation

To compare the performance of the two methods, we randomly generated 882 different PBNs using ASSA-PBN. ASSA-PBN can randomly generate a PBN which satisfies structure requirements given in the form of five input parameters: the node number, the minimum and the maximum number of predictor functions per

node, finally the minimum and maximum number of parent nodes for a predictor function. We generated PBNs with node numbers from $\{15,30,80,100,150,200,300,400,500,1000,2000\}$. We assigned the obtained PBNs into three different classes with respect to the density measure $\mathscr{D}$: *dense models* with density 150–300, *sparse models* with density around 10, and *in-between models* with density 50–100. The two-state Markov chain approach and the Skart method were tested on these PBNs with precision r set to the values in $\{10^{-2}, 5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}, 5 \times 10^{-6}, 1 \times 10^{-6}\}$. We set ε to 10^{-10} for the two-state Markov chain approach and s to 0.95 for both methods.

The experiments were performed on a HPC cluster, with CPU speed ranging between 2.2GHz and 3.07GHz. ASSA-PBN is implemented in Java and the initial and maximum Java virtual machine heap size were set to 503MB and 7.86GB, respectively. We collected 5596 valid (precision being smaller than probability) results with the information on the PBN node number, its density class, the precision value, the estimated steady-state probabilities computed by the two methods, and their CPU time costs. The steady-state probabilities computed by the two methods are comparable in all the cases (data not shown in the paper). For each experimental result i, we compare the time costs of the two methods. Let $t_{TS}(i)$ and $t_{Skart}(i)$ be the time cost for the two-state Markov chain approach and the Skart method, respectively. We say that the two-state Markov chain approach is by k per cent faster than the Skart method if $\frac{(t_{Skart}(i) - t_{TS}(i))}{t_{Skart}(i)} \geqslant \frac{k}{100}$. The definition for the Skart method to be faster than the two-state Markov chain approach is symmetric. In Table 3 we show the percentage of cases in which the two-state approach was by k per cent faster than Skart and vice versa for different k. In general, *in nearly 70% of the results, the two-state Markov chain approach was faster than the Skart method*. The number of cases the two-state Markov chain approach was faster than the Skart method is also larger than in the opposite case.

Next, we analyse the results with a machine learning technique, i.e., logistic regression, in MATLAB. We use the node number, the precision, and the density class as features. We label each result as 1 if the two-state Markov chain approach is by k per cent faster than the Skart method and as 0 otherwise. We plot the *receiver operating characteristic* (ROC) curve, which is commonly used to illustrate the performance of a binary classifier against varying discrimination threshold and we give the computed area under the curve (AUC) for different k in Figure 2a. When $k \geqslant 15$, the AUC value is over 0.7, which means that the prediction is very good. In another word, for a given PBN and precision requirement, we are able to predict whether the two-state Markov chain approach will be by 15 per cent faster than the Skart method in a very high accuracy rate.

We show in Table 4 (left part) the regression coefficient estimates of the three features. Clearly, the precision plays an important role in the prediction since the absolute value is always the largest. We further analyse how the performance of the two methods change with precision and show in Table 5 the percentage of cases that the two-state Markov chain approach is faster than the Skart method with respect to different precisions. *The two-*

TABLE 3: Performance comparison of the Skart and the two-state MC methods.

k	0	5	10	15	20	25	30
$t_{TS} \leq t_{Skart}$	69.03%	54.04%	40.06%	30.19%	25.24%	22.22%	20.18%
$t_{Skart} \leq t_{TS}$	30.97%	19.32%	11.98%	8.27%	6.42%	5.39%	4.83%

TABLE 4: Logistic regression coefficient estimates for performance prediction.

k	$t_{TS} \leq t_{Skart}$						$t_{Skart} \leq t_{TS}$					
	0	5	10	15	20	25	0	5	10	15	20	25
node number	-0.17	-0.15	-0.09	-0.01	0.04	0.09	0.17	0.20	0.21	0.27	0.25	0.28
precision	0.34	0.49	0.68	0.84	0.93	0.92	-0.34	-0.09	0.16	0.38	0.45	0.48
density	-0.15	-0.19	-0.29	-0.41	-0.52	-0.53	0.15	0.11	-0.04	-0.15	-0.27	-0.37

TABLE 5: Performance of the two methods with respect to different precisions.

precision	10^{-2}	5×10^{-3}	10^{-3}	5×10^{-4}	10^{-4}	5×10^{-5}	10^{-5}	5×10^{-6}	10^{-6}
$t_{TS} \leq t_{Skart}$	84%	76%	67%	64%	65%	59%	73%	75%	85%
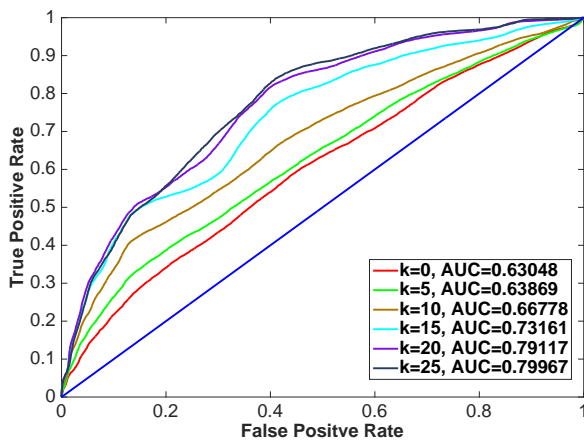
(a) ROC ($t_{TS} \leq t_{Skart}$).
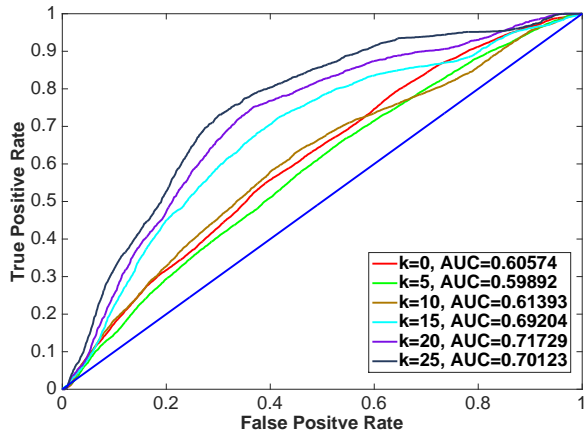
(b) ROC ($t_{Skart} \leq t_{TS}$).

Fig. 2: Prediction on the performance of the the Skart and the two-state MC methods.

state Markov chain approach has a larger chance to be faster than the Skart method in all the studied precisions, especially when the precision r is low (e.g., 10^{-2}) or very high (e.g., equal to or less than 10^{-5}). Notably, the chance that the two-state Markov chain approach is faster than the Skart method becomes very large when the precision is very high. This is due to the fact that the

Skart method requires a large memory to keep track of the large size trajectory when the precision is high. The CPU performance drops when operating on a large memory; on the other hand, the Skart method may run out of memory.

Moreover, we analyse the situation when the Skart method is by k per cent faster than the two-state Markov chain approach. This time it becomes difficult to make an accurate prediction as the largest AUC is only about 0.72 for $k = 20$ (see Figure 2b). Besides, the coefficient estimates in the right part of Table 4 also vary a lot with k, and precision is not always the dominating factor. The detailed experiment data can be obtained at http://satoss.uni.lu/software/ASSA-PBN/benchmark/benchmark.xlsx.

From the above analysis, we conclude that the two-state Markov chain approach outperforms the Skart method (state-of-the-art) in analysing large PBNs, especially for computing steady-state probabilities with very high precision.

5 A BIOLOGICAL CASE STUDY

In [18], a large-scale Boolean network of apoptosis (see Figure 3) in hepatocytes was introduced, where the assigned Boolean interactions for each molecule were derived from literature study. In [17], the original multi-value Boolean model was cast into the PBN framework: a binary PBN model, so-called 'extended apoptosis model' which comprised 91 nodes (state-space of size 2^{91}) and 102 interactions was constructed. In this extended version the possibility of activation of NF-κB through Caspase 8 (C8*), as described in [17], was included. The model was fitted to steady-state experimental data obtained in response to six different stimulations of the input nodes, see [17] for details.

As can be seen from the wiring of the network, the activation of complex2 (co2) by RIP-deubi can take place in two ways: 1) by a positive feedback loop from activated C8* and P $\rightarrow$ tBid $\rightarrow$ Bax $\rightarrow$ smac $\rightarrow$ RIP-deubi $\rightarrow$ co2 $\rightarrow$ C8*-co2 $\rightarrow$ C8*, and 2) by the positive signal from UV-B irradiation (input nodes UV(1) or UV(2)) $\rightarrow$ Bax $\rightarrow$ smac $\rightarrow$ RIP-deubi $\rightarrow$ co2. The former to be active requires the stimulation of the type 2 receptor (T2R). The latter way requires complex1 (co1) to be active, which cannot happen without the stimulation of the TNF receptor-1. Therefore, RIP-deubi can activate co2 only in the condition of co-stimulation by TNF and either UV(1) or UV(2). In consequence, it was suggested in [17] that the interaction of activation of co2 via RIP-deubi is not relevant and could be removed from the model in the context of modelling primary hepatocyte. However, due
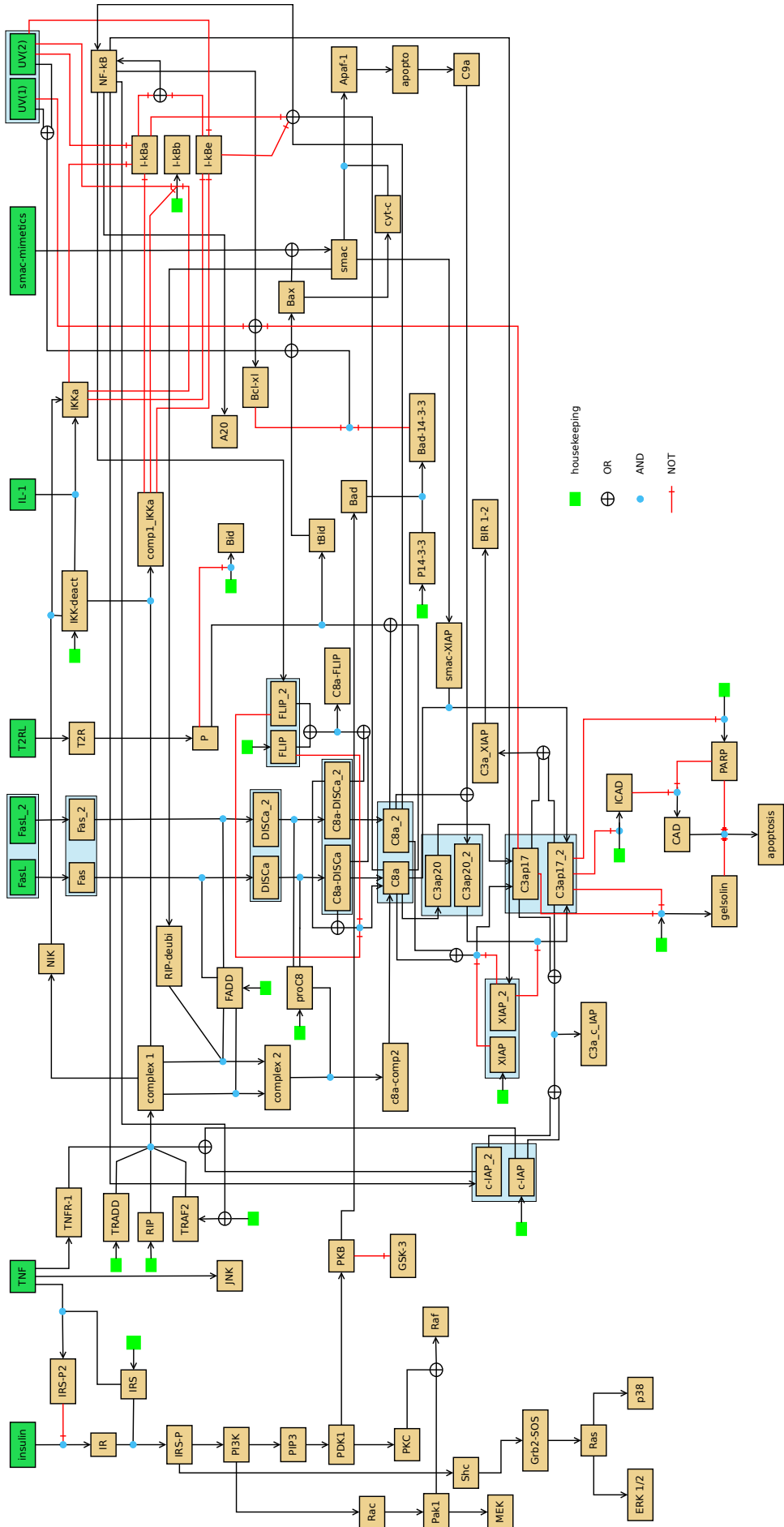
Fig. 3: The wiring of the Boolean model of apoptosis.

to the problem with efficient generation of very long trajectories in optPBN toolbox, quantitative analysis was hindered and this hypothesis could not be verified ([17]).

In this work, we take up this challenge and we quantitatively investigate the relevancy of the interaction of activation of co2 via RIP-deubi. We perform an extensive analysis in the context of co-stimulation by TNF and either UV(1) or UV(2): we compute long-term influences of parent nodes on the co2 node and the long-run sensitivities with respect to various perturbations related to specific predictor functions and their selection probabilities. For this purpose we apply the two-state Markov chain approach as implemented in our ASSA-PBN tool [15] to compute the relevant steady-state probabilities for the best-fit models described in [17]. Due to the efficient implementation, the ASSA-PBN tool can easily deal with trajectories of length exceeding 2×10^9 for this case study.

We consider 20 distinct parameter sets of [17] that resulted in the best fit of the 'extended apoptosis model' to the steady-state experimental data in six different stimulation conditions. In [17], parameter estimation was performed with steady-state measurements for the nodes apoptosis, C3ap17 or C3ap17_2 depending on the stimulation condition considered, and NF-κB. The optimisation procedure used was Particle Swarm and fit score function considered was the sum of squared errors of prediction (SSE) and the sum was taken over the three nodes in the six stimulation conditions. We took all the optimisation results from the three independent parameter estimation runs of [17], each containing 7500 parameter sets. We sorted them increasingly with respect to the cost function value obtained during optimisation, removed duplicates, and finally took the first 20 best-fit parameter sets.

As mentioned above, we fix the experimental context to co-stimulation of TNF and either UV(1) or UV(2). We note that originally in [18] UV-B irradiation conditions were imposed via a multi-value input node UV which could take on three values, i.e., 0 (no irradiation), 1 ($300 J/m^2$ UV-B irradiation), and 2 ($600 J/m^2$ UV-B irradiation). In the model of [17], UV input node was refined as UV(1) and UV(2) in order to cast the original model into the binary PBN framework. Therefore, we consider in our study two cases: 1) co-stimulation of TNF and UV(1) and 2) co-stimulation of TNF and UV(2). Node co2 has two independent predictor functions: co2 = co1 $\wedge$ FADD or co2 = co1 $\wedge$ FADD $\wedge$ RIP-deubi. The selection probabilities are denoted as $c_1^{(co2)}$ and $c_2^{(co2)}$, respectively. Their values have been optimised in [17].

We start with computing the influences with respect to the steady-state distribution, i.e., the long-term influences on co2 of each of its parent nodes: RIP-deubi, co1, and FADD, in accordance with the definition in Section 2.2. Notice that the computation of the three influences requires several joint steady-state probabilities to be estimated with the two-state Markov chain approach, e.g., (co1=1,FADD=1,RIP-deubi=0) or (co1=1,FADD=0). Each probability determines a specific split of the original Markov chain. For example, in the case of the estimation of the joint steady-state probability for (co1=1,FADD=0), the states of the underlying Markov chain of the apoptosis PBN model in which co1=1 and FADD=0 constitute meta state 1 and all the remaining states form meta state 0. Therefore, the estimation of influences is computationally demanding. The summarised results for the 20 parameter sets are presented for the co-stimulation of TNF and UV(1) or TNF and UV(2) in Table 6. They are consistent across the different parameter sets and clearly indicate that the

influence of RIP-deubi on co2 is small compared to the influence of co1 or FADD on co2. However, the influence of RIP-deubi is not negligible.

We take the analysis of the importance of the interaction between RIP-deubi and co2 further and we compute various long-run sensitivities with respect to selection probability perturbation. In particular, we perturb the selection probability $c_2^{(co2)}$ by $\pm 5\%$, i.e., we set the new value by multiplying the original value by (1 ± 0.05), and compute in line with Definition 2.1 how the joint steady-state distribution for (apoptosis,C3ap17,NFκB) differs from the non-perturbed one with respect to the l_1 norm, i.e., $|| \cdot ||_1$. We notice that the computation of the full steady-state distribution for the considered PBN model of apoptosis is practically intractable, i.e., it would require the estimation of 2^{91} values. Therefore, we restrict the computations to the estimation of eight joint steady-state probabilities for all possible combinations of values for (apoptosis,C3ap17,NFκB), i.e., the experimentally measured nodes. Each estimation is obtained by a separate run of the two-state Markov chain approach with the split into meta states determined by the considered probability as explained above in the case of the computation of long-term influences. To compare the estimated distributions we choose the l_1 norm after [19], where it is used in the computations of similar types of sensitivities for PBNs to these defined in Section 2.2. Notice that the l_1 norm of the difference of two probability distributions on a finite sample space is twice the *total variation distance*. The latter is a well-established metric for measuring the distance between probability distributions defined as the maximum difference between the probabilities assigned to a single event by the two distributions (see, e.g., [20]). Additionally, we check the difference when $c_2^{(co2)}$ is set to 0 (and, in consequence, $c_1^{(co2)}$ is set to 1). The obtained results for the 20 parameter sets in the conditions of co-stimulation of TNF and UV(1) and co-stimulation of TNF and UV(2) are summarised in Table 7. In all these cases, the sensitivities are very small. Therefore, the system turns to be insensitive to small perturbations of the value of $c_2^{(co2)}$. Also the complete removal of the second predictor function for co2 does not cause any drastic changes in the joint steady-state distribution for (apoptosis,C3ap17,NF-κB).

Finally, we compute the long-run sensitivity with respect to permanent on/off perturbations of the node RIP-deubi in accordance with Definition 2.2. As before, we consider the joint steady-state distributions for (apoptosis,C3ap17,NF-κB) and we choose the l_1-norm. The results, given in Table 8, show that in both variants of UV-B irradiation the sensitivities are not negligible and the permanent on/off perturbations of RIP-deubi have impact on the steady-state distribution.

To conclude, all the obtained results indicate that in the context of co-stimulation of TNF and either UV(1) or UV(2) the interaction between RIP-deubi and co2 plays a certain role. Although the elimination of the interaction does not invoke significant changes to the considered joint steady-state distribution, the long-term influence of RIP-deubi on co2 is not negligible and may be important for other nodes in the network.

6 DISCUSSION AND CONCLUSION

Most current tools for statistical model checking, a simulation-based approach using hypothesis testing to infer whether a stochastic system satisfies a property, are restricted for bounded properties which can be checked on finite executions of the system. Recently, both the Skart method [21] and the perfect

TABLE 6: Long-term influences of RIP-duebi, co1, and FADD on co2 in the 'extended apoptosis model' in [17] under the co-stimulation of both TNF and UV(1) or UV(2).

	TNF and UV(1)			TNF and UV(2)		
	$I_{\text{RIP-deubi}}$	I_{co1}	I_{FADD}	$I_{\text{RIP-deubi}}$	I_{co1}	I_{FADD}
Best fit	0.2614	0.9981	0.9935	0.2615	0.9980	0.9936
Min	0.0000	0.9979	0.9935	0.0000	0.9979	0.9936
Max	0.3145	0.9988	0.9944	0.3146	0.9990	0.9947
Mean	0.2087	0.9982	0.9937	0.2088	0.9982	0.9938
Std	0.0735	0.0002	0.0002	0.0735	0.0002	0.0003

TABLE 7: Long-run sensitivities w.r.t selection probability perturbations.

	TNF and UV(1)			TNF and UV(2)		
$c_2^{(co2)}$	$+5\%$	-5%	$=0$	$+5\%$	-5%	$=0$
Best fit	0.0003	0.0002	0.0011	0.0002	0.0004	0.0011
Min	0.0002	0.0002	0.0003	0.0002	0.0002	0.0002
Max	0.0008	0.0008	0.0014	0.0012	0.0007	0.0013
Mean	0.0005	0.0005	0.0009	0.0004	0.0004	0.0009
Std	0.0001	0.0001	0.0003	0.0002	0.0001	0.0003

TABLE 8: Long-run sensitivities w.r.t permanent on/off perturbations of RIP-deubi.

RIP-deubi f. pert.	Best fit	Min	Max	Mean	Std
TNF & UV(1)	0.3075	0.0130	0.3595	0.2089	0.0823
TNF & UV(2)	0.3097	0.0105	0.3612	0.2105	0.0827

simulation algorithm [22] have been explored for statistical model checking of steady state and unbounded until properties. The perfect simulation algorithm for sampling the steady-state of an ergodic DTMC is based on the indigenous idea of the *backward coupling scheme* [11]. It allows to draw independent samples which are distributed exactly in accordance with the steady-state distribution of a DTMC. However, due to the nature of this method, each state in the state space needs to be considered at each step of the coupling scheme. If a DTMC is monotone, then it is possible to sample from the steady-state distribution by considering the maximal and minimal states only [11]. This was exploited in [22] for model checking large queuing networks. Unfortunately, it is not applicable to PBNs with perturbations. In consequence, the perfect simulation algorithm is only suited for at most medium-size PBNs and large-size PBNs are out of its scope. Thus, we have only compared the performance of the two-state Markov chain approach with the Skart method.

Moreover, in this study we have identified a problem of generating biased results by the original two-state Markov chain approach and have proposed three heuristics to avoid wrong initialisation. Finally, we demonstrated the potential of the two-state Markov chain approach on a study of a large, 91-node PBN model of apoptosis in hepatocytes. The two-state Markov chain approach facilitated the quantitative analysis of the large network and the investigation of a previously formulated hypothesis regarding the relevance of the interaction of activation of co2 via RIP-deubi. In the future, we aim to investigate the usage of the discussed statistical methods for approximate steady-state analysis in a research project on systems biology, where we will apply them to develop new techniques for minimal structural interventions to alter steady-state probabilities for large regulatory networks.

APPENDIX A
DERIVATION OF THE NUMBER OF "BURN-IN" ITERATIONS

Let $\{Z_t\}_{t \geq 0}$ be a discrete-time two-state Markov chain as given in Figure 1b. Z_t has the value 0 or 1 if the system is in state 0 or state 1 at time n, respectively. The transition probabilities satisfy $0 < \alpha, \beta < 1$ and the transition matrix for this chain has the following form

$$P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}.$$

Matrix P has two distinct eigenvalues: 1 and $\lambda = (1 - \alpha - \beta)$. Notice that $|\lambda| < 1$.

The chain is ergodic and the unique steady-state distribution is $\pi = [\pi_0 \ \pi_1] = [\frac{\beta}{\alpha+\beta} \ \frac{\alpha}{\alpha+\beta}]$. Let $\mathbb{E}_\pi(Z_t)$ denote the expected value of Z_t for any fixed $t \geq 0$, with respect to the steady-state distribution π. We have that $\mathbb{E}_\pi(Z_t) = \frac{\alpha}{\alpha+\beta}$.

The m-step transition matrix can be written, as can be checked by induction, in the form

$$P^m = \begin{bmatrix} \pi_0 & \pi_1 \\ \pi_0 & \pi_1 \end{bmatrix} + \frac{\lambda^m}{\alpha+\beta} \cdot \begin{bmatrix} \alpha & -\alpha \\ -\beta & \beta \end{bmatrix},$$

where λ is the second eigenvalue of P.

Suppose we require m to be such that the following condition is satisfied

$$\begin{bmatrix} |\mathbb{P}[Z_m = 0 \,|\, Z_0 = j] - \pi_0| \\ |\mathbb{P}[Z_m = 1 \,|\, Z_0 = j] - \pi_1| \end{bmatrix} < \begin{bmatrix} \varepsilon \\ \varepsilon \end{bmatrix} \quad (1)$$

for some $\varepsilon > 0$. If $e_0 = [1 \ 0]$ and $e_1 = [0 \ 1]$, then for $j \in \{0, 1\}$ we have that

$$\begin{bmatrix} \mathbb{P}[Z_m = 0 \,|\, Z_0 = j] \\ \mathbb{P}[Z_m = 1 \,|\, Z_0 = j] \end{bmatrix} = (e_j P^m)^{\mathrm{T}} = (P^m)^{\mathrm{T}} (e_j)^{\mathrm{T}},$$

where T is the transposition operator. For any vector $v = [v_1 \; v_2 \; \ldots \; v_n]^{\mathrm{T}} \in \mathbb{R}^n$ we use $|v|$ to denote $[|v_1| \; |v_2| \; \ldots \; |v_n|]^{\mathrm{T}}$. Therefore, condition (1) can be rewritten as

$$\left| (P^m)^{\mathrm{T}} (e_j)^{\mathrm{T}} - \begin{bmatrix} \pi_0 \\ \pi_1 \end{bmatrix} \right| < \begin{bmatrix} \varepsilon \\ \varepsilon \end{bmatrix}.$$

For $j = 0$ and $j = 1$ the above simplifies to

$$\left| \frac{\lambda^m}{\alpha + \beta} \cdot \begin{bmatrix} \alpha \\ -\alpha \end{bmatrix} \right| < \begin{bmatrix} \varepsilon \\ \varepsilon \end{bmatrix} \quad \text{and} \quad \left| \frac{\lambda^m}{\alpha + \beta} \cdot \begin{bmatrix} -\beta \\ \beta \end{bmatrix} \right| < \begin{bmatrix} \varepsilon \\ \varepsilon \end{bmatrix},$$

respectively. Therefore, it is enough to consider the following two inequalities

$$\left| \frac{\lambda^m \alpha}{\alpha + \beta} \right| < \varepsilon \quad \text{and} \quad \left| \frac{\lambda^m \beta}{\alpha + \beta} \right| < \varepsilon,$$

which, since $\alpha, \beta > 0$, can be rewritten as

$$|\lambda^m| < \frac{\varepsilon(\alpha + \beta)}{\alpha} \quad \text{and} \quad |\lambda^m| < \frac{\varepsilon(\alpha + \beta)}{\beta}.$$

Equivalently, m has to satisfy

$$|\lambda^m| < \frac{\varepsilon(\alpha + \beta)}{\max(\alpha, \beta)}.$$

By the fact that $|\lambda^m| = |\lambda|^m$ this can be expressed as

$$|\lambda|^m < \frac{\varepsilon(\alpha + \beta)}{\max(\alpha, \beta)}.$$

Then, by taking the logarithm to base 10 on both sides[1], we have that

$$m \cdot \log(|\lambda|) < \log\left(\frac{\varepsilon(\alpha + \beta)}{\max(\alpha, \beta)} \right)$$

and in consequence, since $|\lambda| < 1$ and $\log|\lambda| < 0$,

$$m > \frac{\log\left(\frac{\varepsilon(\alpha + \beta)}{\max(\alpha, \beta)} \right)}{\log(|\lambda|)}.$$

APPENDIX B
DERIVATION OF THE SAMPLE SIZE

By the Law of Large Numbers for irreducible positive recurrent Markov chains $\overline{Z}_n \to \pi_1$ $a.\,s.$ with $n \to \infty$, where $\overline{Z}_n = \frac{1}{n} \sum_{t=1}^n Z_t$. Now, by a variant of the Central Limit Theorem for non-independent random variables[2], for n large, $\overline{Z}_n$ is approximately normally distributed with mean $\pi_1 = \frac{\alpha}{\alpha + \beta}$ and asymptotic variance $\sigma_{\mathrm{as}}^2 = \frac{1}{n} \frac{\alpha\beta(2 - \alpha - \beta)}{(\alpha + \beta)^3}$, see Section C for the derivation of the asymptotic variance. Let X be the standardised $\overline{Z}_n$, i.e.,

$$X = \frac{\overline{Z}_n - \pi_1}{\sigma_{\mathrm{as}}/\sqrt{n}}.$$

If follows that X is normally distributed with mean 0 and variance 1, i.e., $X \sim N(0,1)$.

Now, we require n to be such that the condition $\mathbb{P}[\pi_1 - r \leq \overline{Z}_n \leq \pi_1 + r] = s$ is satisfied for some specified r and s. This condition can be rewritten as

$$\mathbb{P}[-r \leq \overline{Z}_n - \pi_1 \leq r] = s,$$

---

1. In fact, by the formula for change of base for logarithms, the natural logarithm (ln), the logarithm to base 2 ($\log_2$), or a logarithm to any other base could be used to calculate m instead of log. Notice that m does **not** depend on the choice of the base of the logarithm!

2. Notice that the random variables Z_t, Z_{t+1} which values are consecutive states of a trajectory are correlated and are not independent.

and further as

$$\mathbb{P}\left[-r \cdot \frac{\sqrt{n}}{\sigma_{\mathrm{as}}} \leq \frac{\overline{Z}_n - \pi_1}{\sigma_{\mathrm{as}}/\sqrt{n}} \leq r \cdot \frac{\sqrt{n}}{\sigma_{\mathrm{as}}} \right] = s,$$

which is

$$\mathbb{P}\left[-r \cdot \frac{\sqrt{n}}{\sigma_{\mathrm{as}}} \leq X \leq r \cdot \frac{\sqrt{n}}{\sigma_{\mathrm{as}}} \right] = s.$$

Since $X \sim N(0,1)$ and $N(0,1)$ is symmetric around 0, it follows that

$$\mathbb{P}\left[0 \leq X \leq r \cdot \frac{\sqrt{n}}{\sigma_{\mathrm{as}}} \right] = \frac{s}{2}$$

and

$$\mathbb{P}\left[X \leq r \cdot \frac{\sqrt{n}}{\sigma_{\mathrm{as}}} \right] = \frac{1}{2} + \frac{s}{2} = \frac{1}{2}(1 + s).$$

Let $\Phi(\cdot)$ be the standard normal cumulative distribution function. Then the above can be rewritten as

$$\Phi\left(r \cdot \frac{\sqrt{n}}{\sigma_{\mathrm{as}}} \right) = \frac{1}{2}(1 + s).$$

Therefore, if we denote the inverse of the standard normal cumulative distribution function with $\Phi^{-1}(\cdot)$, we have that

$$r \cdot \frac{\sqrt{n}}{\sigma_{\mathrm{as}}} = \Phi^{-1}\left(\frac{1}{2}(1 + s) \right).$$

In consequence,

$$n = \frac{\sigma_{\mathrm{as}}^2}{\left\{ \frac{r}{\Phi^{-1}(\frac{1}{2}(1+s))} \right\}^2} = \frac{\frac{\alpha\beta(2 - \alpha - \beta)}{(\alpha + \beta)^3}}{\left\{ \frac{r}{\Phi^{-1}(\frac{1}{2}(1+s))} \right\}^2}.$$

APPENDIX C
DERIVATION OF THE ASYMPTOTIC VARIANCE

By the Central Limit Theorem for stationary stochastic processes[3] $\sqrt{n}(\overline{Z}_n - \pi_1) \xrightarrow{d} N(0, \sigma_{\mathrm{as}}^2)$ as $n \to \infty$, where σ_{as}^2 is the so-called asymptotic variance given by

$$\sigma_{\mathrm{as}}^2 = \mathrm{Var}_\pi(Z_j) + 2 \sum_{k=1}^\infty \mathrm{Cov}_\pi(Z_j, Z_{j+k}) \tag{2}$$

and $\mathrm{Var}_\pi(\cdot)$ and $\mathrm{Cov}_\pi(\cdot)$ denote the variance and covariance with respect to the steady-state distribution π, respectively. We proceed to calculate σ_{as}^2. First, observe that $\mathbb{E}_\pi(Z_n Z_{n+1}) = \frac{\alpha}{\alpha + \beta}(1 - \beta)$: $Z_n Z_{n+1} \neq 0$ if and only if the chain is state 1 at time n and remains in 1 at time $n + 1$, i.e., $Z_n = Z_{n+1} = 1$. The probability of this event at steady state is $\frac{\alpha}{\alpha + \beta}(1 - \beta)$. Then, by the definition of covariance, we have that the steady-state covariance between consecutive random variables of the two-state Markov chain, i.e., $\mathrm{Cov}_\pi(Z_n, Z_{n+1})$ is

$$\mathrm{Cov}_\pi(Z_n, Z_{n+1}) = \mathbb{E}_\pi\left[(Z_n - \mathbb{E}_\pi(Z_n))(Z_{n+1} - \mathbb{E}_\pi(Z_{n+1})) \right]$$

$$= \mathbb{E}_\pi\left[\left(Z_n - \frac{\alpha}{\alpha + \beta} \right)\left(Z_{n+1} - \frac{\alpha}{\alpha + \beta} \right) \right]$$

$$= \mathbb{E}_\pi\left[Z_n Z_{n+1} - \frac{\alpha}{\alpha + \beta}(Z_n + Z_{n+1}) + \frac{\alpha^2}{(\alpha + \beta)^2} \right]$$

$$= \mathbb{E}_\pi(Z_n Z_{n+1}) - \frac{\alpha}{\alpha + \beta}(\mathbb{E}_\pi(Z_n) + \mathbb{E}_\pi(Z_{n+1})) + \frac{\alpha^2}{(\alpha + \beta)^2}$$

$$= \frac{\alpha(1 - \beta)}{\alpha + \beta} - 2\frac{\alpha^2}{(\alpha + \beta)^2} + \frac{\alpha^2}{(\alpha + \beta)^2}$$

$$= \frac{\alpha\beta(1 - \alpha - \beta)}{(\alpha + \beta)^2}.$$

---

3. After discarding the 'burn-in' part of the trajectory, we can assume that the Markov chain in a stationary stochastic process.

Further, we have that $\text{Var}_\pi(Z_n) = \pi_0 \cdot \pi_1 = \frac{\alpha\beta}{(\alpha+\beta)^2}$ (variance of the Bernoulli distribution) and it can be shown that $\text{Cov}_\pi(Z_n, Z_{n+k}) = (1-\alpha-\beta)^k \cdot \text{Var}_\pi(Z_n)$ for any $k \geq 1$. Now, according to Equation (2), we have

$$
\begin{aligned}
\sigma_{\text{as}}^2 &= \text{Var}_\pi(X_j) + 2\sum_{k=1}^{\infty} \text{Cov}_\pi(X_j, X_{j+k}) \\
&= \frac{\alpha\beta}{(\alpha+\beta)^2} + 2\sum_{k=1}^{\infty}(1-\alpha-\beta)^k \cdot \frac{\alpha\beta}{(\alpha+\beta)^2} \\
&= \frac{\alpha\beta}{(\alpha+\beta)^2} + \frac{2\alpha\beta}{(\alpha+\beta)^2} \cdot \sum_{k=1}^{\infty}(1-\alpha-\beta)^k \\
&= \frac{\alpha\beta}{(\alpha+\beta)^2} + \frac{2\alpha\beta}{(\alpha+\beta)^2} \cdot \frac{1-\alpha-\beta}{\alpha+\beta} \\
&= \frac{\alpha\beta(2-\alpha-\beta)}{(\alpha+\beta)^3}.
\end{aligned}
$$

In consequence, $\overline{Z}_n$ is approximately normally distributed with mean $\frac{\alpha}{\alpha+\beta}$ and variance $\frac{1}{n}\frac{\alpha\beta(2-\alpha-\beta)}{(\alpha+\beta)^3}$.

APPENDIX D
'PITFALL AVOIDANCE' HEURISTIC METHOD: FORMULA DERIVATIONS

We start with analysing the minimum values $n(\cdot, \cdot)$ can attain. The function is considered on the domain $D = (0,1] \times (0,1]$ and, as mentioned before, the estimated values of α and β are within the range $[\frac{1}{n_0}, 1]$. Computing the partial derivatives, equating them to zero, and solving for α and β yields $\alpha = -\beta$, which has no solution in the considered domain. Hence, the function has neither local minimum nor maximum on D. Let us fix β for a moment and consider $n(\alpha, \beta)$ as a function of α. We denote it as $n_\beta(\alpha)$. By differentiating with respect to α, we obtain

$$
\frac{\partial}{\partial\alpha}n_\beta(\alpha) = \frac{1}{c_{r,s}}\frac{\beta\left(\alpha^2 - \beta^2 - 4\alpha + 2\beta\right)}{(\alpha+\beta)^4},
$$

where

$$
c_{r,s} = \frac{r^2}{\left(\Phi^{-1}\left(\frac{1}{2}(1+s)\right)\right)^2}.
$$

By equating to zero and solving for α we get two solutions: $\alpha_1 = 2 - \sqrt{\beta^2 - 2\beta + 4}$ and $\alpha_2 = 2 + \sqrt{\beta^2 - 2\beta + 4}$. Since the second solution is always greater than 1 on the $(0,1]$ interval, only the first solution is valid. The sign of the second derivative of $n_\beta(\alpha)$ with respect to α at α_1 is negative. This shows that for any fixed β, $n_\beta(\alpha)$ grows on the interval $[\frac{1}{n_0}, \alpha_1]$, attains its maximum at α_1 and decreases on the interval $[\alpha_1, 1]$. Notice that n is symmetric, i.e., $n(\alpha, \beta) = n(\beta, \alpha)$. Thus the minimum value n could attain for α and β estimated from a sample of size n_0 is given by $\min\left(n\left(\frac{1}{n_0}, \frac{1}{n_0}\right), n\left(\frac{1}{n_0}, 1\right)\right)$. After evaluating n we get

$$
n\left(\frac{1}{n_0}, \frac{1}{n_0}\right) = \frac{n_0 - 1}{4c_{r,s}}
$$

and

$$
n\left(\frac{1}{n_0}, 1\right) = \frac{(n_0 - 1) \cdot n_0}{c_{r,s} \cdot (1 + n_0)^3}.
$$

Now, to avoid the situation where the initial estimates of α and β lead to $n(\alpha, \beta) < 2n_0$, it is enough to make sure that given r and s

the following condition is satisfied: $\min(n(\frac{1}{n_0}, \frac{1}{n_0}), n(\frac{1}{n_0}, 1)) \geqslant 2n_0$. This can be rewritten as

$$
\begin{cases}
(8c_{r,s} - 1)n_0 + 1 \leq 0 \\
2c_{r,s}n_0^3 + 6c_{r,s}n_0^2 + (6c_{r,s} - 1)n_0 + 2c_{r,s} + 1 \leq 0
\end{cases}
$$

Both inequalities can be solved analytically. Given that $n_0 > 0$, the solution of the first inequality is

$$
\begin{cases}
n_0 \in \left[-\frac{1}{8 \cdot c_{r,s} - 1}, \infty\right) & c_{r,s} < \frac{1}{8} \\
n_0 \in \emptyset & c_{r,s} \geqslant \frac{1}{8}.
\end{cases} \tag{3}
$$

The solution of the second inequality is more complicated, but can be easily obtained with computer algebra system software (e.g., Maple™). In Table 1 we present some solutions for a number of values for r and s.

ACKNOWLEDGMENTS

REFERENCES

[1] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[2] I. Shmulevich and E. R. Dougherty, *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks*. SIAM Press, 2010.

[3] P. Trairatphisan, A. Mizera, J. Pang, A.-A. Tantar, J. Schneider, and T. Sauter, "Recent development and biomedical applications of probabilistic Boolean networks," *Cell Communication and Signaling*, vol. 11, p. 46, 2013.

[4] R. Albert and H. G. Othmer, "The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster," *Journal of Theoretical Biology*, vol. 223, no. 1, pp. 1–18, 2003.

[5] S. Bornholdt, "Less is more in modeling large genetic networks," *Science*, vol. 310, no. 5747, pp. 449–451, 2005.

[6] S. Huang, "Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation," *Pharmacogenomics*, vol. 2, no. 3, pp. 203–222, 2001.

[7] I. Shmulevich, I. Gluhovsky, R. F. Hashimoto, E. R. Dougherty, and W. Zhang, "Steady-state analysis of genetic regulatory networks modelled by probabilistic Boolean networks," *Comparative and Functional Genomics*, vol. 4, no. 6, pp. 601–608, 2003.

[8] M. K. Cowles and B. P. Carlin, "Markov chain Monte Carlo convergence diagnostics: A comparative review," *Journal of American Statistical Association*, vol. 91, no. 434, pp. 883–904, 1996.

[9] A. Tafazzoli, J. R. Wilson, E. K. Lada, and N. M. Steiger, "Skart: A skewness- and autoregression-adjusted batch-means procedure for simulation analysis," in *Proc. 2008 Winter Simulation Conference*, 2008, pp. 387–395.

[10] J. R. Norris, *Markov Chains*. Cambridge University Press, 1998.

[11] J. Propp and D. Wilson, "Exact sampling with coupled Markov chains and applications to statistical mechanics," *Random Structures & Algorithms*, vol. 9, no. 1, pp. 223–252, 1996.

[12] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[13] A. E. Raftery and S. Lewis, "How many iterations in the Gibbs sampler?" *Bayesian Statistics*, vol. 4, pp. 763–773, 1992.

[14] A. Mizera, J. Pang, and Q. Yuan, "ASSA-PBN: An approximate steady-state analyser for probabilistic Boolean networks," in *Proc. 13th International Symposium on Automated Technology for Verification and Analysis*, ser. LNCS, vol. 9364. Springer, 2015, pp. 214–220.

[15] ——, "ASSA-PBN 2.0: A software tool for probabilistic Boolean networks," in *Proc. 14th International Conference on Computational Methods in Systems Biology*, ser. LNCS, vol. 9859. Springer, 2016, pp. 309–315.

[16] A. Walker, "An efficient method for generating discrete random variables with general distributions." *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 253–256, 1977.

[17] P. Trairatphisan, A. Mizera, J. Pang, A.-A. Tantar, and T. Sauter, "optPBN: An optimisation toolbox for probabilistic boolean networks," *PLOS ONE*, vol. 9, no. 7, p. e98001, 2014.

[18] R. Schlatter, K. Schmich, I. A. Vizcarra, P. Scheurich, T. Sauter, C. Borner, M. Ederer, I. Merfort, and O. Sawodny, "ON/OFF and beyond - a Boolean model of apoptosis," *PLOS Computational Biology*, vol. 5, no. 12, p. e1000595, 2009.

[19] X. Qian and E. R. Dougherty, "On the long-run sensitivity of probabilistic Boolean networks," *Journal of Theoretical Biology*, vol. 257, no. 4, pp. 560–577, 2009.

[20] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[21] C. Rohr, "Simulative model checking of steady state and time-unbounded temporal operators," *Transactions on Petri Nets and Other Models of Concurrency*, vol. 8, pp. 142–158, 2013.

[22] D. El Rabih and N. Pekergin, "Statistical model checking using perfect simulation," in *Proc. 7th Symposium on Automated Technology for Verification and Analysis*, ser. LNCS, vol. 5799. Springer, 2009, pp. 120–134.

[23] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos, "Management of an academic HPC cluster: The UL experience," in *Proc. 12th International Conference on High Performance Computing & Simulation*. IEEE CS, 2014, pp. 959–967.

Dr. Andrzej Mizera received the MSc degree in Computer Science from the University of Warsaw, Poland in 2005. He then obtained his PhD in Computer Science with minor in mathematics in the area of Computational Systems Biology from the Åbo Akademi University, Turku, Finland in 2011. His research interests are related to computational and mathematical modelling of biological systems. He is currently holding a post-doc position at the University of Luxembourg.
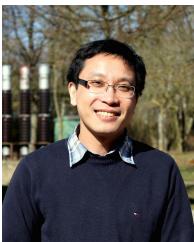
Dr. Jun Pang received his PhD in Computer Science from Vrije Universiteit Amsterdam, The Netherlands in 2004. Currently, he is a senior researcher in the Security and Trust of Software Systems research group at the University of Luxembourg. His research interests include formal methods, security and privacy, big data analytics, and computational systems biology.

Qixia Yuan received his MSc degrees in computer science from both the University of Luxembourg and Shandong University in 2012. He is currently a PhD student at the Computer Science and Communications Research Unit at the University of Luxembourg. His research interests focus on development and application of formal methods in systems biology.