Rare Event Handling in Signalling Cascades

Benoît Barbot, Serge Haddad, Monika Heiner, Claudine Picaronny

Juillet 2014

Research report LSV-14-10 **(Version 1)**

Laboratoire Spécification & Vérification

École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Rare Event Handling in Signalling Cascades

Benoît Barbot[1], Serge Haddad[1], Monika Heiner[2], and Claudine Picaronny[1]

[1] LSV, ENS Cachan & CNRS & Inria, France
{barbot, haddad, picaronny}@lsv.ens-cachan.fr
[2] Brandenburg University of Technology, Cottbus, Germany
monika.heiner@b-tu.de

Abstract Signalling cascades are a recurrent pattern of biological regulatory systems whose analysis has deserved a lot of attention. It has been shown that stochastic Petri nets are appropriate to model such systems and evaluate the probabilities of specific properties. Such an evaluation can be done numerically when the combinatorial state space explosion is manageable or statistically otherwise. However, when the probabilities to be evaluated are too small, random simulation requires more sophisticated techniques for the handling of rare events. In this paper, we show how such involved methods can be successfully applied for signalling cascades. More precisely, we study three relevant properties of a signalling cascade with the help of the Cosmos tool. Our experiments point out interesting dependencies between quantitative parameters of the regulatory system and its transient behaviour. In addition, they demonstrate that we can go beyond the capabilities of Marcie, which provides one of the most efficient numerical solvers.

1 Introduction

Signalling cascades. Signalling processes play a crucial role for the regulatory behaviour of living cells. They mediate input signals, i.e. the extracellular stimuli received at the cell membrane, to the cell nucleus, where they enter as output signals the gene regulatory system. Understanding signalling processes is still a challenge in cell biology. To approach this research area, biologists design and explore signalling networks, which are likely to be building blocks of the signalling networks of living cells. Among them are the type of signalling cascades which we investigate in our paper.

A signalling cascade is a set of reactions which can be grouped into levels. At each level a particular enzyme is produced (e.g. by phosphorylation); the level generally also includes the inverse reactions (e.g., dephosphorylation). The system constitutes a cascade since the enzyme produced at some level is the catalyser for the reactions at the next level. The catalyser of the first level is usually considered to be the input signal, while the catalyser produced by the last level constitutes the output signal. The transient behaviour of such a system presents a characteristic shape, the quantity of every enzyme increases to some stationary value. In addition, the increases are temporally ordered w.r.t.

the levels in the signalling cascade. This behaviour can be viewed as a signal travelling along the levels, and there are many interesting properties to be studied like the travelling time of the signal, the relation between the variation of the enzymes of two consecutive levels, etc.

In [11], it has been shown how such a system can be modelled by a Petri net which can either be equipped with continuous transition firing rates leading to a continuous Petri net which determines a set of differential equations or by stochastic transition firing rates leading to a stochastic Petri net. This approach emphasises the importance of Petri nets which, depending on the chosen semantics, permit to investigate particular properties of the system. In this paper we wish to explore the influence of stochastic features on the signalling behaviour, and thus we focus on the use of stochastic Petri nets.

Analysis of stochastic Petri nets can be performed either numerically or statistically. The former approach is much faster than the latter and provides exact results up to numerical approximations, but its application is limited by the memory requirements due to the combinatory explosion of the state space.

Statistical evaluation of rare events. Statistical analysis means to estimate the results by evaluating a sufficient number of simulations. However, standard simulation is unable to efficiently handle *rare events*, i.e. properties whose probability of satisfaction is tiny. Indeed the number of trajectories to be generated in order to get an accurate interval confidence for rare events becomes prohibitively huge. Thus *acceleration* techniques [17] have been designed to tackle this problem whose principles consist in (1) favouring trajectories that satisfy the property, and (2) numerically adjusting the result to take into account the bias that has been introduced. This can be done by *splitting* the most promising trajectories [15] or *importance sampling* [10], i.e. modifying the distribution during the simulation. In previous work [4], some of us have developed an original importance sampling method based on the design and numerical analysis of a reduced model in order to get the importance coefficients. First proposed for checking "unbounded until" properties over models whose semantics is a discrete time Markov chain, it has been extended to also handle "bounded until" properties and continuous time Markov chains [5].

Our contribution. In this paper we consider three families of properties for signalling cascades that are particularly relevant for the study of their behaviour and that are (depending on a scaling parameter) potentially rare events. From an algorithmic point of view, this case study raises interesting issues since the combinatorial explosion of the model quickly forbids the use of numerical solvers and its intricate (quantitative) behaviour requires elaborated and different abstractions depending on the property to be checked.

Due to these technical difficulties, the signalling cascade analysis has led us to substantially improve our method and in particular the way we obtain the final confidence interval. From a biological point of view, experiments have pointed out interesting dependencies between the scaling parameter of the model and the probability of satisfying a property.

Organisation. In Section 2 we present the biological background, the signalling cascade under study and the properties to be studied. Then in Section 3 after some recalls on stochastic Petri nets, we model signalling cascades by SPNs. We introduce the rare event issue and the importance sampling technique to cope with in Section 4. In Section 5, we develop our method for handling rare events. Then in Section 6 we report and discuss the results of our experiments. Finally in Section 7, we conclude and give some perspectives to our work.

2 Signalling cascades

In technical terms, signalling cascades can be understood as networks of biochemical reactions transforming input signals into output signals. In this way, signalling processes determine crucial decisions a cell has to make during its development, such as cell division, differentiation, or death. Malfunction of these networks may potentially lead to devastating consequences on the organism, such as outbreak of diseases or immunological abnormalities. Therefore, cell biology tries to increase our understanding of how signalling cascades are structured and how they operate. However, signalling networks are generally hard to observe and often highly interconnected, and thus signalling processes are not easy to follow. For this reason, typical building blocks are designed instead, which are able to reproduce observed input/output behaviours.

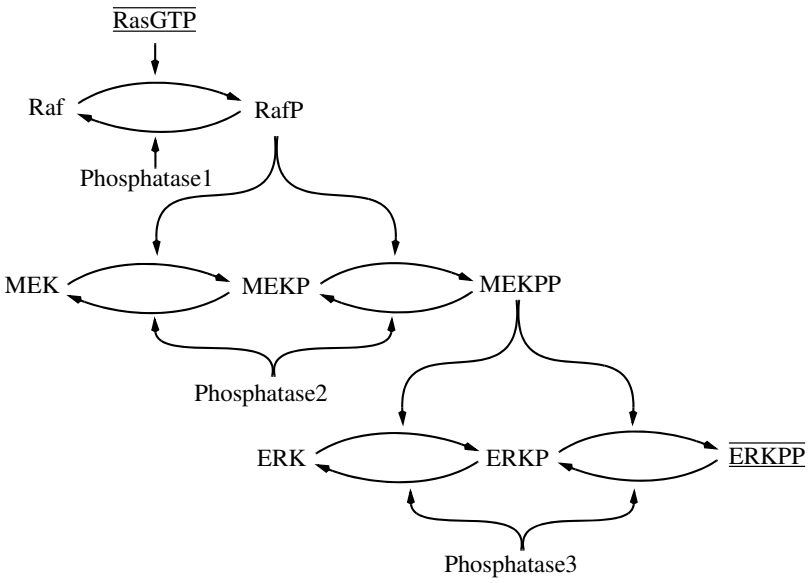

Figure 1. The general scheme of the considered three-level signalling cascade; RasGTP serves as input signal and ERKPP as output signal.

The case study we have chosen for our paper is such a signalling building block: the mitogen-activated protein kinase (MAPK) cascade [16]. This is the core of the ubiquitous ERK/MAPK network that can, among others, convey cell division and differentiation signals from the cell membrane to the nucleus. The description starts at the RasGTP complex which acts as an enzyme (kinase) to phosphorylate Raf, which phosphorylates MAPK/ERK Kinase (MEK), which in turn phosphorylates Extracellular signal Regulated Kinase (ERK). We consider RasGTP as the input signal and ERKPP (activated ERK) as the output signal. This cascade (RasGTP $\rightarrow$ Raf $\rightarrow$ MEK $\rightarrow$ ERK) of protein interactions is known to control cell differentiation, while the strength of the effect depends on the ERK activity, i.e. concentration of ERKPP.

The scheme in Figure 1 describes the typical modular structure for such a signalling cascade, see [7]. Each layer corresponds to a distinct protein species. The protein Raf in the first layer is only singly phosphorylated. The proteins in the two other layers, MEK and ERK respectively, can be singly as well as doubly phosphorylated. In each layer, forward reactions are catalysed by kinases and reverse reactions by phosphatases (Phosphatase1, Phosphatase2, Phosphatase3). The kinases in the MEK and ERK layers are the phosphorylated forms of the proteins in the previous layer. Each phosphorylation/dephosphorylation step applies mass action kinetics according to the pattern $A+E \rightleftharpoons AE \rightarrow B+E$. This pattern reflects the mechanism by which enzymes act: first building a complex with the substrate, which modifies the substrate to allow for forming the product, and then disassociating the complex to release the product; for details see [6].

Having the wiring diagram of the signalling cascade, a couple of interesting questions arise whose answers would shed some additional light on the subject under investigation. Among them are an assessment of the signal strength in each level, and specifically of the output signal. We will consider these properties in Sections 6.1 and 6.2. The general scheme of the signalling cascade also suggests a temporal order of the signal propagation in accordance with the level order. What cannot be derived from the structure is the extent to which the signals are simultaneously produced; we will discuss this property in Section 6.3.

3 Petri net modelling

Stochastic Petri nets. Due to their graphical representation and bipartite nature, Petri nets are highly appropriate to model biochemical networks. When equipped with a stochastic semantics, yielding stochastic Petri nets (SPN) [1], they can be used to perform quantitative analysis.

Definition 1 (SPN). *A stochastic Petri net $\mathcal{N}$ is defined by:*

- *a finite set of places P;*
- *a finite set of transitions T;*
- *a backward (resp. forward) incidence matrix* **Pre** *(resp.* **Post***) from $P \times T$ to $\mathbb{N}$.*

– *a set of state-dependent rates of transitions* $\{\mu_t\}_{t\in T}$ *such that* μ_t *is a mapping from* $\mathbb{N}^P$ *to* $\mathbb{R}_{>0}$.

A *marking* m of an SPN $\mathcal{N}$ is an item of $\mathbb{N}^P$. A transition t is fireable in marking m if for all $p \in P$ $m(p) \geq \mathbf{Pre}(p,t)$. Its firing leads to marking m' defined by: for all $p \in P$ $m'(p) = m(p) - \mathbf{Pre}(p,t) + \mathbf{Post}(p,t)$. It is denoted either as $m \xrightarrow{t} m'$ or as $m \xrightarrow{t}$ omitting the next marking. Let $\sigma = \sigma_1 \dots \sigma_n \in T^*$, then σ is fireable from m and leads to m' if there exists a sequence of markings $m = m_0, m_1, \dots, m_n$ such that for all $0 \leq k < n$, $m_k \xrightarrow{\sigma_k} m_{k+1}$. This firing is also denoted $m \xrightarrow{\sigma} m'$. Let m_0 be an initial marking, the reachability set $Reach(\mathcal{N}, m_0)$ is defined by: $Reach(\mathcal{N}, m_0) = \{m \mid \exists \sigma \in T^* \ m_0 \xrightarrow{\sigma} m\}$. The initialised SPNs $(\mathcal{N}, m_0)$ that we consider do not have deadlocks: for all $m \in Reach(\mathcal{N}, m_0)$ there exists $t \in T$ such that $m \xrightarrow{t}$.

An SPN is a high-level model whose operational semantics is a continuous time Markov chain (CTMC). In a marking m, each enabled transition of the Petri net randomly selects an execution time according to a Poisson process with rate μ_t. Then the transition with earliest firing time is selected to fire yielding the new marking. This can be formalized as follows.

Definition 2 (CTMC of a SPN). *Let* $\mathcal{N}$ *be a stochastic Petri net and* m_0 *be an initial marking. Then the CTMC associated with* $(\mathcal{N}, m_0)$ *is defined by:*

– *the set of states is* $Reach(\mathcal{N}, m_0)$*;*
– *the transition matrix* $\mathbf{P}$ *is defined by:*

$$\mathbf{P}(m, m') = \frac{\sum_{m \xrightarrow{t} m'} \mu_t(m)}{\sum_{m \xrightarrow{t}} \mu_t(m)}$$

– *the rate* λ_m *is defined by:* $\lambda_m = \sum_{m \xrightarrow{t}} \mu_t(m)$

Running case study. We now explain how to model our running case study in the Petri net framework. The signalling cascade is made of several phosphorylation/dephosphorylation steps, which are built on mass/action kinetics. Each step follows the pattern $A + E \rightleftharpoons AE \to B + E$ and is modelled by a small Petri net component depicted in Figure 2. The mass action kinetics is expressed by the rate of the transitions. The marking-dependent rate of each transition is equal to the product of the number of tokens in all its incoming places up to a multiplicative constant given by the biological behaviour (summing up dependencies on temperature, pressure, volume, etc.).

The whole reaction network based on the general scheme of a three-level double phosphorylation cascade, as given in Figure 1, is modelled by the Petri net in Figure 3. The input signal is the number of tokens in the place RasGTP, and the output signal is the number of tokens in the place ERKPP.

This signalling cascade model represents a self-contained and closed system. It is covered with place invariants (see the appendix), specifically each layer in the cascade forms a P-invariant consisting of all states a protein can undergo;

thus the model is bounded. Assuming an appropriate initial marking, the model is also live and reversible; see [11] for more details, where this Petri net has been developed and analysed in the qualitative, stochastic and continuous modelling paradigms. In our paper we extend these analysis techniques for handling properties corresponding to rare events.

We introduce a scaling factor N to parameterize how many tokens are spent to specify the initial marking. Increasing the scaling parameter can be interpreted in two different ways: either an increase of the biomass circulating in the closed system (if the biomass value of one token is kept constant), or an increase of the resolution (if the biomass value of one token inversely decreases, called level concept in [11]). The kind of interpretation does not influence the approach we pursue in this paper.

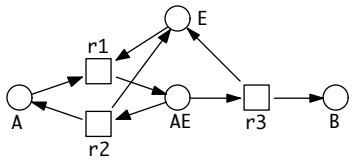

Figure 2. Petri net pattern for mass action kinetics $A + E \rightleftharpoons AE \rightarrow B + E$.

Table 1. Development of the state space for increasing N.

N	number of states	N	number of states
1	24,065 (4)	6	769,371,342,640 (11)
2	6,110,643 (6)	7	5,084,605,436,988 (12)
3	315,647,600 (8)	8	27,124,071,792,125 (13)
4	6,920,337,880 (9)	9	122,063,174,018,865 (14)
5	88,125,763,956 (10)	10	478,293,389,221,095 (14)

Increasing N means to increase the size of the state space and thus of the CTMC, as shown in Table 1 which has been computed with the symbolic analysis tool MARCIE [12]. As expected, the explosion of the state space prevents numerical model checking for higher N and thus calls for statistical model checking.

Furthermore, increasing the number of states means to actually decrease the probabilities to be in a certain state, as the total probability of 1 is fixed. With the distribution of the probability mass of 1 over an increasingly huge number of states, we obtain sooner or later states with very tiny probabilities, and thus rare events. Neglecting rare events is usually appropriate when focusing on the averaged behaviour. But they become crucial when certain jump processes such as mutations under rarely occurring conditions are of interest.

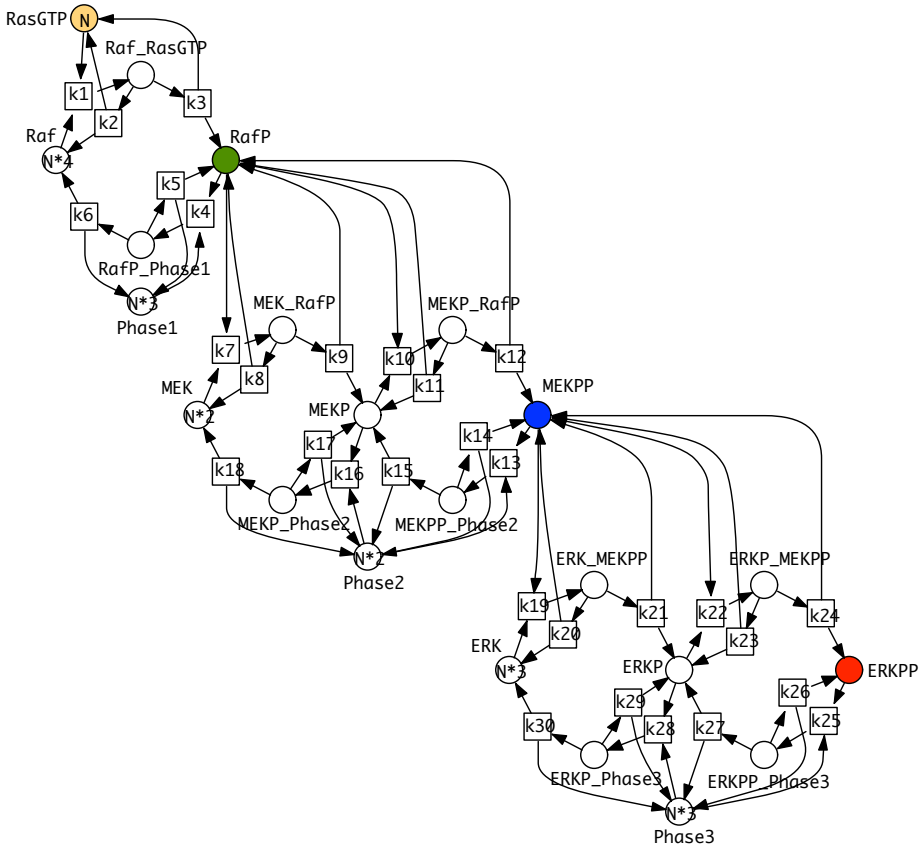

Figure 3. A Petri net modelling the three-level signalling cascade given in Figure 1; k_i are the kinetic constants for mass action kinetics, N the scaling parameter.

4 Statistical model checking with rare events

4.1 Statistical model checking and rare events

Simulation recalls. The statistical approach for evaluating the expectation $\mathbf{E}(X)$ of a random variable X related to a random path in a Markov chain is generally based on three parameters: the number of simulations K, the confidence level γ, and the width of the confidence interval lg (see [2]). Once the user provides two parameters, the procedure computes the remaining one. Then it performs K simulations of the Markov chain and outputs a confidence interval $[L, U]$ with a width of at most lg such that $\mathbf{E}(X)$ belongs to this interval with a probability of at least γ. More precisely, depending on the hypotheses, the confidence level has two interpretations: (1) either the confidence level is truly ensured, or (2) the required probability is only asymptotically valid (when K goes to infinity using central limit theorem). The two usual hypotheses for providing a *true* confidence

level are either that the distribution of X is known up to a parameter (e.g. Bernoulli law with unknown success probability) or that the random variable is bounded allowing to exploit Chernoff-Hoeffding bounds [13].

Statistical evaluation of a reachability probability. Let $\mathcal{C}$ be a discrete time Markov chain (DTMC) with two absorbing states s_+ or s_-, such that the probability to reach s_+ or s_- from any state is equal to 1. Assume one wants to estimate p, the probability to reach s_+. Then the simulation step consists in generating K paths of $\mathcal{C}$ which end in an absorbing state. Let K_+ be the number of paths ending in state s_+. The random variable K_+ follows a binomial distribution with parameters p and K. Thus the random variable $\frac{K_+}{K}$ has a mean value p and since the distribution is parametrised by p, a true confidence level can be ensured. Unfortunately, when $p \ll 1$, the number of paths required for a small confidence interval is too large to be simulated. This issue is known as the *rare event* problem.

Importance sampling. In order to tackle the rare event problem, the importance sampling method relies on a choice of a biased distribution that will artificially increase the frequency of the observed rare event during the simulation. The choice of this distribution is crucial for the efficiency of the method and usually cannot be found without a deep understanding of the system to be studied. The generation of paths is done according to a modified DTMC $\mathcal{C}'$, with the same state space, but modified transition matrix $\mathbf{P}'$. $\mathbf{P}'$ must satisfy:

$$\mathbf{P}(s, s') > 0 \Rightarrow \mathbf{P}'(s, s') > 0 \vee s' = s_- \tag{1}$$

which means that this modification cannot remove transitions that have not s_- as target, but can add new transitions. The method maintains a correction factor called L initialised to 1; this factor represents the *likelihood* of the path. When a path crosses a transition $s \to s'$ with $s' \neq s_-$, L is updated by $L \leftarrow L\frac{\mathbf{P}(s,s')}{\mathbf{P}'(s,s')}$. When a path reaches s_-, L is set to zero. If $\mathbf{P}' = \mathbf{P}$ (i.e. no modification of the chain), the value of L when the path reaches s_+ (resp. s_-) is 1 (resp. 0). Let V_s (resp. W_s) be the random variable associated with the final value of L for a path starting in x in the original model $\mathcal{C}$ (resp. in $\mathcal{C}'$). By definition, the expectation $\mathbf{E}(V_{s_0}) = p$ and by construction of the likelihood, $\mathbf{E}(W_{s_0}) = p$. Of course, a useful importance sampling should reduce the variance of W_{s_0} w.r.t. to the one of V_{s_0} equal to $p(1-p) \approx p$ for a rare event.

5 Our methodology for importance sampling

5.1 Previous work

In [4,5], we provided a method to compute a biased distribution for importance sampling: we manually design an abstract smaller model, with a behaviour close to that of the original model, that we call the *reduced model* and perform numerical computations on this smaller model to obtain the biased distribution.

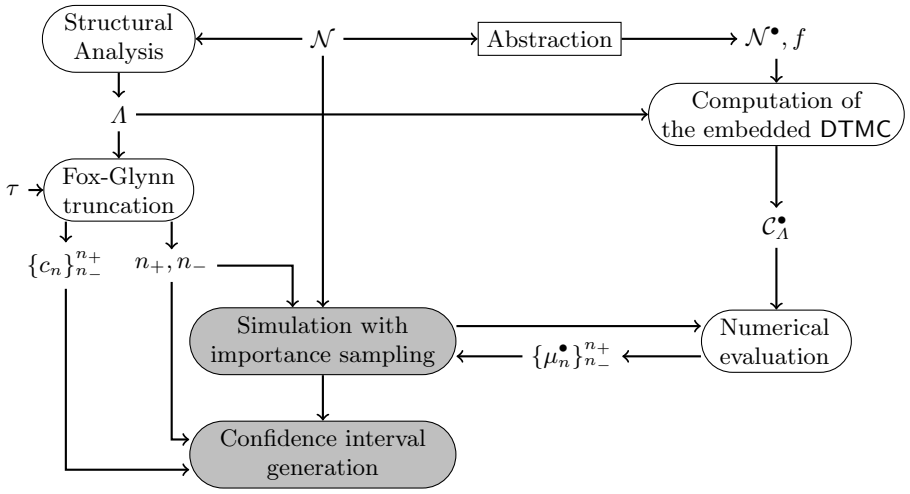

Figure 4. Principles of the methodology

Furthermore, when the correspondence of states between the original model and the reduced one satisfies a good property called the *variance reduction guarantee*, W_{s_0} is a binary random variable (i.e. a rescaled Bernoulli variable) thus allowing to get a true confidence interval with reduced size. We applied this method in order to tackle the estimation of time bounded property in CTMCs, that is the probability to satisfy a formula $aU^{[0,\tau]}b$, when it is a rare event. Let us outline the different steps of the method which is depicted in Figure 4.

Abstraction of the model. As discussed above, given a SPN $\mathcal{N}$ modelling the system to be studied, we manually design an appropriate reduced one $\mathcal{N}^\bullet$ and a correspondence function f from states of $\mathcal{N}$ to states of $\mathcal{N}^\bullet$. Function f is defined at the net level (see Section 6).

Structural analysis. Importance sampling was originally proposed for DTMCs. In order to apply it for CTMC $\mathcal{C}$ associated with net $\mathcal{N}$, we need to uniformize $\mathcal{C}$ (and also $\mathcal{C}^\bullet$ associated with $\mathcal{N}^\bullet$) which means finding a bound Λ for exit rate of states, i.e. markings, considering Λ as the uniform exit rate of states and rescaling accordingly the transition probability matrices [14]. Since the rates of transitions depend on the current marking, determining Λ requires a structural analysis like invariant computations for bounding the number of tokens in places.

Fox-Glynn truncation. Given a uniform chain with initial state s_0, exit rate Λ, and transition probability matrix $\mathbf{P}$, the state distribution π_τ at time τ is obtained by the following formula:

$$\pi_\tau(s) = \sum_{n \geq 0} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \mathbf{P}^n(s_0, s).$$

This value can be estimated, with sufficient precision, by applying [8]. Given two numerical accuracy requirements α and β, truncation points n^- and n^+

and values $\{c_n\}_{n^- \leq n \leq n^+}$ are determined such that for all $n^- \leq n \leq n^+$:

$$c_n(1-\alpha-\beta) \leq \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq c_n \text{ and } \sum_{n<n^-} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq \alpha \sum_{n>n^+} \frac{e^{-\Lambda\tau}(\Lambda\tau)^n}{n!} \leq \beta$$

Computation of the embedded DTMC. Since $\mathcal{N}^\bullet$ has been designed to be manageable, we build the embedded DTMC $\mathcal{C}^\bullet_\Lambda$ of $\mathcal{N}^\bullet$ after uniformization. More precisely, since we want to evaluate the probability to satisfy formula $aU^{[0,\tau]}b$, the states satisfying a (resp. $\neg a \wedge \neg b$) are aggregated into an absorbing accepting (resp. rejecting) state. Thus the considered probability $\mu_\tau(s^\bullet)$ is the probability to be in the accepting state at time τ starting from state $s^\bullet$.

Numerical evaluation. Matrix $\mathbf{P}'$ used for importance sampling simulation in the embedded DTMC of $\mathcal{N}$ to evaluate formulas $aU^{[0,n]}b$ for $n^- \leq n \leq n^+$, is based on the distributions $\{\mu^\bullet_n\}_{0<n\leq n^+}$, where $\mu^\bullet_n(s^\bullet)$ is the probability that a random path of the embedded DTMC of $\mathcal{N}^\bullet$ starting from $s^\bullet$ fulfills $aU^{[0,n]}b$. Such a distribution is computed by a standard numerical evaluation. However since n^+ can be large, depending on the memory requirements, this computation can be done statically for all n or dynamically for a subset of such n during the importance sampling simulation (more details are given in appendix).

Simulation with importance sampling. This is done as for a standard simulation except that the random distribution of the successors of a state depend on both the embedded DTMC $\mathcal{C}_\Lambda$ and the values computed by the numerical evaluation. Moreover, all formulas $aU^{[0,n]}b$ for $n^- \leq n \leq n^+$ have to be evaluated increasing the time complexity of the method w.r.t. the evaluation of an unbounded timed until formula.

Generation of the confidence interval. The result of the simulations is a family of confidence intervals indexed by $n^- \leq n \leq n^+$. Using the Fox-Glynn truncation, we weight and combine the confidence intervals in order to return the final interval.

Algorithmic considerations. The importance sampling simulation needs the family of vectors $\{\mu^\bullet_n\}_{0<n\leq n^+}$. They can be computed iteratively one from the other with overall time complexity $\Theta(mn^+)$ where m is the number of states of $\mathcal{N}^\bullet$. More precisely, given $\mathbf{P}^\bullet$ the transition matrix of $\mathcal{C}^\bullet_\Lambda$ (taking into account the transformation corresponding to the two absorbing states with $s^\bullet_+$ the accepting one):

$$\forall s^\bullet \neq s^\bullet_+ \ \mu^\bullet_0(s^\bullet) = 0, \qquad \mu^\bullet_0(s^\bullet_+) = 1 \qquad \text{and} \qquad \mu^\bullet_n = \mathbf{P}^\bullet \cdot \mu^\bullet_{n-1}$$

Thus as illustrated by Algorithm 1, one can perform this computation before starting the importance sampling simulation. But for large values of n^+, the space complexity to store them becomes intractable. However looking more carefully at the importance sampling specification, it appears that at simulation time $n^+ - n$ one only needs two vectors $\{\mu^\bullet_n\}$ and $\{\mu^\bullet_{n-1}\}$ [5]. So depending on the memory requirements, we propose three alternative methods.

Let $l(< n^+)$ be an integer. In the precomputation stage, the second method only stores the $\lfloor \frac{n^+}{l} \rfloor + 1$ vectors $\mu^\bullet_n$ with n multiple of l in list Ls and $\mu^\bullet_{l\lfloor \frac{n^+}{l} \rfloor + 1}, \ldots, \mu^\bullet_{n^+}$ in list K (see the precomputation stage of Algorithm 2). During the simulation stage, at time n, with $n = ml$, the vector $\mu^\bullet_{n-1}$ is present neither in Ls nor in K. So the method uses the vector $\mu^\bullet_{l(m-1)}$ stored in Ls to compute iteratively all vectors $\mu^\bullet_{l(m-1)+i} = P^{\bullet i} \cdot \mu^\bullet_{l(m-1)}$ for i from 1 to $l-1$ and store them in K (see the step computation stage of Algorithm 2). Then it proceeds to l consecutive steps of simulation without anymore computations. We choose l close to $\sqrt{n^+}$ in order to minimize the space complexity of such a factorization of steps.

Let $k = \lfloor \log_2(n^+) \rfloor + 1$. In the precomputation stage, the third method only stores $k+1$ vectors in Ls. More precisely, initially using the binary decomposition of n^+ ($n^+ = \sum_{i=0}^{k} a_{n^+,i} 2^i$), the list Ls of $k+1$ vectors consists of $w_{i,n} = \mu^\bullet_{\sum_{j=i}^{k} a_{n,j} 2^j}$, for all $1 \leq i \leq k+1$ (see the precomputation step of Algorithm 3). During the simulation stage at time n, with the binary decomposition of n ($v = \sum_{i=0}^{k} a_{n,i} 2^i$), the list Ls consists of $w_{i,n} = \mu^\bullet_{\sum_{j=i}^{k} a_{n,j} 2^j}$, for all $1 \leq i \leq k+1$. Observe that the first vector $w_{1,n}$ is equal to $\mu^\bullet_n$. We obtain $\mu^\bullet_{n-1}$ by updating Ls according to $n-1$. Let us describe the updating of the list performed by the stepcomputation of Algorithm 3. Let i_0 be the smallest index such that $a_{n,i_0} = 1$. Then for $i > i_0$, $a_{n-1,i} = a_{n,i}$, $a_{n-1,i_0} = 0$ and for $i < i_0$, $a_{n-1,i} = 1$. The new list Ls is then obtained as follows. For $i > i_0$ $w_{i,n-1} = w_{i,n}$, $w_{i_0,n-1} = w_{i_0-1,n}$. Then the vectors for $i_0 < i$, the vectors $w_{i,n-1}$ are stored along iterated $2^{i_0-1}-1$ matrix-vector products starting from vector $w_{i_0,n-1}$: $w(j, v-1) = P^{\bullet 2^j} w(j+1, n-1)$.

The computation at time n requires $1+2+\cdots+2^{i_0-1}$ products matrix-vector, i.e. $\Theta(m2^{i_0})$. Noting that the bit i is reset at most $m2^{-i}$ times, the complexity of the whole computation is $\sum_{i=1}^{k} 2^{k-i}\Theta(m2^i) = \Theta(mn^+ \log(n^+))$.

The fourth method consists in computing vector $\mu^\bullet_v$ from the initial vector at each step (see Algorithm 4). In this method we only need to store two copies of the vector.

Algorithm 1

`Precomputation`$(n^+, \mu_0^\bullet, P^\bullet)$ **Result**: Ls
`// List` Ls `fulfills` $Ls(i) = \mu_i^\bullet$
$Ls(0) \leftarrow \mu_0^\bullet$
for $i = 1$ **to** n^+ **do**
  $Ls(i) \leftarrow P^\bullet Ls(i-1)$

Algorithm 2

`Precomputation`$(n^+, \mu_0^\bullet, P^\bullet)$
  Result: Ls, K
`// List` Ls `fulfills` $Ls(i) = \mu_{i \cdot l}^\bullet$
$l \leftarrow \lfloor \sqrt{n^+} \rfloor \quad w \leftarrow \mu_0^\bullet$
for i **from** 1 **to** $\lfloor \frac{n^+}{l} \rfloor l$ **do**
  $w \leftarrow P^\bullet w$ **if** $i \mod l = 0$ **then**
    $Ls(\frac{i}{l}) \leftarrow w$
`// List` K `contains` $\mu^\bullet_{\lfloor \frac{n^+}{l} \rfloor l+1}, \ldots, \mu^\bullet_{n^+}$
for i **from** $\lfloor \frac{n^+}{l} \rfloor l + 1$ **to** n^+ **do**
  $w \leftarrow P^\bullet w \quad K(i \mod l) \leftarrow w$

`Stepcomputation`$(n, l, P^\bullet, K, Ls)$
  `// Updates` K `when needed`
if $n \mod l = 0$ **then**
  $w \leftarrow Ls(\frac{n}{l} - 1)$
  for i **from** $(\frac{n}{l} - 1)l + 1$ **to** $n - 1$ **do**
    $w \leftarrow P_0^\bullet w \quad K(i \mod l) \leftarrow w$

Algorithm 3

`Precomputation`$(n^+, \mu_0^\bullet, P^\bullet)$ **Result**: Ls
`// ` Ls `fulfills` $Ls(i) = \mu^\bullet_{\sum_{j=i}^{k} a_{n^+,j} 2^j}$
$k \leftarrow \lfloor \log_2(n^+) \rfloor + 1 \quad v \leftarrow \mu_0^\bullet \quad Ls(k+1) \leftarrow v$
for i **from** k **downto** 0 **do**
  if $a_{n^+,i} = 1$ **then**
    for j **from** 1 **to** 2^i **do**
      $w \leftarrow P^\bullet w$
  $Ls(i) \leftarrow w$

`Stepcomputation`$(n, l, P^\bullet, Ls)$ `// Ls is updated accordingly to` $n-1$
$i_0 \leftarrow \min(i \mid a_{n,i} = 1) \quad w \leftarrow Ls(i_0 + 1)$
$Ls(i_0) \leftarrow n$
for i **from** $i_0 - 1$ **downto** 0 **do**
  for $j = 1$ **to** 2^i **do**
    $w \leftarrow P^\bullet w$
  $Ls(i) \leftarrow w$

Algorithm 4

`Stepcomputation`$(n, \mu_0^\bullet, P^\bullet)$ **Result**: v
`// Vector` v `equal to` $\mu_n^\bullet$
$v \leftarrow \mu_0^\bullet$
for $i = 1$ **to** n **do**
  $v' \leftarrow P^\bullet v \quad v \leftarrow v'$

Table 2. Compared complexities

Complexity	Algorithm 1	Algorithm 2	Algorithm 3	Algorithm 4
Space	mn^+	$2m\sqrt{n^+}$	$m \log n^+$	$2m$
Time for the precomputation	$\Theta(mn^+)$	$\Theta(mn^+)$	$\Theta(mn^+)$	0
Additional time for the simulation	0	$\Theta(mn^+)$	$\Theta(mn^+ \log(n^+))$	$\Theta(m(n^+)^2)$

5.2 Tackling signalling cascades

The reduced net that we design for signalling cascades does not satisfy the variance reduction guarantee. This has two consequences: (1) we can perform a much more efficient importance sampling simulation and (2) we need to propose

different ways of computing "approximate" confidence intervals. We now detail these issues.

Importance sampling for multiple formulas. A naive implementation would require to apply statistical model checking of formulas $aU^{[0,n]}b$ for all n between n^- and n^+, but such a number can be large. A more tricky alternative consists in producing all trajectories until time horizon n^+ and updating the simulation results at the end of a trajectory for all the intervals $[0, n]$ with $n^- \leq n \leq n^+$ as follows. If the trajectory has reached the absorbing rejecting state s^- then it is an unsuccessful trajectory for all intervals. Otherwise if it has reached the absorbing accepting state s^+ at time n_0 then for all $n \geq n_0$ it is a successful trajectory and for all $n < n_0$ it is unsuccessful. Doing this way, every trajectory contributes to all evaluations, and we significantly increase the sample size without increasing computational cost. The accuracy of the results is improved. However, this requires that the importance sampling associated with time interval $[0, n^+]$ is also appropriate for the other intervals and in particular with time interval $[0, n^-]$ which is the case for our experiments (see section 6).

Confidence interval estimation. The result of each trajectory of the simulation is a realisation of the random variable $W_{s_0} = X_{s_0} L_{s_0}$ where the binary variable X_{s_0} indicates whether a trajectory starting from s_0 is succesful and the positive random variable L_{s_0} is the (random) likelihood. Observe that $\mathbf{E}(W_{s_0}) = \mathbf{E}(L_{s_0}|X_{s_0} = 1)\mathbf{E}(X_{s_0})$. Since X_{s_0} follows a Bernoulli distribution, a true confidence interval can be produced for $\mathbf{E}(X_{s_0})$. For $\mathbf{E}(L_{s_0}|X_{s_0} = 1)$ several approaches are possible among them we have selected three possible computations ranked by conservation degree.

1. The more classical way to compute confidence intervals is to suppose that the distribution is Gaussian; this is asymptotically valid if the variance is finite, thanks to the central limit theorem.
2. Another method is to use a pseudo Chernoff-Hoeffding bound. Whenever the random variable is bounded, this method is asymptotically valid. In our case we will use the minimal and maximal values observed during the simulation as the bounds of L_{s_0}.
3. The last method, which is more conservative than the previous one, consists in returning the minimal and maximal observed values as the confidence interval.

6 Experiments

We have analysed three properties, the last two are inspired by [11]. Recall that the initial marking of the model is parametrized by a scaling factor N. For the first two properties, the reduced model is the same model but with local smaller scaling factors on the different layers of phosphorylation. Every state of the initial model is mapped (by f) to a state of the abstract model which has the "closest" proportion of chemical species. For instance let $N = 4$ which corresponds to 16 species of the first layer, a state with 6 tokens in Raf and 10 tokens in RafP is

mapped, for a reduced model with $N = 3$, to a state with $4 = \lfloor 6 \times 3/4 \rfloor$ tokens in Raf and $8 = \lceil 10 \times 3/4 \rceil$ tokens in RafP (see the appendix for a specification of f).

All statistical experiments have been carried out with our tool Cosmos [3]. Cosmos is a statistical model checker for the HASL logic [3]. It takes as input a Petri net (or a high-level Petri net) with general distributions for transitions. It performs an efficient statistical evaluation of the stochastic Petri net by generating a code per model and formula. In the case of importance sampling, it additionally takes as inputs the reduced model and the mapping function specified by a C function and returns the different confidence intervals.

All experiments have been performed on a machine with 16 cores running at 2 GHz and 32 GB of memory both for the statistical evaluation of Cosmos and the numerical evaluation of Marcie.

6.1 Maximal peak of the output signal

The first property is expressed as a time-bounded reachability formula assessing the strength of the output signal of the last layer: "What is the probability to reach within 10 time units a state where the total mass of ERK is doubly phosphorylated?", associated with probability p_1 defined by:

$$p_1 = \Pr(\text{True } \mathbf{U}^{\leq 10}(\mathsf{ERKPP} = 3N))$$

The inner formula is parametrized by N, the scaling factor of the net (via its initial marking). The reduced model that we design for Cosmos uses different scaling factors for the three layers in the signalling cascade. The first two layers of phosphorylation which are based on Raf and MEK always use a scaling factor of 1, whereas the last layer involving ERK uses a scaling factor of N. The second column of Table 3 shows the ratio between the number of reachable states of the original and the reduced models.

Table 3. Computational complexity related to the evaluation of p_1

N	Cosmos			Marcie	
	Reduction factor	time	memory	time	memory
1	-	-	-	4	514MB
2	38	20,072	3,811MB	326	801MB
3	558	15,745	15,408MB	43,440	13,776MB
4	4667	40,241	3,593MB	Out of Memory: >32GB	
5	27353	51,120	19,984MB		

Experimental Results. We have performed experiments with both Cosmos and Marcie. The time and memory consumptions for increasing values of N are reported in Table 3. For each value of N we generate one million trajectories with Cosmos. We observe that the time consumption significantly increases

between $N = 3$ and $N = 4$. This is due to a change of strategy in the space/time trade-off in order to not exceed the machine memory capacity. MARCIE suffers an exponential increase w.r.t. both time and space resources. When $N = 3$, it is slower than COSMOS and it is unable to handle the case $N = 4$.

Table 4. Numerical values associated with p_1

N	COSMOS			MARCIE
	Gaussian CI	Chernoff CI	MinMax CI	Output
1				2.07E-12
2	[3.75E-27,5.88E-26]	[3.75E-27,4.54E-25]	[3.75E-27,1.57E-23]	8.18E-26
3	[4.34E-42,1.72E-39]	[4.34E-42,1.82E-38]	[4.43E-42,1.87E-37]	2.56E-39
4	[1.54E-57,8.54E-56]	[1.54E-57,1.98E-55]	[1.78E-57,7.05E-55]	Out of Memory
5	[3.97E-73,2.33E-70]	[3.97E-73,7.30E-70]	[5.44E-73,2.24E-69]	

Table 4 depicts the values returned by the two tools: MARCIE returns a single value, whereas COSMOS returns three confidence intervals (discussed above) with a confidence level set to 0.99. We observe that confidence intervals computed by the Gaussian analysis neither contain the result, the ones computed by Chernoff-Hoeffding do not contain it for $N = 3$, and the most conservative ones always contain it (when this result is available). An analysis of the likelihood L_{s_0} is detailed in the appendix.

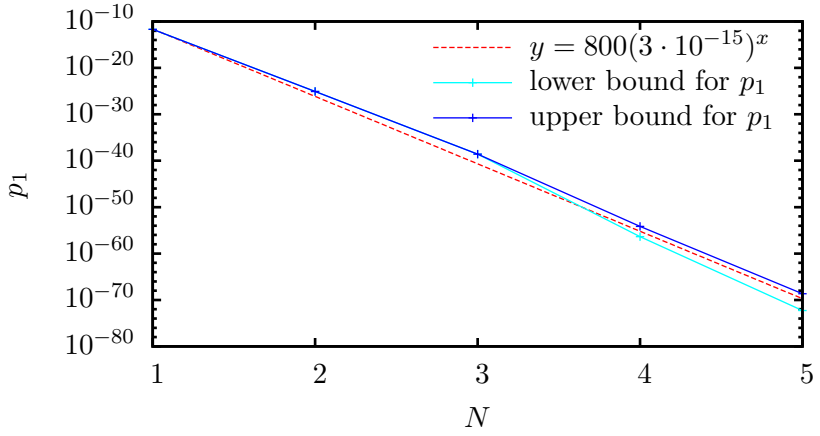

Figure 5. Highlighting an exponential dependency

Figure 5 illustrates the dependency of p_1 with respect to the scaling factor N. It appears that the probability p_1 depends on N in an exponential way. The constants occurring in the formula could be interpreted by biologists.

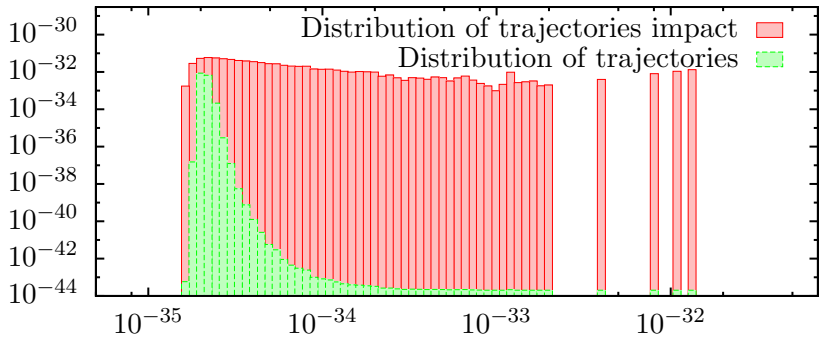

Figure 6. Distribution of trajectories and their contribution

Mapping function. We describe here formally the reduction function f. The reduction function must map each marking of the Petri net to a marking of the reduced Petri net.

First we observe that the signalling cascades SPN contains three places invariants of interest:

- The total number of tokens in the set of places {Raf,Raf_RasGTP,RafP_Phase1,RafP, MEK_RafP, MEKP_RafP} is equal to $4N$.
- The number of tokens in the set of places {MEK, MEK_RafP, MEKP_Phase2, MEKP, MEKP_RafP, MEKPP_Phase2, MEKPP, ERK_MEKPP, ERKP_MEKPP} is equal to $2N$
- The number of tokens in the set of places {ERK, ERK_RafP, ERKP_Phase2, ERKP, ERKP_RafP, ERKPP_Phase2, ERKPP} is equal to $3N$

We also introduce three sequences of places one for each layer of phosphorylation.

- S_1 = [Raf,Raf_RasGTP,RafP_Phase1,RafP]
- S_2 = [MEK, MEK_RafP, MEKP_Phase2, MEKP, MEKP_RafP, MEKPP_Phase2,MEKPP]
- S_3 = [ERK, ERK_RafP, ERKP_Phase2, ERKP, ERKP_RafP, ERKPP_Phase2, ERKPP]

Let us remark that a marking of the SPN $\mathcal{N}$ is uniquely determined by its values on places in S_1, S_2 and S_3.

We define a function g such that: for all positive integer m, positive real number p and vector of integers of size k, $\mathbf{v} = (v_i)_1^k$, $g(p, m, \mathbf{v})$ is the vector of integers of size k, $\mathbf{u} = (u_i)_1^k$, defined by:

$$\forall i > 1,\ u_i = \min\left(\lceil v_i \cdot p \rceil, m - \sum_{l=i+1}^{k} u_l\right) \text{ and } u_1 = m - \sum_{l=2}^{k} u_l$$

One can see that the g is properly defined and that the sum of the components of $\mathbf{u}$ are equal to m.

The reduction function f for the two properties is a mapping from the set of states of SPN $\mathcal{N}$ to the set of states of the reduced SPN $\mathcal{N}^\bullet$. This function takes as input the marking of a sequence of places that uniquely define the state. This sequence can be decompose on the three layers of phosphorylation, that is S_1 for the first layer, S_2 for the second layer and S_3 for the last layer.

Recall that layers are not independent one from the others because proteins of one layer are used to activate the following layer; this can be seen on the P-invariant that contains places of the following layer. The mapping function that we construct preserve these P-invariants.

Roughly, on each layer S_i, this function f applies a function of the form $g(p_i, m_i, -)$.

Precisely, given a scaling factor N and a scaling factor for each of the three layers of the reduced model, respectively N_1, N_2 and N_3, the reduction function f maps the marking m on the marking $m^\bullet$ defined as follow:

- $(m^\bullet(p))_{p\in S_3} = g\left(\frac{N_3}{N}, 3N_3, (m(p))_{p\in S_3}\right)$
- $(m^\bullet(p))_{p\in S_2} = g\left(\frac{N_2}{N}, 2N_2 - m^\bullet(\mathsf{ERK\_MEKPP}) - m^\bullet(\mathsf{ERKP\_MEKPP}), (m(p))_{p\in S_2}\right)$
- $(m^\bullet(p))_{p\in S_1} = g\left(\frac{N_1}{N}, 4N_1 - m^\bullet(\mathsf{MEK\_RafP}) - m^\bullet(\mathsf{MEKP\_RafP}), (m(p))_{p\in S_1}\right)$

One can see that the three P-invariants are preserved in the reduced model by f. We choose $N_1 = N_2 = 1$ and $N_3 = N$.

Experimental analysis of the likelihood. We describe here some technical details of the simulation done for evaluating probability p_1. Recall the likelihood of a trajectory requires the distribution of the random variable W_{s_0} (see subsection 4.1). Proposition 6 of [4] ensures that W_{s_0} takes values in $\{0\} \cup [\mu^\bullet_{n^+}(f(s)), \infty[$. This was proven for DTMCs but can be adapted in a straightforward way for CTMCs. Values taken by L_{s_0} are taken by W_{s_0} when at the end of a successful trajectory, therefore these values are in $[\mu^\bullet_{n^+}(f(s)), \infty[$.

We simulate the system for the first formula with $N = 2$ and a discrete horizon of 615 (615 is the right truncation point given by Fox-Glynn algorithm). The result of the simulation is represented as an histogram shown in Figure 6. The total number of trajectories is 69000, 49001 of them are not successful. We observe that most of the successful trajectories end with a value close to 2.10^{-35}, and that a few trajectories have a value close to 10^{-32}. This is represented by an histogram which is shown as the green part of Figure 6 (with a logarithmic scale for the abscissa). We also represent the histogram of the contribution of the trajectories for the estimation of the mean of L_{s_0}, that is the red part of the figure (with a logarithmic scale for the ordinate). We observe that the contribution to this mean is almost uniform. Thus a trajectory ending with a likelihood close to 10^{-32} have a larger impact than one ending with a likelihood close to $10^3 4$. This means that an estimator of the mean value of $L_{(s_0,u)}$ will underestimate the expectation of $L_{(s_0,u)}$. To produce a framing of the result, one has to use a very conservative method to avoid underestimating the result.

6.2 Conditional maximal signal peak

The network structure of each layer in the signalling cascade presents a cyclic behaviour, i.e. phosphorylated proteins, serving as signal for the next layer, can also be dephosphorylated again, which corresponds to a decrease of the signal strength. Thus an interesting property of the signalling cascade is the probability of a further increase of the signal strength under the condition that a certain strength has already been reached. We estimate this quantity for the first layer in the signalling cascade, i.e. RafP, and ask specifically for the probability to reach its maximal strength, $4N$: "What is the probability of the concentration of RafP to continue its increase and reach $4N$, when starting in a state where the concentration is for the first time at least L?". This is a special use case of the general pattern introduced in [11].

$$p_2 = \Pr_\pi((\mathsf{RafP} \geq L)\ \mathbf{U}\ (\mathsf{RafP} \geq 4N))$$

where π is the distribution over states when satisfying for the first time the state formula $\mathsf{RafP} \geq L$ (previously called a filter).

This formula is parametrized by threshold L and scaling factor N. The results for increasing N and L are reported in Table 5 (confidence intervals are computed by Chernoff-Hoeffding method). As before, MARCIE cannot handle the case $N = 3$, the bottleneck being here the execution time.

It is clear that p_2 is an increasing function of L. More precisely, experiments point out that p_2 increases approximatively exponentially by at least one magnitude order when L is incremented. However this dependency is less clear than the one of the first property.

The reduced model is the one used for the first property except for the values of the following parameters: here we choose $N_1 = 1$, $N_2 = N$ and $N_3 = 0$.

Table 5. Numerical values associated with p_2

N	L	COSMOS		MARCIE		
		confidence interval	time	result	time	memory
2	2	[2.39E-13 , 1.07E-09]	31	5.55E-10	90	802 MB
2	3	[2.18E-10 , 6.92E-08]	110	6.64E-08	136	816 MB
2	4	[9.33E-08 , 3.54E-05]	256	3.01E-06	276	798 MB
2	5	[1.16E-05 , 6.08E-04]	1000	7.16E-05	759	801 MB
2	6	[5.42E-04 , 1.21E-03]	5612	1.27E-03	3180	804 MB
3	5	[1.82E-12 , 9.78E-09]	459	Time > 48 hours		
3	6	[3.41E-10 , 9.66E-08]	1428			
3	7	[1.81E-08 , 2.23E-06]	7067			
3	8	[8.72E-07 , 2.71E-06]	4460			
3	9	[1.42E-06 , 4.59E-05]	4301			
3	10	[2.69E-04 , 9.34E-04]	6420			
4	10	[5.12E-09 , 2.75E-08]	8423	Memory > 32GB		
4	11	[8.23E-08 , 2.97E-07]	7157			
4	12	[9.84E-07 , 1.86E-06]	18730			

6.3 Signal propagation

To demonstrate that the increases of the signals are temporally ordered w.r.t. the layers in the signalling cascade, and by this way proving the travelling of the signals along the layers, we explore the following property: "What is the probability that, given the initial concentrations of RafPP, MEKPP and ERKPP being zero, the concentration of RafPP rises above some level L while the concentrations of MEKPP and ERKPP remain at zero, i.e. RafPP is the first species to react?". While this property has its focus on the beginning of the signalling cascade, it is obvious how to extend the investigation by further properties covering the entire signalling cascade.

$$p_3 = \Pr((\mathsf{MEKPP} = 0) \wedge (\mathsf{ERKPP} = 0))\mathbf{U}(\mathsf{RafPP} > L))$$

This formula is parametrized by L. Due to the lack of space only some values of L in $[0, 4N[$ are reported. The results for increasing N and L are given in Table 6. As can be observed, the probability to satisfy this property is not a rare event thus no importance sampling is required. Instead results are obtained by a plain Monte Carlo simulation generating 10 millions of trajectories. For $N > 3$ MARCIE requires more than $32GB$ of memory thus the computation was stopped. On the other hand, the memory requirement of COSMOS is around $50MB$ for all experiments.

We also observed that as expected the probability exponentially decreases with respect to L.

Table 6. Experiments associated with p_3

N	L	COSMOS		MARCIE		
		confidence interval	time	result	time	memory
2	2	[0.8018,0.8024]	4112	0.8021	75	730MB
2	3	[0.4201,0.4209]	7979	0.4205	137	723MB
2	4	[0.1081,0.1086]	10467	0.1084	163	725MB
2	5	[0.0122,0.0124]	11122	0.0123	123	725MB
2	6	[6.20E-4,6.61E-4]	11185	6.32E-4	129	725MB
2	7	[1.02E-5,1.61E-5]	11194	1.24E-5	156	725MB
3	6	[0.0136,0.0138]	14648	0.0137	17420	10.3GB
3	7	[1.45E-3,1.51E-3]	14752	1.48E-3	18155	10.3GB
3	8	[9.99E-5,1.17E-4]	14739	1.06E-4	18433	10.3GB
3	9	[3.53E-6,7.36E-6]	14734	4.86E-6	18353	10.3GB
3	10	[1.03E-8,9.27E-7]	14743	1.29E-7	18355	10.3GB
3	11	[0 ,5.30E-7]	14766	1.48E-9	18047	10.3GB

N	L	COSMOS	
		confidence interval	time
4	8	[1.47E-3,1.53E-3]	17669
4	9	[1.52E-4,1.73E-4]	17628
4	10	[9.99E-6,1.59E-5]	17656
4	11	[1.54E-7,1.57E-6]	17632
4	12	[0 ,5.30E-7]	17664
5	8	[6.92E-3,7.06E-3]	20367
5	9	[1.13E-3,1.19E-3]	20421
5	10	[1.46E-4,1.67E-4]	20419

7 Conclusion

We have studied rare events in signalling cascades with the help of an improved importance sampling method implemented in COSMOS. As demonstrated by means of our scalable case study, our method has been able to cope with huge

models that could not be handled neither by numerical computations nor by standard simulations. In addition, analysis of the experiments has pointed out interesting dependencies between the scaling parameter and the quantitative behaviour of the model.

In future work we intend to incorporate other types of quantitative properties, such as the mean time a signal needs to exceed a certain threshold, the mean travelling time from the input to the output signal, or the relation between the variation of the enzymes of two consecutive levels. We also plan to analyse other biological systems for which the evaluation of tiny probabilities might be relevant like mutation rates in growing bacterial colonies [9].

References

1. M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with generalized stochastic Petri nets*. John Wiley & Sons, Inc., 1994.
2. L. J. Bain and M. Engelhardt. *Introduction to Probability and Mathematical Statistics, Second Edition*. Duxbury Classic Series, 1991.
3. P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin. HASL: An expressive language for statistical verification of stochastic models. In *Proceedings of the 5th International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'11)*, pages 306–315, Cachan, France, May 2011.
4. B. Barbot, S. Haddad, and C. Picaronny. Coupling and importance sampling for statistical model checking. In C. Flanagan and B. König, editors, *TACAS*, volume 7214 of *Lecture Notes in Computer Science*, pages 331–346. Springer, 2012.
5. B. Barbot, S. Haddad, and C. Picaronny. Importance sampling for model checking of continuous time Markov chains. In Petre Dini and Pascal Lorenz, editors, *Proceedings of the 4th International Conference on Advances in System Simulation (SIMUL'12)*, pages 30–35, Lisbon, Portugal, November 2012. XPS.
6. R. Breitling, D. Gilbert, M. Heiner, and R. Orton. A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. *Briefings in Bioinformatics*, 9(5):404–421, September 2008.
7. V. Chickarmane, B. N. Kholodenko, and H. M. Sauro. Oscillatory dynamics arising from competitive inhibition and multisite phosphorylation. *Journal of Theoretical Biology*, 244(1):68–76, January 2007.
8. B. L. Fox and P. W. Glynn. Computing Poisson probabilities. *Commun. ACM*, 31(4):440–445, 1988.
9. D. Gilbert, M. Heiner, F. Liu, and N. Saunders. Colouring Space - A Coloured Framework for Spatial Modelling in Systems Biology. In JM Colom and J Desel, editors, *Proc. PETRI NETS 2013*, volume 7927 of *LNCS*, pages 230–249. Springer, June 2013.
10. P. W. Glynn and D. L. Iglehart. Importance sampling for stochastic simulations. *Management Science*, 35(11):1367–1392, 1989.
11. M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *SFM 2008*, volume 5016 of *LNCS*, pages 215–264. Springer, 2008.
12. M. Heiner, C. Rohr, and M. Schwarick. MARCIE - Model checking And Reachability analysis done effiCIEntly . In J.M. Colom and J. Desel, editors, *Proc. PETRI NETS 2013*, volume 7927 of *LNCS*, pages 389–399. Springer, 2013.

13. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):pp. 13–30, 1963.
14. A. Jensen. Markoff chains as an aid in the study of markoff processes. *Skand. Aktuarietidskr*, 1953.
15. P. L'Ecuyer, V. Demers, and B. Tuffin. Rare events, splitting, and quasi-Monte Carlo. *ACM Trans. Model. Comput. Simul.*, 17(2), 2007.
16. A. Levchenko, J. Bruck, and P.W. Sternberg. Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. *Proc Natl Acad Sci USA*, 97(11):5818–5823, 2000.
17. G. Rubino and B. Tuffin. *Rare Event Simulation using Monte Carlo Methods*. Wiley, 2009.