Reviving the Two-state Markov Chain Approach
(Technical Report)

Andrzej Mizera¹, Jun Pang¹,², and Qixia Yuan¹⋆

¹ University of Luxembourg, FSTC, Luxembourg
² University of Luxembourg, SnT, Luxembourg
firstname.lastname@uni.lu

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 fact, statistical methods for steady-state approximation are the only viable means when dealing with large networks. In this paper, we revive the two-state Markov chain approach presented in the literature. We first identify a problem of generating biased results, due to the size of the initial sample with which the approach needs to start and we propose a few heuristics to avoid such a pitfall. Second, we conduct an extensive experimental comparison of the two-state Markov chain approach and another approach based on the Skart method and we show that statistically the two-state Markov chain has a better performance. Finally, we apply this approach to a large PBN model of apoptosis in hepatocytes.

1 Introduction

Systems biology aims to study biological systems from a holistic perspective, with the goal to provide a comprehensive, system-level understanding of cellular behaviour. Proper functioning of a living cell requires a finely-tuned and orchestrated interplay of many complex processes. Complex interactions within a biological system lead to emergent properties which are crucial for sustaining life. Therefore, understanding the machinery of life requires the use of holistic approaches which enable the study of a system as a whole, in contrast to the reductionist approach. Computational modelling plays a prominent role in the field of systems biology. Construction and analysis of a computational model for some biological process enables the systematisation of available biological knowledge, identification of missing biological information, provides formal means for understanding and reasoning about the concerted interplay between different parts of the model, finally reveals directions for future experimental work which could provide data for better understanding of the process under study.

⋆ Supported by the National Research Fund, Luxembourg (grant 7814267) and partially by Google Summer of Code 2014.
Unfortunately, computational modelling of biological processes that take place in a living cell poses significant challenges with respect to the size of the state-space that needs to be considered. Modelling of certain parts of cellular machinery such as gene regulatory networks (GRNs) or signal transduction pathways often leads to dynamical models characterised by huge state-spaces of sizes that surpass the sizes of any human-designed systems by orders of magnitude. Therefore, profound understanding of biological processes asks for the development of new methods and approaches that would provide means for formal analysis and reasoning about such huge systems.

In this study we concentrate on the analysis of the steady-state dynamics of biological processes, in particular GRNs, 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]. 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 [3]. 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. However, moving towards a higher-level description by ignoring the molecular details allows to grasp the system-level behaviour of the network [4]. In consequence, these coarse-grained formalisms are broadly applied in systems biology studies of systems where the predictions of exact reaction times are not of main interest. For example, this is the case in the study of dynamical attractors of a regulatory network, which seem to depend on the circuit wiring rather than kinetic constants (such as production rates or interaction rates) [5]. In this sense modelling biological systems with more abstract, high-level view formalisms has certain unquestionable advantages.

One of the key aspects in the analysis of such dynamic systems is the comprehension of their steady-state (long-run) behaviour. For example, attractors of such systems were hypothesised to characterise cellular phenotypes [6]. Another complementary conjecture is that attractors correspond to functional cellular states such as proliferation, apoptosis, or differentiation [7]. 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, considered collectively or individually, and that enable the identification of elements with highest impact. In this way they provide insight into the control mechanisms of the network.

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 [8]. However, due to the difficulties with the assessment of the convergence rate to the steady-state distribution (see, e.g., [9]), certain care is required when applying these methods in practice. A number of statistical methods exists, which allow to empirically determine when to stop the simulation and output estimates. We employ in our study: (1) the two-state Markov chain approach [10] and (2) the Skart batch-means method of [11]. The two-state Markov chain approach was introduced in 1992 by Raftery and Lewis; and Shmulevich et al. [8] 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 its usage for approximating steady-state probabilities of large PBNs, which often arise in computational systems biology as models of full-size genetic networks. We identify a problem concerned with the choice of the initial sample size in the two-state Markov chain approach. As we show, an unconscious choice may lead to biased results. We propose a few heuristics to avoid this problem. By extensive experiments, we show that the two-state Markov chain approach outperforms the Skart method in most cases, where the batch-means Skart method is considered the current state-of-the-art approach. In this way we show that the two-state Markov chain is often the optimal choice for the analysis of large PBN models of biological systems.

Structure of the paper. After presenting some preliminaries in Section 2, we describe the two-state Markov chain approach in Section 3 and identify a problem of generating biased results, due to the size of the initial sample with which the approach needs to start. We then propose some heuristics for the approach to avoid unfortunate initialisations. We perform an extensive evaluation and comparison of the two-state Markov chain approach and the Skart method in Section 4 on a large number of randomly generated PBNs. In most cases, the two-state Markov chain approach seems to have a better performance in terms of computational cost. Finally, we apply the two-state Markov chain approach to study a large PBN model of apoptosis in hepatocytes consisting of 91 nodes in Section 5. We compute the steady-state influences and long-run sensitivities and confirm previously formulated hypothesis. We conclude our paper with some discussions in Section 6.

2 Preliminaries

2.1 Finite discrete-time Markov chains (DTMCs)

Let $S$ be a finite set of states. A (first-order) discrete-time Markov chain 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, $P(X_{t+1} = s_{t+1} | X_t = s_t, X_{t-1} = s_{t-1}, \ldots, X_0 = s_0) = P(X_{t+1} = s_{t+1} | X_t = s_t)$ for all $s_{t+1}, s_t, \ldots, s_0 \in S$. Here, we consider time-homogenous Markov chains, i.e., chains where $P(X_{t+1} = s' | X_t = s)$, 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 \( P_{s,s'} \geq 0 \) and \( \sum_{s' \in S} P_{s,s'} = 1 \) for all \( s \in S \). We denote by \( \pi \) a probability distribution on \( S \). If \( \pi = \pi P \), then \( \pi \) is a stationary distribution of the DTMC (also referred to as the invariant distribution). A path of length \( n \) is a sequence \( s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_n \) 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 [12] an ergodic chain has a unique stationary distribution (also referred to as the steady-state distribution) given by \( \lim_{n \to \infty} \pi_0 P^n \), where \( \pi_0 \) is any initial probability distribution on \( S \). In consequence, the limiting distribution for an ergodic chain is independent of the choice of \( \pi_0 \). 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 \( X_t \) where \( \{X_t\}_{t \in \mathbb{N}} \) 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 when the transition function \( \phi(., u) \) is monotonic for all \( u \), then the Markov chain is said to be monotone ([13][14]).

### 2.2 Probabilistic Boolean networks (PBNs)

A PBN \( G(V,F) \) consists of a set of binary-valued nodes (also known as genes) \( V = \{v_1,v_2,\ldots,v_n\} \) and a list of sets \( F = \{F_1,F_2,\ldots,F_n\} \). 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 \( F_i \in F \): \( \epsilon_j^{(i)} \) is the probability of selecting \( f_j^{(i)} \in F_i \) as the next predictor for \( v_i \) and it holds that \( \sum_{j=1}^{l(i)} \epsilon_j^{(i)} = 1 \). 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 given 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 \( f_k = (f_{k_1}^{(1)},f_{k_2}^{(2)},\ldots,f_{k_n}^{(n)}) \in F_1 \times F_2 \times \cdots \times F_n \),
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 \( f(t) \). Due to independence, \( \mathbb{P}(f_k) = \mathbb{P}(f(t) = f_k) = \prod_{i=1}^{N} c_{k_i}^{(i)} \).

In PBNs with perturbations, a perturbation parameter \( p \in (0, 1) \) is introduced to sample the perturbation vector \( \gamma(t) = (\gamma_1(t), \gamma_2(t), \ldots, \gamma_n(t)) \), 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 \( s(t+1) = f(t)(s(t)) \) if \( \gamma(t) = 0 \) and as \( s(t+1) = s(t) \oplus \gamma(t) \) 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 [15]. The transition matrix is given by

\[
P_{s,s'} = (1-p)^n \sum_{k=1}^{N} \mathbb{1}_{f_k(s) = s'} \mathbb{P}(f_k) + (1-(1-p)^n)p^{\eta(s,s')} (1-p)^{n-\eta(s,s')},
\]

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 limit 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, which is often of interest for biological systems.

The density of a PBN is measured with its function number and parent nodes number. For a PBN \( G \), its density is defined as \( D(G) = \frac{n}{N_f} \sum_{i=1}^{N_f} \omega(i) \), 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 \( i \)th predictor function.

### 3 The Two-state Markov Chain Approach

#### 3.1 Description

We recall the two-state Markov chain approach originally introduced in [10]. The two-state Markov chain approach 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 \( q \), 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 \( \{Z_i\}_{t \geq 0} \) 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 \( t \). \( \{Z_i\}_{t \geq 0} \) is a binary (0-1) stochastic process, but in general it is not a Markov chain. However, as argued in [10], a reasonable assumption is that the dependency in \( \{Z_i\}_{t \geq 0} \) falls off rapidly with lag. Therefore, a new process \( \{Z_i^{(k)}\}_{t \geq 0} \), where \( Z_t^{(k)} = Z_{1+(t-1)k} \), will be approximately a first-order Markov chain for \( k \) large
The steady-state probability estimate $\hat{q}$ 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 $|P[Z_t^{(k)} = i | Z_0^{(k)} = j] - \pi_i| < \epsilon$ for a given $\epsilon > 0$ 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 $\hat{q}$ is required to satisfy $P[q - r \leq \hat{q} \leq q + r] \geq s$, 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 $\hat{q}$, 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 $N = 1 + \lceil m(\alpha, \beta) \rceil k$, where $t = \lceil m(\alpha, \beta) \rceil$. The functions $m$ and $n$ depend on the transitions probabilities $\alpha$ and $\beta$ and are given by

$$m(\alpha, \beta) = \frac{\log \left( \frac{\epsilon(\alpha + \beta)}{\max(\alpha, \beta)} \right)}{\log \left( \frac{1 - \alpha - \beta}{\max(\alpha, \beta)} \right)} \quad \text{and} \quad n(\alpha, \beta) = \frac{\alpha \beta (2 - \alpha - \beta)}{(\alpha + \beta)^3} \left( \Phi^{-1}(\frac{1}{2}(1 + s)) \right)^2,$$

where $\Phi^{-1}$ is the inverse of the standard normal cumulative distribution function. The expressions for $m$ and $n$ were originally presented in [10]. Derivations
however were not provided and the expressions contain two oversights: in the
formula for $m$ the absolute value is missing in the denominator and in the
formula for $n$ the inverse of $\Phi$ should be used instead of $\Phi$. We provide detailed
derivations of the expressions for $m$ and $n$ in the Appendices A and B
respectively.

Since $\alpha$ and $\beta$ are unknown, they need to be estimated. This is achieved
iteratively in the two-state Markov chain approach of [10]. It starts with sam-
ppling an arbitrary initial length trajectory, which is then used for estimating
the values of $\alpha$ and $\beta$. $M$ and $N$ are calculated based on these estimates. Next,
the trajectory is extended to reach the required length, and $\alpha$ and $\beta$ 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 [10].

The two-state Markov chain approach can be viewed as an aggregation
method for reducing the state space. Generically, aggregation methods aggregate
groups of nodes in the original Markov chain in accordance with a given
partition function, which leads to a smaller transition graph. In principle, the
aggregation can be any Markov chain over this smaller transition system. What
remains is the choice of the specific aggregation which is determined by the
choice of the transition probabilities. The partition function is also used to ob-
tain the so-called projection of the original process, i.e., the realisation of the
original Markov chain is projected through the partition function. Ideally, the
aggregated chain and the projected realisation should coincide. However, since
the projection is in general not Markovian, the aggregation which is “closest”
to the projection is considered instead and the “closeness” has to be appro-
priately defined. The problem of finding the optimal partition function in the
case where the distance between the projection and the aggregation is quanti-
fied by the Kullback-Leibler divergence rate (KLDR) has been recently studied, see,
e.g., [16,17,18]. The focus in these works is on finding the optimal partition func-
tion, for which the KLDR distance between the projection and aggregation is
minimised. In our case the partition function which classifies the original states
into the two meta states is specified by the biological question under study. As
shown in [17] and [18], for a given partition function, the aggregation closest to
the projection can be analytically obtained provided the steady-state distribu-
tion of the original chain is available. The steady-state probabilities are however
our goal, thus these techniques cannot be exploited for the determination of $\alpha$
and $\beta$ transition probabilities. The iterative, statistical estimation of the transi-
tion probabilities for the aggregation remains the only viable solution.

3.2 The choice of the initial sample size

Given good estimates of $\alpha$ and $\beta$, the theory of the two-state Markov chain
presented above guarantees that the obtained value satisfies the imposed precision
requirements. However, the two-state Markov chain approach starts with gener-
ating 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 $m_0$ it the ‘burn-in’ period and $n_0$ is the sample size of the two-state Markov chain abstraction. An unfortunate choice may lead to first estimates of $\alpha$ and $\beta$ 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 $\alpha$, $\beta$ 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 example 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 $k = 1$, $\epsilon = 10^{-6}$, $r = 10^{-5}$, $s = 0.95$, $m_0 = 5$, and $n_0 = 1,920$ the first estimated values for $\alpha$ and $\beta$ were $1.1918 (0.0005214)$ 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 $\alpha$ and $\beta$ were $1.1997$ 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 $r$. Independently repeating the estimation for $10^4$ times resulted in estimates of the steady-state probabilities that were outside the pre-specified precision interval $90\%$ of times. Given the rather large number of repetitions, it can be concluded that the specified $95\%$ confidence interval was not reached in this case.

The reason for the biased result is the unfortunate initial value for $n_0$ and the fact that the real value of $\alpha$ is small. In the initialisation phase the value of $\alpha$ is underestimated and $\lceil n(\alpha, \beta) \rceil$ calculated based on the estimated values of $\alpha$ and $\beta$ is almost the same as $n_0$. Hence, subsequent extension of the trajectory does not provide any improvement to the underestimated value of $\alpha$ since the elongation is short and the algorithm halts after the next iteration.

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. Let $n_0$ be the initial size of the sample used for initial estimation of $\alpha$ and $\beta$. We assume that neither $\alpha$ nor $\beta$ is zero. Then, the smallest possible estimates for both $\alpha$ and $\beta$ are greater than $10^{-4}$ if $\alpha$ and $\beta$ 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 $\alpha$ and $\beta$ and the algorithm asks for
a significant increase of the sample size. We therefore conclude that the bad-mixing Markov chain case can be properly handled by the algorithm. 

- Both first estimates of \( \alpha \) and \( \beta \) 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 \( \alpha, \beta > 0.98 \) than 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.

- 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.** In order to avoid this situation, we determine \( n_0 \) which in principle could lead to initial inaccurate estimates of \( \alpha \) or \( \beta \) and such that the next sample size given by \( \lceil n(\alpha, \beta) \rceil \) would practically not allow for an improvement of the estimates. We reason as follows. 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 \( n_0 \). Therefore, 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 \( s \).

| \( r \) | 0.01 | 0.001 | 0.0001 |
|---|---|---|---|
| \( s \) | 0.9 | 0.95 | 0.975 |
| \( n_0 \in \emptyset \) | [2.136] | [2.1161] | [2.1383] |
| | [2.1582] | [2.11628] | [2.13857] |
| | [2.15847] | [2.13857] | [2.15847] |

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 \( s \).

**Second approach: controlled initial estimation of \( \alpha \) and \( \beta \).** The formula for \( n \) is asymptotically valid for a two-state Markov chain provided that the values for \( \alpha \) and \( \beta \) 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 a 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. Let $n_0$ be the initial sample size. The algorithm samples a trajectory of the given Markov chain and estimates the values of $\alpha$ and $\beta$. It might be the case that the initial sample size is not big enough to provide non-zero estimates for the two parameters. 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 both $\alpha$ and $\beta$ are obtained. We introduce the following notation: $\hat{\alpha}$ and $\hat{\beta}$ are the non-zero estimates of the values of $\alpha$ and $\beta$, respectively. Furthermore, let $n_0'$ be the sample size used to obtain first non-zero estimates $\hat{\alpha}$ and $\hat{\beta}$, i.e., $n_0'$ is either $n_0$ or is a power of 2 multiple of $n_0$.

Once the non-zero estimates are available, 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, let us assume from now on that $\hat{\alpha} < \hat{\beta}$, which suggests that $\min(\alpha, \beta) = \alpha$. The aim is to have a good estimate for the value of $\alpha$. Notice that the smallest possible initial value $\hat{\alpha}$ can have is $\frac{1}{n_0}$. We refer to this value as the resolution of estimation. Given the resolution, one cannot distinguish between values 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. Therefore, 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 even without a single execution of the main part. Consequently, the computational burden is larger than required by the main part to reach the desired precision specified by $r$ and $s$ parameters.

In order to reduce this unnecessary computational 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 meta-states. Let us name them 0 and 1 in such a way that $\alpha$ is the probability of making the transition from meta-state 0 to meta-state 1. We use $0 \rightarrow 1$ to denote this transition. The estimate $\hat{\alpha}$ is computed as a ratio of the number of transitions from meta-state 0 to meta-state 1 to the number of transition from meta-state 0. Let $n_{0,\alpha}'$ be the number of transitions in the sample starting from meta-state 0. Let $X_i, i = 1, 2, \ldots, n_{0,\alpha}'$, be a random variable defined as follows:

$$X_i = \begin{cases} 1 & \text{ith transition from meta-state 0 is } 0 \rightarrow 1, \\ 0 & \text{otherwise.} \end{cases}$$

Notice that meta-state 0 is an accessible atom in the terminology of the theory of Markov chains, i.e., the Markov chain regenerates after entering meta-state 0, and hence the random variables $X_i, i = 1, 2, \ldots, n_{0,\alpha}'$, are independent. They are Bernoulli distributed with parameter $\alpha$. The unbiased estimate of the population
variance from the sample, denoted \( \hat{s}^2 \), is given by \( \hat{s}^2 = \hat{\alpha} \cdot (1 - \hat{\alpha}) \cdot \frac{n_{0,\alpha}}{n_{0,\alpha}-1} \). Due to independence, \( \hat{s}^2 \) is also the asymptotic variance and, in consequence, the sample size that provides the specified confidence level for the estimate of the value of \( \alpha \) 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 meta-state 0 with steady-state probability \( \beta \alpha + \beta \). Then, given that the chain reached the steady-state distribution, the expected number of regenerations, i.e., returns to meta-state 0, in a sample of size \( n \) is given by \( \frac{n \cdot \beta \alpha + \beta}{n \cdot \beta \alpha + \beta} \). Therefore, the sample size used to estimate the value of \( \alpha \) with the specified confidence level \( s \) is given by \( \frac{\alpha + \beta}{\beta} \cdot n_{\alpha,s}(\hat{\alpha}, n'_{0,\alpha}) \). As the real values of \( \alpha \) and \( \beta \) are unknown, the estimated values \( \hat{\alpha} \) and \( \hat{\beta} \) can be used in the above formula.

**Third approach: simple heuristics.** When performing the initial estimation of \( \alpha \) and \( \beta \), we require both the count of transitions from meta-state 0 to meta-state 1 and the count of transitions from meta-state 1 to meta-state 0 be at least 2. 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 leads to good initial estimates of the \( \alpha \) and \( \beta \) probabilities.

### 4 Evaluation

We implemented the two-state Markov chain approach with the simple heuristics presented in Section 3 and the Skart method of [11] in the tool ASSA-PBN, which was specially designed for steady-state analysis of large PBNs [19] (see Section 2.2 for the theoretical background of PBNs). We verified with experiments that with use of the simple heuristics, the two-state Markov chain approach could meet the predefined precision requirement even in the case of an unlucky initial sample size. For the steady-state analysis of large PBNs, applications of these two methods necessitate generation of trajectories of significant length. To achieve this in an efficient way, we applied the alias method [20] to sample the consecutive trajectory state. This enables ASSA-PBN, for example, to simulate 4,800 steps within 1s for a 2,000 nodes PBN (state-space of size \( 2^{2,000} \)).

We choose the Skart method [11] 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 [11]. 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 [11]. In a brief, high-level summary, 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 [11].
Table 2: Performance comparison of the Skart and the two-state MC methods. Explanations in the text.

The Skart method differs in two 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. Second, the Skart method applies the student distribution for skewness adjustment while the two-state Markov chain approach makes use of the normal distribution for confidence interval calculations.

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 the set \{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 $D$: 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 $\{10^{-2}, 5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}, 8 \times 10^{-5}, 5 \times 10^{-5}\}$. We set $\epsilon$ 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 5263 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 $\frac{t_{Skart}(i) - t_{TS}(i)}{t_{Skart}(i)} \geq \frac{k}{100}$. The definition for the Skart method to be faster than the two-state Markov chain approach is symmetric. In Table 2 we show the percentage of cases in which the two-state Markov chain approach was by $k$ per cent faster than Skart and vice versa for different $k$. In general, in about 70% of the results, the two-state Markov chain was faster than Skart. It is also clear that the number of cases the two-state Markov chain approach was faster than Skart is larger than in the opposite case.

| $k$ | 0  | 5  | 10 | 15 | 20 | 25 | 30 |
|-----|----|----|----|----|----|----|----|
| $t_{TS} \leq t_{Skart}$ | 69.83% | 56.10% | 41.02% | 31.03% | 25.86% | 22.76% | 20.05% |
| $t_{Skart} \leq t_{TS}$ | 30.38% | 18.68% | 11.10% | 7.39% | 5.49% | 4.43% | 3.74% |

Table 2: Performance comparison of the Skart and the two-state MC methods.
We show in Table 3 the trajectory sizes and the time costs for computing steady-state probabilities of two large PBNs using the two-state Markov chain approach and the Skart method for different precision requirements. The two analysed PBNs consist of 1,000 and 2,000 nodes, which give rise to state spaces of sizes exceeding $10^{300}$ and $10^{600}$, respectively.

### Table 3: Approximate steady-state analysis of two large PBNs

| node number | method                  | precision | 0.01 | 0.005 | 0.001 | 0.01 | 0.005 | 0.001 |
|-------------|-------------------------|-----------|------|-------|-------|------|-------|-------|
|             | the two-state Markov chain |           | 35,066 | 133,803 | 3,402,637 | 37,909 | 139,672 | 3,272,949 |
|             | trajectory size          |           | 6.19  | 23.53 | 616.26 | 7.02 | 24.39 | 590.26 |
|             | time cost (s)             |           | 3,402,637 | 37,909 | 139,672 | 3,272,949 | 616.26 | 7.02 | 24.39 | 590.26 |
| 2,000       | trajectory size          |           | 64,057 | 240,662 | 5,978,309 | 63,674 | 273,942 | 5,936,060 |
|             | time cost (s)             |           | 20.42 | 67.60 | 1,722.86 | 20.65 | 78.53 | 1,761.05 |

5 A Biological Case study

A multicellular organism consists of cells that form a highly organised community. The number of cells in this system is tightly controlled by mechanisms that regulate the cell division and the cell death. One of these mechanisms is the programmed cell death, also referred to as apoptosis: if cells are damaged, infected, or no longer needed, the intracellular death program is activated, which leads to fragmentation of the DNA, shrinkage of the cytoplasm, membrane changes and cell death without lysis or damage to neighbouring cells. This process is regulated by a number of signaling pathways which are extensively linked by cross-talk interactions. In [21], a large-scale Boolean network of apoptosis in hepatocytes was introduced, where the assigned Boolean interactions for each molecule were derived from literature study. In [22], the original multi-value Boolean model was recast into the PBN framework: a binary PBN model which comprises 91 nodes (state-space of size $2^{91}$) and 102 interactions was constructed (see Figure 2 in Appendix E for the wiring of the PBN model). With respect to the original multi-value Boolean model, the PBN model was extended by including the possibility of activation of NFκB through Caspase 8; cf. the so-called ‘extended apoptosis model’ in [22]. The model was fitted to experimental data obtained in response to six different stimulations of the input nodes (see [22] for details).

As can be seen from the wiring of the network (see Figure 2 in Appendix E), the activation of complex2 by RIP-deubi can take place in two ways: 1) by a positive feedback loop from activated Caspase 8 (C8*) and P → truncated Bid (tBid) → Bax → smac → RIP-deubi → complex2 → C8*-complex2 → C8*, and 2) by the positive signal from UV-B irradiation (input node UV) → Bax → smac → RIP-deubi → complex2. The former to be active requires the stimulation of the type 2 receptor (T2R). The latter way requires complex1 to be active, which cannot happen without the stimulation of the TNF receptor-1. Therefore, RIP-deubi can activate complex2 only in the condition of co-stimulation by TNF and
In consequence, it was suggested in [22] that the interaction of activation of complex2 via RIP-deubi is not relevant and could be removed from the model in the context of modelling primary hepatocyte. However, due to the problem with efficient generation of very long trajectories in optPBN toolbox, this hypothesis could not be supported with quantitative results ([22]).

In this work, we take up this challenge and we quantitatively investigate the relevancy of the interaction of activation of complex2 via RIP-deubi. We perform an extensive analysis in the context of co-stimulation by TNF and UV: we compute steady-state influences for the complex2 node as well as 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 [19] to compute the relevant steady-state probabilities for the best-fit models described in [22]. Due to the efficient implementation, the ASSA-PBN tool can easily deal with trajectories of length exceeding $2 \times 10^9$.

We consider 20 distinct parameter sets that resulted in the best fit of the model to the experimental data. We took all the optimisation results from the three independent runs from [22], 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 all these resulted in reasonable models with respect to the fit.

As mentioned above, we fix the experimental context to co-stimulation of TNF and UV. We note that originally in [21] UV-B irradiation conditions were imposed via the 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 [22], 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) as well as 2) co-stimulation of TNF and UV(2). Node complex2 has two independent predictor functions: complex2 = complex1 $\land$ FADD or complex2 = complex1 $\land$ FADD $\land$ RIP-deubi. The selection probabilities are denoted as $c_1^{(\text{complex2})}$ and $c_2^{(\text{complex2})}$, respectively. Their values have been optimised in [22].

We start with computing the influences with respect to the steady-state distribution, i.e., the long-term influences on complex2 from each of its parent nodes: RIP-deubi, complex1, and FADD, in accordance with the definition in [23]. 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 4. They are consistent across the different parameter sets and clearly indicate that the influence of RIP-deubi on complex2 is small compared to the influence of complex1 or FADD on complex2. However, the influence of RIP-deubi is not negligible.

We take the analysis of the importance of the interaction between RIP-deubi and complex2 further and we compute the long-run sensitivities with respect to various perturbations as defined in [24]. In particular, we perturb the selection probability $c_2^{(\text{complex2})}$ by $\pm 5\%$ and compute how the joint steady-state distribution for (apoptosis,C3ap17,NF$\kappa$B) differs from the non-perturbed one with
Table 4: Long-term influences of RIP-deubi, complex1, and FADD on complex2 in the ‘extended apoptosis model’ in [22] under the co-stimulation of TNF and UV(1) or TNF and UV(2).

|          | TNF and UV(1) | TNF and UV(2) |          | TNF and UV(1) | TNF and UV(2) |
|----------|---------------|---------------|----------|---------------|---------------|
| Uniform  | 0.2370        | 0.9981        | Uniform  | 0.2370        | 0.9980        |
| Best fit | 0.2370        | 0.9936        | Best fit | 0.2370        | 0.9937        |
| Min      | 0.2369        | 0.9980        | Min      | 0.2370        | 0.9979        |
| Max      | 0.2370        | 0.9982        | Max      | 0.2371        | 0.9982        |
| Mean     | 0.2370        | 0.9981        | Mean     | 0.2370        | 0.9981        |
| Std      | <10^{-4}      | <10^{-4}      | Std      | <10^{-4}      | <10^{-4}      |

Table 5: Long-run sensitivities with respect to perturbation of the probability value for the second predictor function for complex2 by ±5% and with respect to the elimination of the second predictor function in the ‘extended apoptosis model’ in [22] under the co-stimulation of TNF and UV(1) or TNF and UV(2).

| c2^{complex2} | TNF and UV(1) | TNF and UV(2) |          | TNF and UV(1) | TNF and UV(2) |
|---------------|---------------|---------------|----------|---------------|---------------|
| ±5%           | 0.0003        | 0.0002        | ±5%      | 0.0002        | 0.0004        |
| −5%           | 0.0011        | 0.0003        | −5%      | 0.0004        | 0.0002        |
| =0            | 0.0002        | 0.0002        | =0       | 0.0002        | 0.0007        |
| Mean          | 0.0005        | 0.0005        | Mean     | 0.0004        | 0.0009        |
| Std           | 0.0001        | 0.0001        | Std      | 0.0002        | 0.0001        |

We use the $l_1$ norm after [24]. 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., [25]). Additionally, we check the difference when the possibility of activation via Caspase 8 is removed from the system, i.e., $c^{(complex2)}_8$ is set to 0 (and, in consequence, $c^{(complex2)}_2$ 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 3. In all these cases, the sensitivities are very small. Therefore, the system turns to be insensitive to small perturbations of the value of $c^{(complex2)}_2$. Also the complete removal of the second predictor function for complex2 does not cause any drastic changes in the joint steady-state distribution for (apoptosis,C3ap17,NFκB).

Finally, we consider RIP-deubi. We fix the value of RIP-deubi to subsequently 0 and 1 and compute the respective joint steady-state distributions for (apoptosis,C3ap17,NFκB). Finally, we take the long-run sensitivity as the maximum of the changes in distributions with respect to the non-perturbed case measured with the $l_1$-norm (see [24] for details). The results, given in Table 6, show that in both variants of

**Reviving the Two-state Markov Chain Approach**

TNF and UV(1) 15

TNF and UV(2)

I

RIP-deubi

I

complex1

I

FADD

I

RIP-deubi

I

complex1

I

FADD

Best fit 0.2370 0.9981 0.9936 0.2370 0.9980 0.9937

Min 0.2369 0.9980 0.9935 0.2370 0.9979 0.9935

Max 0.2370 0.9982 0.9937 0.2371 0.9982 0.9939

Mean 0.2370 0.9981 0.9936 0.2370 0.9981 0.9937

Std <10^{-4} <10^{-4} 10^{-4} <10^{-4} 10^{-4} 10^{-4}
UV-B irradiation the sensitivities are not negligible and the function perturbations considering RIP-deubi have some impact on the steady-state distribution.

To conclude, all the obtained results indicate that in the context of co-stimulation of TNF and UV the interaction between RIP-deubi and complex2 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 complex2 is not negligible and may be important for some nodes in the network other than apoptosis, C3ap17, or NFκB.

6 Discussion and Conclusion

In this paper, we focused on two statistical methods for estimating steady-state probabilities of large PBNs: the two-state Markov chain approach and the Skart method. The Skart method follows a continuous development [11], while the two-state Markov chain approach was originally introduced by Raftery and Lewis in 1992, and only recently it was explored for the analysis of a relatively large PBN model in [22]. To revive the application of the two-state Markov chain approach, we propose a few heuristics to avoid a problem with the size of the initial sample which can lead to biased results. By extensive experiments, we show that the two-state Markov chain approach outperforms the Skart method in most cases. In the end, we illustrated the usability of the two-state Markov chain approach on a realistic biological system.

Our work in the current paper is closely related to statistical model checking [26,27], a simulation-based approach using hypothesis testing to infer whether a stochastic system satisfies a property. Most current tools for statistical model checking are restricted for bounded properties which can be checked on finite executions of the system. In recent year, both the Skart method and the perfect simulation algorithm have been explored for statistical model checking of steady state and unbounded until properties [28,29], which was considered as a future step of statistical model checking [30]. The perfect simulation algorithm for sampling the steady-state of an ergodic DTMC is based on the indigenous idea of the backward coupling scheme originally proposed by Propp and Wilson in [13]. 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. Of course, a special, more efficient variant of this method exists. If a DTMC is monotone, then it is possible to sample from the steady-state distribution by considering the maximal and minimal states only [13,14]. For example, this approach was exploited in [28] for model checking large queuing networks. Unfortunately, it is not applicable in the case of 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, in this paper we have only compared the performance of the two-state Markov chain approach with the Skart method.

In the future, we aim to investigate the usage of the discussed statistical methods for approximate steady-state analysis in a wide project on systems biology. For instance, we will further apply them to develop new techniques for minimal structural interventions to alter steady-state probabilities, which will enable the synthesis of optimal control strategies for large regulatory networks.

Acknowledgment. Experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [31] (http://hpc.uni.lu).

References

1. Shmulevich, I., Dougherty, E.R., Zhang, W.: From boolean to probabilistic boolean networks as models of genetic regulatory networks. Proceedings of the IEEE 90(11) (2002) 1778–1792
2. 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
3. Albert, R., Othmer, H.G.: The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. Journal of Theoretical Biology 223(1) (2003) 1–18
4. Bornholdt, S.: Less is more in modeling large genetic networks. Science 310(5747) (2005) 449–451
5. Wagner, A.: Circuit topology and the evolution of robustness in two-gene circadian oscillators. PNAS 102(33) (2005) 11775–11780
6. Kauffman, S.A.: Homeostasis and differentiation in random genetic control networks. Nature 224 (1969) 177–178
7. Huang, S.: Genomics, complexity and drug discovery: insights from boolean network models of cellular regulation. Pharmacogenomics 2(3) (2001) 203–222
8. Shmulevich, I., Gluhovsky, I., Hashimoto, R.F., Dougherty, E.R., Zhang, W.: Steady-state analysis of genetic regulatory networks modelled by probabilistic Boolean networks. Comparative and Functional Genomics 4(6) (2003) 601–608
9. Cowles, M.K., Carlin, B.P.: Markov chain Monte Carlo convergence diagnostics: A comparative review. J. American Statistical Association 91(434) (1996) 883–904
10. Raftery, A.E., Lewis, S.: How many iterations in the Gibbs sampler? Bayesian Statistics 4 (1992) 763–773
11. Tafazzoli, A., Wilson, J.R., Lada, E.K., Steiger, N.M.: Skart: A skewness- and autoregression-adjusted batch-means procedure for simulation analysis. In: Proc. 2008 Winter Simulation Conference. (2008) 387–395
12. Norris, J.: Markov Chains. Cambridge UK: Cambridge University Press (1998)
13. Propp, J., Wilson, D.: Exact sampling with coupled markov chains and applications to statistical mechanics. Random Structures & Algorithms 9(1) (1996) 223–252
14. Bušić, A., Ganjali, B., Vincent, J.M.: Perfect simulation and non-monotone markovian systems. In: Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools. ValueTools ’08. Brussels, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) (2008) 27:1–27:10
15. Shmulevich, I., Dougherty, E.R.: Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks. Philadelphia PA: SIAM Press (2010)
16. Deng, K., Mehta, P.G., Meyn, S.P.: A simulation-based method for aggregating Markov chains. In: Proceedings of the 48th IEEE Conference on Decision and Control, Shanghai, IEEE (December 2009) 4710–4716
17. Deng, K., Mehta, P.G., Meyn, S.P.: Optimal Kullback-Leibler aggregation via spectral theory of Markov chains. IEEE Transactions on Automatic Control 56(12) (2011) 2793–2808
18. Geiger, B.C., Petrov, T., Kubin, G., Koepp, H.: Optimal Kullback-Leibler aggregation via information bottleneck. IEEE Transactions on Automatic Control 60(4) (2015) 1010–1022
19. 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. LNCS, Springer (2015) Available at http://satos.unilu/software/ASSA-PBN/.
20. Walker, A.: An efficient method for generating discrete random variables with general distributions. ACM Trans. on Mathematical Software 3(3) (1977) 253–256
21. 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
22. Trairatphisan, P., Mizera, A., Pang, J., Tantar, A.A., Sauter, T.: optPBN: An optimisation toolbox for probabilistic boolean networks. PLOS ONE 9(7) (2014)
23. Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W.: Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 18(2) (2002) 261–274
24. Qian, X., Dougherty, E.R.: On the long-run sensitivity of probabilistic Boolean networks. Journal of Theoretical Biology 257(4) (2009) 560–577
25. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. American Mathematical Society (2009)
26. Younes, H.S., Simmons, R.: Probabilistic verification of discrete event systems using acceptance sampling. In: Proc. 14th Conference on Computer Aided Verification. Volume 2404 of LNCS., Springer (2002) 223–235
27. Sen, K., Viswanathan, M., Agha, G.: On statistical model checking of stochastic systems. In: Proc. 17th Conference on Computer Aided Verification. Volume 3576 of LNCS., Springer (2005) 266–280
28. El Rabih, D., Pekergin, N.: Statistical model checking using perfect simulation. In: Proc. 7th Symposium on Automated Technology for Verification and Analysis. Volume 5799 of LNCS. (2009) 120–134
29. Rohr, C.: Simulative model checking of steady state and time-unbounded temporal operators. Transactions on Petri Nets and Other Models of Concurrency 8 (2013) 142–158
30. Legay, A., Delahaye, B., Bensalem, S.: Statistical model checking: An overview. In: Proc. 1st Conference on Runtime Verification. Volume 6418 of LNCS., Springer (2010) 122–135

31. Varrette, S., Bouvry, P., Cartiaux, H., Georgatos, F.: Management of an academic HPC cluster: The UL experience. In: Proc. Conference on High Performance Computing & Simulation, IEEE CS (2014) 959–967
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 $\pi = [\pi_0 \, \pi_1]$ =

\[
\begin{bmatrix}
\beta \\
\alpha + \beta
\end{bmatrix}
\]

Let $E_\pi(Z_t)$ denote the expected value of $Z_t$ for any fixed $t \geq 0$, with respect to the steady-state distribution $\pi$. We have that $E_\pi(Z_t) = \frac{\alpha}{\alpha + \beta}$.

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

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

where $\lambda$ is the second eigenvalue of $P$.

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

$$ \left| \begin{bmatrix} P[Z_m = 0 \mid Z_0 = j] \\ P[Z_m = 1 \mid Z_0 = j] \end{bmatrix} - \begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} \right| < [\epsilon \, \epsilon] $$

for some $\epsilon > 0$. For any vector $v = [v_1 \, v_2 \ldots \, v_n]^T \in \mathbb{R}^n$ we use $|v|$ to denote $|v_1| \cdot |v_2| \ldots \cdot |v_n|^T$, where $T$ is the transposition operator. If $e_0 = [1 \, 0]$ and $e_1 = [0 \, 1]$, then for $j \in \{0, 1\}$ we have that

$$ \left| \begin{bmatrix} P[Z_m = 0 \mid Z_0 = j] \\ P[Z_m = 1 \mid Z_0 = j] \end{bmatrix} - \begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} \right| = e_j P^m. $$

With (1) and (3), condition (2) can be rewritten as

$$ \left| e_j \left( \begin{bmatrix} \pi_0 & \pi_1 \\ \pi_0 & \pi_1 \end{bmatrix} + \frac{\lambda^m}{\alpha + \beta} \cdot \begin{bmatrix} \alpha & -\alpha \\ -\beta & \beta \end{bmatrix} \right) - \begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} \right| < [\epsilon \, \epsilon]. $$

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

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

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

$$ \frac{\lambda^m \alpha}{\alpha + \beta} < \epsilon \quad \text{and} \quad \frac{\lambda^m \beta}{\alpha + \beta} < \epsilon, $$

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

$$ |\lambda|^m < \frac{\epsilon(\alpha + \beta)}{\alpha} \quad \text{and} \quad |\lambda|^m < \frac{\epsilon(\alpha + \beta)}{\beta}. $$
Equivalently, $m$ has to satisfy

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

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

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

Then, by taking the logarithm to base 10 on both sides\(^3\), we have that

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

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

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

### B Derivation of the sample size

By the Law of Large Numbers for irreducible positive recurrent Markov chains $Z_n \to \pi_1 \ a.s.$ with $n \to \infty$, where $Z_n = \frac{1}{n} \sum_{t=1}^{n} Z_t$. Now, by a variant of the Central Limit Theorem for non-independent random variables\(^4\) for $n$ large, $Z_n$ is approximately normally distributed with mean $\pi_1 = \frac{\alpha}{\alpha + \beta}$ and asymptotic variance $\sigma^2_{as} = \frac{1}{n} \frac{\alpha \beta (2 - \alpha - \beta)}{(\alpha + \beta)}$, see Section[C] for the derivation of the asymptotic variance. Let $X$ be the standardised $Z_n$, i.e.,

$$X = \frac{Z_n - \pi_1}{\sigma_{as}}.$$ 

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 $P[\pi_1 - r \leq Z_n \leq \pi_1 + r] = s$ is satisfied for some specified $r$ and $s$. This condition can be rewritten as

$$P[-r \leq Z_n - \pi_1 \leq r] = s,$$

and further as

$$P[-r \cdot \frac{\sqrt{n}}{\sigma_{as}} \leq \frac{Z_n - \pi_1}{\sigma_{as}/\sqrt{n}} \leq r \cdot \frac{\sqrt{n}}{\sigma_{as}}] = s.$$

\(^3\) In fact, by the formula for change of base for logarithms, the natural logarithm (ln), the logarithm to base 2 (log₂), 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!

\(^4\) Notice that the random variables $Z_t, Z_{t+1}$ which values are consecutive states of a trajectory are correlated and are not independent.
which is
\[ P[-r \frac{\sqrt{n}}{\sigma_{as}} \leq X \leq r \frac{\sqrt{n}}{\sigma_{as}}] = s. \]

Since \( X \sim N(0, 1) \) and \( N(0, 1) \) is symmetric around 0, it follows that
\[ P[0 \leq X \leq r \frac{\sqrt{n}}{\sigma_{as}}] = \frac{s}{2} \]
and
\[ P[X \leq r \frac{\sqrt{n}}{\sigma_{as}}] = \frac{1}{2} + \frac{s}{2} = \frac{1}{2}(1 + s). \]

Let \( \Phi(\cdot) \) be the standard normal cumulative distribution function. Then the above can be rewritten as
\[ \Phi(r \frac{\sqrt{n}}{\sigma_{as}}) = \frac{1}{2}(1 + s). \]

Therefore, if we denote the inverse of the standard normal cumulative distribution function with \( \Phi^{-1}(\cdot) \), we have that
\[ r \frac{\sqrt{n}}{\sigma_{as}} = \Phi^{-1}(\frac{1}{2}(1 + s)). \]

In consequence,
\[ n = \frac{\sigma_{as}^2}{\left\{ \Phi^{-1}(\frac{r}{2}(1+s)) \right\}^2} = \frac{\alpha \beta(2-\alpha-\beta)}{(\alpha+\beta)^3} \left\{ \Phi^{-1}(\frac{r}{2}(1+s)) \right\}^2. \]

### C Derivation of the asymptotic variance

By the Central Limit Theorem for stationary stochastic processes, \( \sqrt{n}(Z_n - \pi_1) \overset{d}{\rightarrow} N(0, \sigma_{as}^2) \) as \( n \to \infty \), where \( \sigma_{as}^2 \) is the so-called asymptotic variance given by
\[ \sigma_{as}^2 = \text{Var}_\pi(Z_j) + 2 \sum_{k=1}^\infty \text{Cov}_\pi(Z_j, Z_{j+k}) \quad (4) \]
and \( \text{Var}_\pi(\cdot) \) and \( \text{Cov}_\pi(\cdot, \cdot) \) denote the variance and covariance with respect to the steady-state distribution \( \pi \), respectively. We proceed to calculate \( \sigma_{as}^2 \).

First, observe that \( \mathbb{E}_\pi(Z_nZ_{n+1}) = \frac{\alpha \beta (1-\beta)}{\alpha + \beta} \): \( Z_nZ_{n+1} \neq 0 \) 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

\(^5\) After discarding the ‘burn-in’ part of the trajectory, we can assume that the Markov chain in a stationary stochastic process.
covariance, we have that the steady-state covariance between consecutive random variables of the two-state Markov chain, i.e., $\text{Cov}_\pi(Z_n, Z_{n+1})$ is

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

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

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

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

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

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 (4), we have

$$\sigma_n^2 = \text{Var}_\pi(X_j) + 2 \sum_{k=1}^{\infty} \text{Cov}_\pi(X_{j+k})$$

$$= \frac{\alpha \beta}{(\alpha + \beta)^2} + 2 \sum_{k=1}^{\infty} (1 - \alpha - \beta)^k \cdot \frac{\alpha \beta}{(\alpha + \beta)^2}$$

$$= \frac{\alpha \beta}{(\alpha + \beta)^2} + \frac{2 \alpha \beta}{(\alpha + \beta)^2} \cdot \sum_{k=1}^{\infty} (1 - \alpha - \beta)^k$$

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

$$= \frac{\alpha \beta (2 - \alpha - \beta)}{(\alpha + \beta)^4}.$$ 

In consequence, $Z_n$ is approximately normally distributed with mean $\frac{\alpha}{\alpha + \beta}$ and variance $\frac{1}{n} \cdot \frac{\alpha \beta (2 - \alpha - \beta)}{(\alpha + \beta)^4}.$

D Derivations for the pitfall avoidance heuristics

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 $\alpha$ and $\beta$ are within the range $[\frac{1}{n_0}, 1]$. Computing the partial derivatives, equating them to zero, and solving for $\alpha$ and $\beta$ 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 $\beta$ for a moment and consider $n(\alpha, \beta)$.
as a function of \( \alpha \). We denote it as \( n_{\beta}(\alpha) \). By differentiating with respect to \( \alpha \), 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 \( c_{r,s} = \frac{r^2}{(\Phi^{-1} \left( \frac{1}{2}(1 + s) \right))^2} \).

By equating to zero and solving for \( \alpha \) we get two solutions:
\[
\alpha_1 = 2 - \sqrt{\beta^2 - 2 \beta + 4} \quad \text{and} \quad \alpha_2 = 2 + \sqrt{\beta^2 - 2 \beta + 4}.
\]
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 \( \alpha \) at \( \alpha_1 \) is negative. This shows that for any fixed \( \beta \), \( n_{\beta}(\alpha) \) grows on the interval \([1, \alpha_1]\), attains its maximum at \( \alpha_1 \) 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 \( \alpha \) and \( \beta \) 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
\[
n \left( \frac{1}{n_0}, \frac{1}{n_0} \right) = \frac{n_0 - 1}{4 c_{r,s}} \quad \text{and} \quad n \left( \frac{1}{n_0}, 1 \right) = \frac{(n_0 - 1) \cdot n_0}{c_{r,s} \cdot (1 + n_0)^3}.
\]

Now, to avoid the situation where the initial estimates of \( \alpha \) and \( \beta \) lead to \( n(\alpha, \beta) < 2n_0 \), it is enough to make sure that given \( r \) and \( s \) the following condition is satisfied: \( \min(n(\frac{1}{n_0}, \frac{1}{n_0}), n(\frac{1}{n_0}, 1)) \geq 2n_0 \). 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
\[
\begin{cases}
n_0 \in \left[ -\frac{1}{8c_{r,s} - 1}, \infty \right) \\
n_0 \in \emptyset \quad \text{if} \quad c_{r,s} < \frac{1}{8} \\
n_0 \in \emptyset \quad \text{if} \quad c_{r,s} \geq \frac{1}{8}.
\end{cases}
\]

The solution of the second inequality is more complicated, but can be easily obtained with computer algebra system software (e.g., Maple\textsuperscript{TM}). In Table \[\text{Table}1\] we present some solutions for a number of values for \( r \) and \( s \).
E The Boolean model of apoptosis

Fig. 2: The wiring of the probabilistic Boolean model of apoptosis originally introduced in [22].