Efficient stochastic simulation of systems with multiple time scales via statistical abstraction

Efficient stochastic simulation of systems with multiple time scales via statistical abstraction *

Luca Bortolussi[1 2 3], Dimitrios Milios[4], and Guido Sanguinetti[4 5]

[1] Modelling and Simulation Group, University of Saarland, Germany
[2] Department of Mathematics and Geosciences, University of Trieste
[3] CNR/ISTI, Pisa, Italy
[4] School of Informatics, University of Edinburgh
[5] SynthSys, Centre for Synthetic and Systems Biology, University of Edinburgh

Abstract. Stiffness in chemical reaction systems is a frequently encountered computational problem, arising when different reactions in the system take place at different time-scales. Computational savings can be obtained under time-scale separation. Assuming that the system can be partitioned into slow- and fast- equilibrating subsystems, it is then possible to efficiently simulate the slow subsystem only, provided that the corresponding kinetic laws have been modified so that they reflect their dependency on the fast system. We show that the rate expectation with respect to the fast subsystem's steady-state is a continuous function of the state of the slow system. We exploit this result to construct an analytic representation of the modified rate functions via statistical modelling, which can be used to simulate the slow system in isolation. The computational savings of our approach are demonstrated in a number of non-trivial examples of stiff systems.

1 Introduction

The presence of multiple scales, either temporal, spatial, or organisational, is one of the hallmarks of complexity of biological systems. Multi-scale systems present daunting challenges to their mathematical and computational treatment, as the cost of analysis and simulation is significantly increased. In order to tame such complexity, a common practice is to rely on abstraction techniques, simplifying some scales of the model, yet still capturing relevant features of the dynamics. Examples are the abstraction of the complex intra-cellular state as a finite state automaton, a typical approach to build cell population models, the abstraction of the local dynamics of epidemic spreading in country-level models [12], or the averaging of fast dynamics in enzyme kinetics [13, 6]. The downside of such approaches is that the abstractions that are constructed are non-trivial and model-specific, and often require considerable efforts from the modellers.

In this paper, we explore the idea that model abstraction can be simplified by relying on statistical methodologies which can be learned automatically from

* L.B. is partially supported by EU-FET project QUANTICOL (nr. 600708) and by FRA-UniTS. D.M. and G.S. are supported by the ERC under grant MLCS 306999.

(few) exploratory runs of the models. We focus on the specific sub-problem of multiple-time scales, related to stiffness, a well studied issue but still problematic, especially for stochastic systems. We build upon the two common theoretical frameworks of Quasi-Steady-State (QSSA) [13, 11] and Quasi-Equilibrium (QE) [2] for stochastic models of chemical reaction networks. These approaches provide recipes to construct abstracted models, by decomposing a model in a fast and a slow subsystems (more time scales can be considered, but this generalisation is not considered here for simplicity). The fast subsystem is assumed to equilibrate at a time scale which is much faster than the characteristic time scale of the slow subsystem, hence it is abstracted by averaging out fast variables according to their equilibrium distribution, conditional on a fixed state of the slow subsystem. This averaging is performed on the kinetic rate functions of the slow subsystem. This theoretical recipe can produce accurate results, when the QSSA or QE assumptions are satisfied, yet it is very hard to obtain analytical expressions for the kinetic rates of the slow subsystem, which hinders its use in practice.

In this paper, we propose a method to circumvent the problem by exploiting ideas from machine learning, in particular Gaussian Processes [14], to learn the abstracted slow kinetic rates, as a function of slow variables. This approach allows us to construct statistical surrogates of the reduced rate functions in a fully automatic and computationally cheap way, without analytical efforts from the modeller side. It relies only on continuity properties of slow rates, which are also investigated in the paper. Such statistical abstraction of the slow model can then be used to perform simulation efficiently. In the paper we present the novel simulation algorithm, and assess its performance with respect to other slow scale simulation methods proposed in literature. Furthermore, our approach has another advantage: using the same learning strategy, and at a mild additional preprocessing cost, we can additionally learn slow rates as a function of some model parameters, enabling efficient parameter exploration in the stiffness regime.

The paper is organised as follows: in Section 2, we introduce the relevant background material and related work, as well as the QSSA and QE model reduction strategies. The continuity results and the statistical abstraction procedure, together with the resulting simulation algorithm, are presented in Section 4. Section 5 contains the experimental validation of the proposed approach, while final comments are discussed in Section 6. Throughout the paper we will use a simple enzyme-substrate model as a running example.

2 Background & Related Work

Chemical Reaction Networks. We will describe biochemical systems using the widespread formalism of (bio)Chemical Reaction Networks (CRN). The main entities involved are species and reactions.

- Each species represents a molecule described in the model; the vector $\boldsymbol{X}(t) = (X_1(t), \ldots, X_n(t)) \in S \subseteq \mathbb{N}^n$ counts the number of molecules of each species in the system at time t.

– Reactions describe how the system state can change. Each reaction is of the form

$$r_1 X_1 + \dots r_n X_n \xrightarrow{f(\boldsymbol{X})} s_1 X_1 + \dots s_n X_n,$$

where the left hand side represent molecules that are consumed by the reaction, the right hand side describe which molecules are created, and $f(\boldsymbol{X})$ is the kinetic rate function, giving the speed of the reaction as a function of the system state. For each reaction R_j, we can define the vector $\boldsymbol{r}_j$ (respectively $\boldsymbol{s}_j$), encoding how many agents are consumed (respectively produced) in the reaction, so that $\boldsymbol{v}_j = \boldsymbol{s}_j - \boldsymbol{r}_j$ gives the net change of species.

We will consider the stochastic interpretation of biochemical reaction networks [9], in which the dynamics of the system is described by a Continuous-Time Markov Chain (CTMC), a Markovian (i.e. memoryless) stochastic process defined on a finite or countable state space S and evolving in continuous time [8]. In general, we can think of CTMCs as a collection of random variables $\boldsymbol{X}(t)$ on the state space S, indexed by time $t \in [0, \infty)$.

Molecular systems described by CRN are located in a finite volume V, and one can reason on concentrations, rather than molecule numbers, by dividing variables by the volume V. We will indicate with capital letters $\boldsymbol{X}$ the molecular numbers and with small letters $\boldsymbol{x} = \boldsymbol{X}/V$ the concentrations. Rate functions can be expressed either in terms of molecular numbers or concentrations, modulo a rescaling of parameters [9]. We will denote with $f_j(\boldsymbol{X})$ and $f_j(\boldsymbol{x})$ the same rate function, expressed in molecular numbers or concentrations, respectively.

For most CRNs, it is impossible or prohibitively expensive to numerically solve the underlying CTMC directly, so it is a common practice to explore the system's behaviour via stochastic simulation. The standard simulation approach is known as the *Gillespie algorithm* [9], and it is exact in the sense that it simulates every single reaction event happening.

Running example - Part I. We demonstrate the main concepts of the paper on a simple enzyme-substrate model [13]. The system state is represented as a vector $\boldsymbol{X} = (X_E, X_S, X_{ES}, X_P)$ that denotes the populations for an enzyme E, a substrate S, the complex ES formed by the combination of the enzyme with the substrate, and a product P. The state can be changed by the reactions:

$$\begin{aligned} E + S &\xrightarrow{f_1(\boldsymbol{X})} ES, & f_1(\boldsymbol{X}) &= c_1 X_E X_S \\ ES &\xrightarrow{f_2(\boldsymbol{X})} E + S, & f_2(\boldsymbol{X}) &= c_2 X_{ES} \\ ES &\xrightarrow{f_3(\boldsymbol{X})} E + P, & f_3(\boldsymbol{X}) &= c_3 X_{ES} \end{aligned} \tag{1}$$

Related work. The approach to model reduction exploiting time scale separation presented here falls within the scope of Quasi-Steady-State Approximation (QSSA) for stochastic models [13, 10, 6, 5, 4, 3]. In these approaches, species are partitioned into fast and slow, and transitions are separated accordingly. Then, the fast system, conditional on the slow one, is averaged away assuming it is at steady state. The issue with all these approaches is that they require a-priori

identification of fast and slow species, which is usually a choice left to intuition of the modeller. A similar approach, known as Quasi-Equilibrium [2], instead, starts by partitioning the transitions into fast and slow, and then separating species, possibly defining new species by taking a linear combination of the original ones. In both cases, the so obtained system satisfies the decomposition discussed in this section, hence our simulation algorithm can be applied.

A common characteristic of these earlier works on quasi-equilibrium reduction is that they rely on model-dependent expressions to calculate or approximate the rate expectations of the slow reactions, de facto limiting the applicability of the derived simulation algorithms [4–6]. In this work, we investigate the potential of automatically learning these expectations using a regression technique. Under the quasi-equilibrium assumption, our approach relies on no more assumptions regarding the form or the structure of the fast subsystem.

A generic approach to approximate the rate expectation for the slow reactions is prescribed in [15], where a *Nested Stochastic Simulation Algorithm* (Nested-SSA) is proposed to approximate the steady-state of the fast subsystem. We have implemented Nested-SSA following its description in the original paper, in order to produce some comparative results. The step parameter for Nested-SSA has been explored experimentally such that the efficiency of the two approximate simulation approaches has been roughly the same, in order to perform a fair comparison in terms of approximation quality. Another approach related to Nested-SSA has been recently proposed in [16].

3 Quasi-Equilibrium Reduction

Gillespie's exact simulation approach can have high computational costs in presence of stiffness, where a small number of reactions dominate computations. We will now introduce an approach to address such problems by partitioning the system in two separate subsystems with different time-scales. We will first discuss how to construct the reduced model, and then comment on how such fast and slow subsystems can be identified. We will make some strong assumptions on the structure, commenting later on how to relax them.

Partition of species and reactions. We assume that species $\boldsymbol{X} = X_1, \ldots, X_n$ of the system can be partitioned in two disjoint subsets: fast species, denoted by $\boldsymbol{Y} = Y_1, \ldots, Y_m$, and slow species, indicated with $\boldsymbol{Z} = Z_1, \ldots, Z_s$, with $m + s = n$. Hence, the state space S is decomposed into the fast S_f and the slow S_s subspaces, so that $S = S_f \times S_s$. We will use this notation consistently in rates, writing $f_j(\boldsymbol{Y}, \boldsymbol{Z})$ in place of $f_j(\boldsymbol{X})$.

Similarly, we assume that the set $\mathcal{R}$ of reactions is also partitioned into fast and slow subsets, denoted respectively $\mathcal{R}_{fast}$ and $\mathcal{R}_{slow}$. The idea is that fast reactions act only on fast variables (i.e. for each $R_j \in \mathcal{R}_{fast}$, $\boldsymbol{v}_j$ is zero in correspondence to slow variables), and quickly bring the fast subsystem to equilibrium. Hence, the evolution of slow variables will essentially sense the fast system only via its steady state distribution. Slow reactions, instead, can modify both fast and slow subsystems.

Reduced model. Given a partition of species and reactions into fast and slow classes, we can construct the fast and the slow subsystems. The fast subsystem is defined conditionally on a fixed value of the slow variables $\boldsymbol{Z}$. It is a CRN with species $\boldsymbol{Y} = Y_1, \ldots, Y_m$ and reactions $\mathcal{R}_{fast}$. In particular, the rate functions of reactions in $\mathcal{R}_{fast}$ are computed by instantiating the slow variables with their fixed value. Here we assume that such kinetic rate functions depend on slow variables via their concentration, $f_j = f_j(\boldsymbol{Y}, \boldsymbol{z})$, hence the fast subsystem will be parameterised by the concentration $\boldsymbol{z}$ of slow species, which can take values in $\mathbb{R}^s_{\geq 0}$ or on a compact subset, if the state space S_s is finite. This dependency will be made explicit in the notation $\boldsymbol{Y}_{|\boldsymbol{z}}$.

At this stage, we need to make a crucial assumption for the method to work, namely that the conditional fast process $\boldsymbol{Y}_{|\boldsymbol{z}}(t)$ is an *irreducible* and *positive recurrent* CTMC on the fast subspace S_f, for any value of $\boldsymbol{z}$. This will guarantee *existence and uniqueness of the steady state distribution* $\boldsymbol{Y}_{|\boldsymbol{z}}(\infty)$ of $\boldsymbol{Y}_{|\boldsymbol{z}}(t)$. In the following, we will denote the conditional expectation of a function $f(\boldsymbol{Y}, \boldsymbol{z})$, with respect to the steady state distribution $\boldsymbol{Y}_{|\boldsymbol{z}}(\infty)$ of the conditional fast process by $\mathbb{E}_{|\boldsymbol{z}}[f(\boldsymbol{Y}, \boldsymbol{z})]$, to stress the fact that this will be a function of the concentration of slow species.

The slow subsystem, instead, is a CRN on the slow species $\boldsymbol{Z}$, with dynamics given by the slow reactions $\mathcal{R}_{slow}$ only. However, all reactions R_j in $\mathcal{R}_{slow}$ are modified by

1. removing fast species from the left and right hand side of the rule of R_j,[6]
2. replacing the rate function $f_j(\boldsymbol{Y}, \boldsymbol{z})$ by $\hat{f}_j(\boldsymbol{z}) = \mathbb{E}_{|\boldsymbol{z}}[f_j(\boldsymbol{Y}, \boldsymbol{z})]$, i.e. averaging out fast variables with respect to the steady state distribution of fast species, conditional on a given concentration of slow species.

Running example: Part II. In the enzyme-substrate example, stiffness can easily arise if we assume that $c_1, c_2 \gg c_3$. In that case, the reactions in (1) can be partitioned into fast and slow subsets $\mathcal{R}_{fast} = \{R_1, R2\}$ and $\mathcal{R}_{slow} = \{R_3\}$ correspondingly. Consequently, we have fast species $\boldsymbol{Y} = (X_E, X_S, X_{ES})$ and slow species $\boldsymbol{Z} = (X_P)$. We therefore obtain the following fast subsystem:

$$\begin{aligned} E + S &\xrightarrow{f_1(\boldsymbol{Y},\boldsymbol{z})} ES, \qquad f_1(\boldsymbol{Y}, \boldsymbol{z}) = c_1 X_E (N - X_{ES} - X_P) \\ ES &\xrightarrow{f_2(\boldsymbol{Y},\boldsymbol{z})} E + S, \quad f_2(\boldsymbol{Y}, \boldsymbol{z}) = c_2 X_{ES} \end{aligned} \tag{2}$$

where N is a constant that denotes the total enzyme/substrate population in the system; in this way, the dependency on the slow system is reflected in the reaction rates. The slow subsystem is then described by the following reactions:

$$\emptyset \xrightarrow{\hat{f}_3(\boldsymbol{z})} P, \quad \hat{f}_3(\boldsymbol{z}) = \mathbb{E}_{|\boldsymbol{z}}[f_3(\boldsymbol{Y}, \boldsymbol{z})] \tag{3}$$

[6] This is a technically sound operation, as the fast subsystem has a unique steady state distribution, depending only on the state $\boldsymbol{z}$ of the slow subsystem, which is reached immediately after the firing of a slow reaction.

4 Approximation of Rate Expectations

4.1 Continuity of rates of the slow system

We start by proving a crucial property for our method, namely that the rate functions of the reduced slow subsystem are continuous as a function of the concentration of slow species, taking values on the whole $\mathbb{R}^s_{\geq 0}$ (or on a compact connected subset). This property is a consequence of mild regularity properties of the original kinetic rate functions, and is captured by the following theorem, whose proof can be found in the appendix.

Theorem 1. *Let $f(\boldsymbol{Y}, \boldsymbol{z})$ be a locally Lipschitz continuous function w.r.t. (normalised) slow variables. Assume that the fast process, conditional on a fixed concentration $\boldsymbol{z}$ of the slow variables, is irreducible and positive recurrent for each $\boldsymbol{z}$. Then $\mathbb{E}_{|\boldsymbol{z}}[f(\boldsymbol{Y}, \boldsymbol{z})]$ is a continuous function of $\boldsymbol{z}$.* □

Theorem 1 enables us to use powerful techniques based on statistical emulation, which will be discussed in the following subsection, and which are the key of our simulation algorithm.

4.2 Exploring rate expectation via pre-simulation runs

As discussed in Section 3, for many systems exhibiting time-scale separation, it is possible to obtain a good approximation of the system by introducing an auxiliary system where the time scales are separated. Hence, the slow variables are treated as *statistically independent* random variables from the fast variables, and the time-scale separation is equivalent to a mean-field approximation which replaces the true transition rates of the slow variables (which in general depend on the actual fast variables) with their averages with respect to the equilibrium distribution of the fast variables. While this approximation in principle offers huge computational savings, in practice for most systems the equilibrium distribution of the fast variables cannot be computed analytically, and its expectation can consequently be computed only from a set of simulations. Furthermore, in most cases the statistics of the equilibrium distribution of the fast variables will themselves depend on the slow variables. This feedback mechanism engenders stiffness which effectively negates the computational benefits of time-scale separation: for every simulation step in the slow variables, a whole (large) set of complete simulations for the fast variables must be executed to obtain reliable estimates of the equilibrium statistics of the fast variables.

A possible solution to this computational problem would be to explore the functional dependency of the equilibrium statistics of the fast variables on the state of the slow variables. This in principle would greatly facilitate computations, replacing the need for simulations of the fast variables with a lookup table for the statistics. However, in general the number of states visited by slow variables may be very high, resulting in a need for very long precomputing steps. To obviate this problem, we exploit the results of Section 4.1, which imply that the equilibrium statistics of the fast variables are a continuous function of the

slow variables (rescaled to concentrations). This enables us to leverage powerful machine learning techniques to construct a statistical approximation to the equilibrium statistics from a potentially much smaller number of pre-simulation runs. We use Gaussian Processes (GP) regression, a flexible non-parametric Bayesian method for non-linear regression, although other methods are also possible in principle. GPs provide us with a fast analytical approximation to the unknown function from a set of precomputed values of the function; importantly, their flexibility guarantees that they can approximate arbitrarily well any continuous function [1]. We refer the reader to [14] for a comprehensive introduction to GP regression, which we do not provide for space reasons.

4.3 Stochastic simulation via statistical abstraction

We propose a stochastic simulation algorithm via statistical abstraction (SA-SSA), which involves simulating the slow system only. The algorithm works in two phases. In an initialisation phase, we construct an analytical approximation of the rates of the slow subsystem. In the simulation phase, these approximate rates are used in place of the true slow kinetic rate functions to simulate the slow subsystem with standard Gillespie simulation [9]. As the simulation phase is standard, we shall focus on the first phase.

The construction of these analytic approximations during the *initialisation process* is broken down to two steps. The first step involves estimating the rate expectations $\hat{f}_j(\boldsymbol{z}), \forall R_j \in \mathcal{R}_{slow}$ for a grid of n population vectors, which correspond to n different states of the slow process. For each population vector, the fast subsystem is simulated until steady-state is reached, and the expectation of $f_j(\boldsymbol{Y}, \boldsymbol{z})$ is calculated as follows:

$$\hat{f}_j(\boldsymbol{z}) = 1/t_f \int_{t_0}^{t_0+t_f} f_j(\boldsymbol{Y}, \boldsymbol{z}) dt \tag{4}$$

where t_0 is the time required to reach equilibrium and t_f is sufficiently large to compute accurately the time average. This is estimated using a simple heuristic: the rate expectation is measured for regular subsequent time intervals, and steady-state is considered to have been reached if the change observed is less than 1%. Since we have assumed that the fast process is ergodic, there should be exactly one steady-state distribution, therefore the expectation can be calculated using a single trajectory for each of the n states. We stress that our approach is independent of the choice of the method to estimate the steady state, which can be safely replaced.

At the end of this pre-simulation process, we have a collection of n population vectors paired with n noisy observations of the rate expectation as a function of the state of the slow system. GP regression is a natural and fully automated choice to obtain estimations *for the the expectations for any point in the state-space*, since it transfers information across neighbouring points.

To comment on the cost of the initialisation process, we have to consider the cost of the pre-simulation runs and the regression step. One of the main

assumptions of QE reduction is that steady-state is reached quickly for the fast subsystem, therefore pre-simulation avoids the excessive simulation of the fast system that occurs when stiffness is present. The cost of regression is dominated by the solution of a linear system, whose complexity is $O(n^2)$, where n is the number of training points[7]. This cost can be further reduced by employing sparse approximations to GPs, which is a subject well studied in the machine-learning community [14]. An important note on the initialisation cost is that it has to be paid only once, and then name trajectories can be efficiently sampled from the slow subsystem. If the rate expectations are learned as a function of the system parameters as well, then it is possible to approximate an entire family of stiff systems. The relationship between the initialisation cost and the computational savings achieved is demonstrated in the experiments of Section 5.

5 Experimental evaluation

In order to demonstrate the computational savings and assess the approximation quality of our approach, we consider two stiff examples of bio-chemical reaction networks. We have generated samples from the distributions of the slow species, using both the standard Gillespie algorithm [9] and SA-SSA. The approximation quality is evaluated by in terms of the *histogram distance* between the samples from the exact and the approximate simulation process. To put the histogram distance in a context, this has to be compared with the corresponding self-distance. A distance value smaller than the self-distance implies that the two distributions are practically indistinguishable for a given number of samples. The self-distance is estimated using the following result of Cao & Petzold [7]: an upper bound for the average histogram self-distance is given by $\sqrt{(4K)/(\pi N)}$, for N samples and K intervals in the histogram. For the examples that follow, we consider $K = 50$.

5.1 Stiff enzyme-substrate reaction

We perform numerical experiments on the enzyme-substrate example, given the partitioning described by Equations (2) and (3). We consider kinetic constants $c_1 = 0.01$, $c_2 = 1$ and $c_3 = 10^{-4}$, and initial state $\boldsymbol{X}_0 = (220, 3000, 0, 0)$. The rate expectation for R_3 in (3) has been approximated via GP regression. For the training set, we have sampled 1000 population values for the slow variable P between 0 and 3000.

The results of simulating the slow subsystem can be seen in Table 1, which summarises the histogram distances from the true distribution for the population of P, at four time-points. Most of the distances recorded are lower than the estimated upper bound for the average self-distance (i.e. 0.252). We also report the corresponding histogram distances for the Nested-SSA method of Weinan et

[7] GP regression typically involves matrix inversion, but this can be avoided as we make no use of predictive variances.

al [15], which was parametrised so that it has been as efficient as out method (see Table 2). For the given level of efficiency, our method resulted in lower values for the histogram distance in most cases. Most importantly, the simulation strategy that we propose has been significantly more efficient that exact Gillespie simulation, as can be seen in Table 2. We also report the time required for initialisation, which is broken down to pre-simulation runs, hyperparameter optimisation, and the training of the GP regression model.

Table 1. Enzyme-substrate model: histogram distances for 10^3 simulation runs (estimated self-distance: 0.252).

	P	
Time	Nested-SSA	SA-SSA
5×10^4	0.290	0.246
10×10^4	0.250	0.204
18×10^4	1.016	0.160
20×10^4	0.940	0.142

Parameter Exploration. We demonstrate an example of learning the expected rates as a function of the slow state in combination with a parameter of the system. This practice allows us to pay the initialisation cost once and then simulate a range of stiff systems using our accelerated simulation approach. For the enzyme-substrate system we consider that c_1 varies in the range $[0.01, 1]$; note that for the values of c_1 considered, the system remains stiff, so the QE reduction is meaningful. We have randomly sampled a grid of 1000 values for $X_P \in [0, 3000]$ and $c_1 \in [0.01, 1]$, which was used as training set for a regression model. By fixing the parameter c_1 to a particular value, we were able to generate trajectories efficiently using SA-SSA. Table 3 summarises the relative mean error observed when approximating the mean value of X_P, for different values of c_1. The total initialisation time for our approach has been 3.562 sec. Parameter exploration via the standard Gillespie algorithm required 1911 sec, while SA-SSA required only 32 sec.

5.2 Viral infection model

We now consider is the viral infection model appeared in [11]. We present the following simplified version of the model which involves three species, the viral template T, the viral genome G, and the viral structural protein S:

$$\begin{array}{llll} T \xrightarrow{f_1(\boldsymbol{X})} G + T, & f_1(\boldsymbol{X}) = k_1 X_T & T \xrightarrow{f_4(\boldsymbol{X})} \emptyset, & f_4(\boldsymbol{X}) = k_4 X_T \\ G \xrightarrow{f_2(\boldsymbol{X})} T, & f_2(\boldsymbol{X}) = k_2 X_G & S \xrightarrow{f_5(\boldsymbol{X})} \emptyset, & f_5(\boldsymbol{X}) = k_5 X_S \\ T \xrightarrow{f_3(\boldsymbol{X})} S + T, & f_3(\boldsymbol{X}) = k_3 X_T & G + S \xrightarrow{f_6(\boldsymbol{X})} V, & f_6(\boldsymbol{X}) = k_6 X_G X_S \end{array} \tag{5}$$

The system state is represented as a vector $\boldsymbol{X} = (X_T, X_G, X_S)$. Regarding the model parameters, we follow [11]; for the kinetic constants we have: $k_1 = 1$,

Table 2. Execution times in seconds for 10^3 simulation runs.

Method		Enzyme-substrate	Viral model
SA-SSA	Pre-simulation	0.291	26.11
	Hyperparam. opt.	1.484	1.68
	Training	0.080	0.05
	Total initialisation	1.855	27.84
	Simulation	153	316
	Exact SSA	6947	2410
	Nested-SSA	209	327

Table 3. Relative mean error values for approximating the mean value of X_P, for 10^3 simulation runs.

	P (RME)			
Time	$c_1 = 0.01$	$c_1 = 0.1$	$c_1 = 0.5$	$c_1 = 1$
5×10^4	1.83×10^{-3}	9.08×10^{-4}	2.35×10^{-3}	2.17×10^{-3}
10×10^4	1.20×10^{-3}	1.49×10^{-3}	1.94×10^{-3}	2.87×10^{-3}
18×10^4	8.04×10^{-4}	3.73×10^{-5}	4.49×10^{-4}	3.05×10^{-4}
20×10^4	9.13×10^{-4}	4.56×10^{-5}	6.06×10^{-5}	3.26×10^{-5}

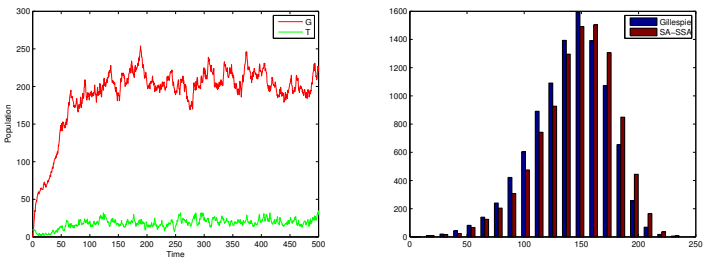

Fig. 1. Left: A random trajectory of the viral infection model, showing the slow species populations. Right: Distribution of the genome population X_G at $t = 50$.

$k_2 = 0.025$, $k_3 = 1000$, $k_4 = 0.25$, $k_5 = 1.9985$ and $k_6 = 7.5e-6$, and initial state $\boldsymbol{X}_0 = (10, 0, 0)$. A random system trajectory can be seen in Figure 1.

Based on the kinetic constants, we consider the set of fast reactions $\mathcal{R}_{fast} = \{R_3, R_5\}$ and slow reactions $\mathcal{R}_{slow} = \{R_1, R_2, R_4, R_6\}$. Therefore, the fast species will be $\boldsymbol{Y} = (X_S)$, and we have slow species $\boldsymbol{Z} = (X_G, X_T)$, give rise to the following fast and slow subsystems correspondingly:

$$\emptyset \xrightarrow{f_3(\boldsymbol{Y},\boldsymbol{z})} S, \quad f_3(\boldsymbol{Y},\boldsymbol{z}) = k_3 X_T$$
$$S \xrightarrow{f_5(\boldsymbol{Y},\boldsymbol{z})} \emptyset, \quad f_5(\boldsymbol{Y},\boldsymbol{z}) = k_5 X_S$$

$$T \xrightarrow{f_1(\boldsymbol{z})} G + T, \quad f_1(\boldsymbol{z}) = k_1 X_T$$
$$G \xrightarrow{f_2(\boldsymbol{z})} T, \quad f_2(\boldsymbol{z}) = k_2 X_G$$
$$T \xrightarrow{f_4(\boldsymbol{z})} \emptyset, \quad f_4(\boldsymbol{z}) = k_4 X_T$$
$$G \xrightarrow{\hat{f}_6(\boldsymbol{z})} V, \quad \hat{f}_6(\boldsymbol{z}) = \mathbb{E}_{|\boldsymbol{z}}[f_6(\boldsymbol{Y},\boldsymbol{z})]$$

The rate of R_6 originally depends on X_G directly, and on X_T indirectly, since the population of T affects the steady-state of the fast process. We consider a random grid of 256 uniformly distributed population values for the genome G and the template T, given upper bounds of 500 and 100 molecules correspondingly. Note that a naïve exploration of the rate expectation would require 50000 evaluations, while we use only 256 for the training set of the GP.

The performance in terms of accuracy for the viral model is summarised in Table 4. We report the histogram distances for slow components, at four time-points. An example of the histograms generated can be seen in Figure 1 for the genome G, at time $t = 50$. We see that in all cases the distance from the true distribution is very close to the self-distance estimated for the given number of samples, a fact that implies a very good approximation of the stochastic properties for the slow system. The computational savings are also significant, as can be seen in Table 2.

Table 4. Viral infection model: histogram distances for 10^3 simulation runs (estimated self-distance: 0.252).

	G		T	
Time	Nested-SSA	SA-SSA	Nested-SSA	SA-SSA
50	0.988	0.308	0.548	0.242
100	0.244	0.414	0.154	0.226
200	0.388	0.406	0.156	0.204
500	0.346	0.432	0.198	0.238

6 Conclusions

Time-scale separation is a well studied approach to efficiently simulate systems that exhibit stiffness, where systems are partitioned into slow and fast subsystems. Nevertheless, most of the approaches proposed in the literature rely on the structure of the system to produce estimations for the rate expectations for the slow process. We have proposed SA-SSA as a generic approach to simulate the slow-scale subsystems, where these rate expectations are approximated via a machine learning method.

Experiments on examples of stiff systems show that SA-SSA requires a small initialisation cost and results in significant computational savings. For a given level of efficiency, SA-SSA achieved similar or better accuracy than Nested-SSA, whose premise is also a generic simulation framework for stiff systems. Besides any performance comparison, there is a qualitative difference between the two methods. Unlike Nested-SSA, our approach is not transparent with respect to the slow process, since it requires a rough estimate of the reachable state-space. However, the efficiency of SA-SSA is not affected by the complexity of the fast subsystem, in contrast with Nested-SSA, as any relevant cost in only paid during the initialisation phase. Moreover, it has been possible to learn the rate

expectations as functions of the model parameters as well; we therefore obtain approximations for a family of systems, provided that these comply with the stiffness assumption.

References

1. L. Bortolussi, D. Milios, and G. Sanguinetti. Smoothed model checking for uncertain continuous time Markov chains. *CoRR ArXiv*, 1402.1450, 2014.
2. L. Bortolussi and R. Paškauskas. Mean-field approximation and quasi-equilibrium reduction of markov population models. 2014.
3. M. Bruna, S. J. Chapman, and M. J. Smith. Model reduction for slowfast stochastic systems with metastable behaviour. *The Journal of Chemical Physics*, 140(17), 2014.
4. Y. Cao, D. T. Gillespie, and L Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *Journal of Computational Physics*, 206(2):395–411, 2005.
5. Y. Cao, D. T. Gillespie, and L. R Petzold. The slow-scale stochastic simulation algorithm. *The Journal of Chemical Physics*, 122(1):14116, 2005.
6. Y. Cao, D. T. Gillespie, and L.R. Petzold. Accelerated stochastic simulation of the stiff enzyme-substrate reaction. *The Journal of Chemical Physics*, 123(14):144917–12, 2005.
7. Y. Cao and L. Petzold. Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems. *Journal of Computational Physics*, 212(1):6–24, 2006.
8. R. Durrett. *Essentials of Stochastic Processes*. Springer, May 2012.
9. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. of Physical Chemistry*, 81(25), 1977.
10. J. Goutsias. Quasi-equilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *The Journal of Chemical Physics*, 122(18), 2005.
11. E. L. Haseltine and J. B. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *The Journal of Chemical Physics*, 117(15):6959, 2002.
12. L. Mari, E. Bertuzzo, L. Righetto, R. Casagrandi, M. Gatto, I. Rodriguez-Iturbe, and A. Rinaldo. Modelling cholera epidemics: the role of waterways, human mobility and sanitation. *Journal of The Royal Society Interface*, 9(67):376–388, 2011.
13. C. V. Rao and A. P. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *The Journal of Chemical Physics*, 118(11):4999, 2003.
14. C. E. Rasmussen and C. K. I Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge, Mass., 2006.
15. E. Weinan, D. Liu, and E. Vanden-Eijnden. Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates. *The Journal of Chemical Physics*, 123(19), 2005.
16. C. Zechner and H. Koeppl. Uncoupled analysis of stochastic reaction networks in fluctuating environments. *PLoS comp. bio.*, 10(12), 2014.