

Research Article

Bayesian Computation Methods for Inference in Stochastic Kinetic Models

Eugenia Koblents,¹ Inés P. Mariño ,^{2,3,4} and Joaquín Míguez ⁵

¹United Technologies Research Center Ireland, Ltd., Machine Learning Group, 4th Floor Penrose Building, Penrose Wharf, Cork City (Co. Cork), Ireland

²Department of Biology and Geology, Physics and Inorganic Chemistry, Universidad Rey Juan Carlos, C/Tulipán s/n, 28933 Móstoles, Madrid, Spain

³Institute for Women's Health, University College London, London WC1E 6BT, UK

⁴Research Laboratory Systemic Medicine of Healthy Ageing, Institute of Biology and Medicine, National Research Lobachevsky State University of Nizhny Novgorod, 603950 Nizhny Novgorod, Russia

⁵Department of Signal Theory & Communications, Universidad Carlos III de Madrid, Avenida de la Universidad 30, 28911 Leganés, Madrid, Spain

Correspondence should be addressed to Inés P. Mariño; ines.perez@urjc.es

Received 22 June 2018; Accepted 3 January 2019; Published 20 January 2019

Academic Editor: Enzo Pasquale Scilingo

Copyright © 2019 Eugenia Koblents et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper we investigate Monte Carlo methods for the approximation of the posterior probability distributions in stochastic kinetic models (SKMs). SKMs are multivariate Markov jump processes that model the interactions among species in biological systems according to a set of usually unknown parameters. The tracking of the species populations together with the estimation of the interaction parameters is a Bayesian inference problem for which Markov chain Monte Carlo (MCMC) methods have been a typical computational tool. Specifically, the particle MCMC (pMCMC) method has been shown to be effective, while computationally demanding method applicable to this problem. Recently, it has been shown that an alternative approach to Bayesian computation, namely, the class of adaptive importance samplers, may be more efficient than classical MCMC-like schemes, at least for certain applications. For example, the nonlinear population Monte Carlo (NPMC) algorithm has yielded promising results with a low dimensional SKM (the classical predator-prey model). In this paper we explore the application of both pMCMC and NPMC to analyze complex autoregulatory feedback networks modelled by SKMs. We demonstrate numerically how the populations of the relevant species in the network can be tracked and their interaction rates estimated, even in scenarios with partial observations. NPMC schemes attain an appealing trade-off between accuracy and computational cost that can make them advantageous in many practical applications.

1. Introduction

Stochastic kinetic models (SKMs) are multivariate systems that model interactions among species in ecological, biological, and chemical problems, according to a set of unknown rate parameters [1]. The aim of this paper is to study computational methods for the approximation of the posterior distribution of the rate parameters and the populations of all species, provided a set of discrete, noisy observations is available. This inference problem has been traditionally addressed in a Bayesian framework using Markov chain Monte Carlo

(MCMC) schemes [1–4]. In [5], a particle MCMC (pMCMC) method [6] has been applied to Lotka-Volterra and prokaryotic autoregulatory models. The pMCMC technique relies on a sequential Monte Carlo (SMC) approximation of the posterior distribution of the populations to compute the Metropolis-Hastings (MH) acceptance ratio.

However, MCMC methods in general and pMCMC in particular suffer from a number of problems. The convergence of the Markov chain is hard to assess and the final set of samples presents correlations which can greatly reduce its efficiency. Besides, MCMC methods do not (easily) allow for

parallel implementation and turn out to be computationally intensive, even for relatively simple models. A classical technique to reduce the complexity of the existing MCMC methods when applied to SKMs is the so-called diffusion approximation of the underlying stochastic jump process [7]. However, the diffusion approximation breaks down in low-concentration scenarios [5]. Other techniques, including linear noise approximations [8–11] and Gaussian-process estimators of the likelihood [12], have also been proposed in the past few years to reduce the computational cost. The parameters of the MCMC proposal are also hard to choose and they determine the performance of the algorithm.

An appealing alternative to MCMC methods is the class of the population Monte Carlo (PMC) algorithms [13]. PMC is an adaptive importance sampling (IS) scheme [14] that yields a discrete approximation of a target probability distribution. The PMC algorithm has important advantages with respect to MCMC techniques. It provides independent samples and asymptotically unbiased estimates at all iterations, which avoids the need of a convergence period. Additionally, PMC may be easily parallelized.

On the other hand, the main weakness of IS and PMC is their low efficiency in high dimensional problems, due to the well-known degeneracy problem [15]. The recently proposed nonlinear PMC (NPMC) scheme [16, 17] mitigates this difficulty by computing nonlinear transformations of the importance weights (IW), in order to smooth their variations and avoid degeneracy. In [17] a rigorous proof of convergence of the nonlinear importance sampling scheme as the number of Monte Carlo samples increases is presented. Similarly to the pMCMC method in [5], the NPMC algorithm resorts to an SMC approximation of the posterior distribution of the populations to compute the IWs.

In this paper we apply the NPMC method to the estimation of both the parameters and the unobserved populations in a complex SKM modelling an autoregulatory feedback network with several species and noisy observations. We present numerical results to compare the performance of the state-of-the-art pMCMC and the proposed NPMC in two scenarios of different dimension and with two different observation models. We show that the NPMC method outperforms the pMCMC method for the same computational cost.

The rest of the paper is organized as follows. In Section 2 we present an introduction to the basics of SKMs and the usual solutions to this Bayesian inference problem. In Sections 3 and 4 we describe the pMCMC and NPMC methods, respectively, when applied to the approximation of posterior distributions in SKMs. In Section 5 we numerically compare the performance of pMCMC and NPMC schemes when applied to an autoregulatory feedback network, an SKM that can be interpreted as a generalization of the paradigmatic Lotka-Volterra model. Finally, Section 6 is devoted to the conclusions.

2. Bayesian Inference for Stochastic Kinetic Models

2.1. Stochastic Kinetic Models. A SKM is a multivariate continuous-time jump process modelling the interactions among

different species or molecules in ecological, biological, and chemical systems [1].

Consider an ecological interaction network that describes the time evolution of the population of V species x_1, \dots, x_V related by means of K rate equations r_1, \dots, r_K

$$\begin{aligned} r_1 : & \quad p_{11}x_1 + p_{12}x_2 + \dots + p_{1V}x_V \\ & \quad \xrightarrow{c_1} q_{11}x_1 + q_{12}x_2 + \dots + q_{1V}x_V, \\ r_2 : & \quad p_{21}x_1 + p_{22}x_2 + \dots + p_{2V}x_V \\ & \quad \xrightarrow{c_2} q_{21}x_1 + q_{22}x_2 + \dots + q_{2V}x_V, \\ & \quad \vdots \\ r_K : & \quad p_{K1}x_1 + p_{K2}x_2 + \dots + p_{KV}x_V \\ & \quad \xrightarrow{c_K} q_{K1}x_1 + q_{K2}x_2 + \dots + q_{KV}x_V, \end{aligned} \quad (1)$$

where $c_k > 0$, $k = 1, \dots, K$ are the random constant rate parameters and p_{kv} and q_{kv} , $k = 1, \dots, K$, $v = 1, \dots, V$, denote the coefficients of the population of the v -th species, before and after the k -th interaction, respectively. We will refer to p_{kv} and q_{kv} as reactant and product coefficients, respectively, by analogy with terminology commonly used in biochemical reactions. A matrix \mathbf{P} of size $K \times V$ contains the reactant coefficients p_{kv} and, similarly, \mathbf{Q} contains the product coefficients q_{kv} . The stoichiometry matrix of size $V \times K$ is defined as $\mathbf{S} = (\mathbf{Q} - \mathbf{P})^\top$. The vector $\mathbf{c} = [c_1, \dots, c_K]^\top$ contains the rate parameters.

Let $x_\nu(t)$, $\nu = 1, \dots, V$, denote the nonnegative, integer population of species x_ν at time t , and let $\mathbf{x}(t) = [x_1(t), \dots, x_V(t)]^\top$ denote the state of the system at this time instant. Let $\mathbf{x}_n = [x_{1,n}, \dots, x_{V,n}]^\top$ denote the state of the system at discrete time instants $t = n\Delta$, $n = 1, \dots, N$, i.e., $x_{\nu,n} = x_\nu(n\Delta)$ where Δ denotes a time-discretization period. We denote by \mathbf{x} the $VN \times 1$ vector containing the population of all species at N consecutive discrete time instants, i.e., $\mathbf{x} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_N^\top]^\top$.

The k -th rate equation describes an interaction that takes place stochastically according to its instantaneous rate or hazard function

$$h_k(t) = c_k \prod_{v=1}^V \binom{x_\nu(t)}{p_{kv}}, \quad k = 1, \dots, K, \quad (2)$$

where the product of binomial coefficients represents the number of combinations in which the k -th interaction can occur, as a function of the population of the reactant species with population x_ν , $\nu = 1, \dots, V$. We additionally define the vector $\mathbf{h}(t) = [h_1(t), \dots, h_K(t)]^\top$. The waiting time to the next interaction is exponentially distributed with parameter $h_0(t) = \sum_{k=1}^K h_k(t)$, and the probability of each type of interaction is given by $h_k(t)/h_0(t)$, $k = 1, \dots, K$.

See, e.g., [1] for a detailed description and discussion of the class of SKMs.

2.2. Bayesian Inference for SKMs. We consider the log-transformed rate parameters $\boldsymbol{\theta} = [\theta_1, \dots, \theta_K]^\top$, where $\theta_k = \log(c_k)$,

$k = 1, \dots, K$, with prior pdf $p(\boldsymbol{\theta})$. The prior pdf (for simplicity of notation, in this section we use p to denote the pdfs in the model. We write conditional pdfs as $p(\mathbf{y} | \mathbf{x})$, and joint densities as $p(\boldsymbol{\theta}) = p(\theta_1, \dots, \theta_K)$. This is an argument-wise notation, hence $p(\theta_1)$ denotes the distribution of θ_1 , possibly different from $p(\theta_2)$) of the initial population vector \mathbf{x}_0 is denoted by $p(\mathbf{x}_0)$. We assume that a linear combination of the populations of a subset of species is observed at discrete time instants corrupted by Gaussian noise, i.e.,

$$\mathbf{y}_n = \mathbf{M}\mathbf{x}_n + \mathbf{w}_n, \quad n = 1, \dots, N, \quad (3)$$

where \mathbf{M} is the observation matrix with dimensions $D \times V$ and $\mathbf{w}_n \sim \mathcal{N}_D(\mathbf{w}_n; \mathbf{0}, \boldsymbol{\Lambda})$ is a multivariate Gaussian noise component with zero mean and covariance matrix $\boldsymbol{\Lambda}$. We denote the complete observation vector with dimension $DN \times 1$ as $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_N^\top]^\top$.

The dynamical behavior of an arbitrary SKM may be described in terms of the following set of equations (that describe a state-space model with unknown parameters):

$$\begin{aligned} \boldsymbol{\theta} &\sim p(\boldsymbol{\theta}) \quad (\text{parameters prior}), \\ \mathbf{x}_0 &\sim p(\mathbf{x}_0) \quad (\text{populations prior}), \\ \mathbf{x}_n &\sim p(\mathbf{x}_n | \mathbf{x}_{n-1}, \boldsymbol{\theta}) \quad (\text{transition equation}), \\ \mathbf{y}_n &\sim p(\mathbf{y}_n | \mathbf{x}_n) \quad (\text{observation equation}), \end{aligned} \quad (4)$$

where $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \boldsymbol{\theta})$ and $p(\mathbf{y}_n | \mathbf{x}_n)$ denote the transition pdf and the likelihood function, respectively. The Gillespie algorithm [18] (see Appendix A for a detailed description) allows to perform exact forward simulations of arbitrary SKMs, drawing samples from the transition densities $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \boldsymbol{\theta})$, $n = 1, \dots, N$, given a set of log-rate parameters $\boldsymbol{\theta}$ and an initial population \mathbf{x}_0 . While the assumption Gaussian observational noise may not be adequate for some applications of SKMs, it has been employed by other authors in the past [5, 11] and, in particular, it has been used for computer experiments involving the prokaryotic autoregulation model [5] that we study numerically in Section 5.

In this paper, we aim at obtaining a Monte Carlo approximation of the full joint posterior distribution of the log-rate parameters $\boldsymbol{\theta}$ and the populations \mathbf{x} , with density

$$p(\boldsymbol{\theta}, \mathbf{x} | \mathbf{y}) \propto p(\mathbf{y} | \mathbf{x}) p(\mathbf{x} | \mathbf{x}_0, \boldsymbol{\theta}) p(\mathbf{x}_0) p(\boldsymbol{\theta}), \quad (5)$$

given the prior distributions $p(\boldsymbol{\theta})$ and $p(\mathbf{x}_0)$, the transition pdf $p(\mathbf{x} | \mathbf{x}_0, \boldsymbol{\theta}) = \prod_{n=1}^N p(\mathbf{x}_n | \mathbf{x}_{n-1}, \boldsymbol{\theta})$, and the likelihood function $p(\mathbf{y} | \mathbf{x}) = \prod_{n=1}^N p(\mathbf{y}_n | \mathbf{x}_n)$ constructed from (3). We note that this state-space model can represent the SKM reaction equations of Section 2.1, yet it is abstract enough to make it applicable to other, relatively similar, models. For example, the assumption of mass action kinetics is not explicitly needed for this representation; hence other reaction rates could be incorporated into the state-space model (and the inference algorithms to be derived from it).

We are also interested in computing approximations of the posterior marginals of the rate parameters $p(\boldsymbol{\theta} | \mathbf{y}) = \int p(\boldsymbol{\theta}, \mathbf{x} | \mathbf{y}) d\mathbf{x}$ and the species populations $p(\mathbf{x} | \mathbf{y}) =$

$\int p(\boldsymbol{\theta}, \mathbf{x} | \mathbf{y}) d\boldsymbol{\theta}$ as well as their moments (e.g., the posterior mean), which are of the form

$$\begin{aligned} E_{p(\boldsymbol{\theta}|\mathbf{y})} [f(\boldsymbol{\theta})] &= \int f(\boldsymbol{\theta}) p(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}, \\ E_{p(\mathbf{x}|\mathbf{y})} [f(\mathbf{x})] &= \int f(\mathbf{x}) p(\mathbf{x} | \mathbf{y}) d\mathbf{x}, \end{aligned} \quad (6)$$

where f is a real, integrable function.

Bayesian inference based on exact stochastic simulations from $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \boldsymbol{\theta})$ generated via the Gillespie algorithm often becomes practically intractable even for models of modest complexity [7]. Thus, it is very common to resort to a continuous approximation of the underlying stochastic process, which is known as the diffusion approximation [1]. This approximation is known to be poor in low-concentration scenarios and thus should be avoided for models involving species with a very low population. Alternatively, in [3] the authors propose a solution based on a moment closure approximation of the stochastic process. Other techniques for the approximate modelling of the underlying jump process that have become relatively popular in the past few years include Gaussian processes [12] and the linear noise approximation [8–10].

This inference problem has been traditionally addressed using MCMC methods, and IS based schemes have been avoided due to their inefficiency in high dimensional spaces [1]. In [2] various MCMC algorithms are evaluated in data-poor scenarios. In [5] a likelihood-free pMCMC scheme [6] is applied to this problem. This method is, to the best of our knowledge, the most powerful, yet computationally expensive, method provided so far for this kind of applications.

In [16] a NPMC scheme is proposed for the approximation of the marginal posterior pdf $p(\boldsymbol{\theta} | \mathbf{y})$. The performance of the NPMC method is tested in a simple SKM known as predator-prey model [20], providing excellent results with a low computational cost.

In this paper we compare the performances of the pMCMC and the NPMC methods in the approximation of the full joint posterior pdf $p(\boldsymbol{\theta}, \mathbf{x} | \mathbf{y})$ in (5), which allows performing Bayesian inference for the rate parameters $\boldsymbol{\theta}$ and the full sample path \mathbf{x} , including unobserved components.

3. Particle MCMC for SKM's

The particle marginal Metropolis-Hastings (PMMH) algorithm is a pMCMC method originally proposed in [6] for Monte Carlo sampling from the full posterior distribution $p(\boldsymbol{\theta}, \mathbf{x} | \mathbf{y})$. The PMMH scheme suggests a proposal mechanism of the form $q(\boldsymbol{\theta}^* | \boldsymbol{\theta}) \tilde{p}^J(\mathbf{x}^* | \mathbf{y}, \boldsymbol{\theta}^*)$. To be specific, a new candidate in the parameter space, $\boldsymbol{\theta}^*$, is drawn from an arbitrary proposal distribution $q(\boldsymbol{\theta}^* | \boldsymbol{\theta})$, while the new candidate in the variable space, \mathbf{x}^* , is generated using an approximation of the posterior marginal $p(\mathbf{x}^* | \mathbf{y}, \boldsymbol{\theta}^*)$ constructed by means of an SMC algorithm (i.e., a particle filter) with J particles and denoted $\tilde{p}^J(\mathbf{x}^* | \mathbf{y}, \boldsymbol{\theta}^*)$. The probability of accepting the proposed pair $(\boldsymbol{\theta}^*, \mathbf{x}^*)$ is

$$\min \left\{ 1, \frac{\tilde{p}^J(\mathbf{y} | \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*)}{\tilde{p}^J(\mathbf{y} | \boldsymbol{\theta}) p(\boldsymbol{\theta})} \times \frac{q(\boldsymbol{\theta} | \boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^* | \boldsymbol{\theta})} \right\}, \quad (7)$$

Initialization ($i = 0$):

1. Sample $\boldsymbol{\theta}^{(0)} \sim p(\boldsymbol{\theta})$ and
2. run a SMC scheme targeting $p(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta}^{(0)})$. Draw $\mathbf{x}^{(0)} \sim \hat{p}^J(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta}^{(0)})$ from the SMC approximation and let $\hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^{(0)})$ denote the marginal likelihood estimate.

Iteration ($i = 1, \dots, I$):

1. Sample $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)})$ and
2. run a SMC scheme targeting $p(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta}^*)$. Draw $\mathbf{x}^* \sim \hat{p}^J(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta}^*)$, let $\hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^*)$ denote the marginal likelihood estimate, and
3. with probability

$$\min \left\{ 1, \frac{\hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^*) p(\boldsymbol{\theta}^*)}{\hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^{(i-1)}) p(\boldsymbol{\theta}^{(i-1)})} \times \frac{q(\boldsymbol{\theta}^{(i-1)} | \boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(i-1)})} \right\}$$

accept the move setting $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$, $\mathbf{x}^{(i)} = \mathbf{x}^*$ and $\hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^{(i)}) = \hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^*)$. Otherwise store the current values $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$, $\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)}$ and $\hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^{(i)}) = \hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^{(i-1)})$.

ALGORITHM 1: Particle MCMC algorithm targeting $p(\boldsymbol{\theta}, \mathbf{x} | \mathbf{y})$ [6].

where $\hat{p}^J(\mathbf{y} | \boldsymbol{\theta}^*)$ is an unbiased approximation of the marginal likelihood of $\boldsymbol{\theta}^*$ (i.e., $p(\mathbf{y} | \boldsymbol{\theta}^*)$), computed, again, by way of a particle filter with J particles. The PMMH algorithm is reproduced in Algorithm 1, and the SMC approximations of $p(\mathbf{y} | \boldsymbol{\theta}^*)$ and $p(\mathbf{x}^* | \mathbf{y}, \boldsymbol{\theta}^*)$ are described in Appendix B. Full details can be found in [6]. Note that the forward simulation of the stochastic process in the particle filter may be performed exactly with the Gillespie algorithm or using a diffusion approximation.

In [5] the proposal is selected as a Gaussian random walk $q(\boldsymbol{\theta}^* | \boldsymbol{\theta}) = \mathcal{N}_{\mathcal{K}}(\boldsymbol{\theta}^*; \boldsymbol{\theta}, \gamma^2 \mathbf{I})$, ignoring the correlation structure of the target distribution. The variance γ^2 has to be tuned and partly determines the performance of the algorithm (see [21] for some practical advice on the choice of this parameter). Low values of γ result in a poor exploration of the space of $\boldsymbol{\theta}$, while high γ values yield low acceptance rates. In both situations the resulting chain presents highly correlated samples and slow mixing properties. To reduce the correlation among samples it is common to *thin* the obtained chain, keeping a subset of equally spaced samples and discarding the rest. After removing the initial burn-in samples and thinning the output, we obtain a Markov chain $\{\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^M$ with M correlated samples. Then, we may construct a sample approximation of the marginal posterior distributions of the parameters $\boldsymbol{\theta}$ and the populations \mathbf{x} , as

$$\begin{aligned} \hat{p}^M(d\boldsymbol{\theta} | \mathbf{y}) &= \frac{1}{M} \sum_{i=1}^M \delta_{\boldsymbol{\theta}^{(i)}}(d\boldsymbol{\theta}), \\ \hat{p}^M(d\mathbf{x} | \mathbf{y}) &= \frac{1}{M} \sum_{i=1}^M \delta_{\mathbf{x}^{(i)}}(d\mathbf{x}), \end{aligned} \quad (8)$$

respectively, where $\delta_{\boldsymbol{\theta}^{(i)}}$ and $\delta_{\mathbf{x}^{(i)}}$ denote the unit delta measure centered at $\boldsymbol{\theta}^{(i)}$ and $\mathbf{x}^{(i)}$, respectively. The approximation of the full joint posterior has the form

$$\hat{p}^M(d\boldsymbol{\theta}, d\mathbf{x} | \mathbf{y}) = \frac{1}{M} \sum_{i=1}^M \delta_{(\boldsymbol{\theta}^{(i)}, \mathbf{x}^{(i)})}(d\boldsymbol{\theta}, d\mathbf{x}). \quad (9)$$

4. Nonlinear PMC for SKM's

The PMC method [13] is an iterative IS scheme that generates a sequence of proposal pdf's $q_\ell(\cdot)$, $\ell = 1, \dots, L$, that approximate a target pdf π along the iterations. The NPMC scheme is proposed in [16] and it introduces nonlinearly transformed IWs (TIWs) in order to mitigate the numerical problems caused by degeneracy in the proposal update scheme.

4.1. NPMC Targeting $p(\boldsymbol{\theta} | \mathbf{y})$. We first consider as a target density the marginal posterior pdf of the parameters $\boldsymbol{\theta}$ given the observation vector \mathbf{y} , i.e., $\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta} | \mathbf{y})$. As in [16], we construct the proposal pdf $q_\ell(\boldsymbol{\theta})$, $\ell = 2, \dots, L$, as a Gaussian approximation of the target pdf obtained at the previous iteration $\ell - 1$, whose mean and covariance parameters are selected to match the moments of the previous sample set. The NPMC algorithm is displayed in Algorithm 2. Details and some simple convergence results can be found in [16]. See [17] for a rigorous analysis.

Similar to the pMCMC algorithm, in the NPMC implementation the densities $p(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta})$ and $p(\mathbf{y} | \boldsymbol{\theta})$ required in steps 2 and 3 are replaced by their SMC approximations, which are given in Appendix B. The NPMC method may also use either exact or approximate samples of the stochastic process, depending on the computational capabilities.

For the *clipping* procedure performed in step 4 we consider, at each iteration ℓ , a permutation i_1, \dots, i_M of the indices in $\{1, \dots, M\}$ such that $w_\ell^{(i_1)*} \geq \dots \geq w_\ell^{(i_M)*}$ and choose a *clipping* parameter $M_T < M$. We select a threshold value $\mathcal{T}_\ell^M = w_\ell^{(i_{M_T})*}$ and *clip* the largest IWs, $w_\ell^{(i_k)*} \geq \mathcal{T}_\ell^M$, $k = 1, \dots, M_T - 1$. This transformation leads to M_T flat TIWs in the region of interest of $\boldsymbol{\theta}$, allowing for a robust update of the proposal. The performance of the algorithm shows little sensitivity to the selection of the *clipping* parameter M_T [16]. For simplicity, step 5 performs multinomial resampling.

At each iteration of the NPMC algorithm we may construct a discrete approximation of the posterior probability distribution described by the pdf $p(\boldsymbol{\theta} | \mathbf{y})$, based on the set of samples and TIWs, as

Iteration ($\ell = 1, \dots, L$):

1. Draw a set of M samples $\{\theta_\ell^{(i)}\}_{i=1}^M$ from the proposal density $q_\ell(\theta)$:
 - (i) at iteration $\ell = 1$, let $q_1(\theta) = p(\theta)$.
 - (ii) at iterations $\ell = 2, \dots, L$ the proposal $q_\ell(\theta)$ is the Gaussian approximation of $p(\theta | \mathbf{y})$ obtained at iteration $\ell - 1$.
2. For $i = 1, \dots, M$, run a SMC scheme with J particles targeting $p(\mathbf{x} | \mathbf{y}, \theta_\ell^{(i)})$ and compute the marginal likelihood estimate $\hat{p}_\ell^J(\mathbf{y} | \theta_\ell^{(i)})$.
3. For $i = 1, \dots, M$, compute the unnormalized IWs

$$w_\ell^{(i)*} \propto \frac{\hat{p}_\ell^J(\mathbf{y} | \theta_\ell^{(i)}) p(\theta_\ell^{(i)})}{q_\ell(\theta_\ell^{(i)})}.$$

4. For $i = 1, \dots, M$, compute normalized TIWs, $\bar{w}_\ell^{(i)}$, by *clipping* the original IWs as

$$\bar{w}_\ell^{(i)*} = \min(w_\ell^{(i)*}, \mathcal{F}_\ell^{M_T}), \quad \bar{w}_\ell^{(i)} = \frac{\bar{w}_\ell^{(i)*}}{\sum_{j=1}^M \bar{w}_\ell^{(j)*}},$$

where the threshold value $\mathcal{F}_\ell^{M_T}$ denotes the M_T -th highest unnormalized IW $w_\ell^{(i)*}$, with $1 < M_T < M$.

5. Resample to obtain an unweighted set $\{\tilde{\theta}_\ell^{(i)}\}_{i=1}^M$: for $i, j = 1, \dots, M$, let $\tilde{\theta}_\ell^{(i)} = \theta_\ell^{(j)}$ with probability $\bar{w}_\ell^{(j)}$.
6. Construct a Gaussian approximation $q_{\ell+1}(\theta) = \mathcal{N}(\theta; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)$ of the posterior $p(\theta | \mathbf{y})$, where the mean vector and covariance matrix are computed as

$$\boldsymbol{\mu}_\ell = \frac{1}{M} \sum_{i=1}^M \tilde{\theta}_\ell^{(i)} \quad \text{and} \quad \boldsymbol{\Sigma}_\ell = \frac{1}{M} \sum_{i=1}^M (\tilde{\theta}_\ell^{(i)} - \boldsymbol{\mu}_\ell) (\tilde{\theta}_\ell^{(i)} - \boldsymbol{\mu}_\ell)^\top. \quad (*)$$

ALGORITHM 2: Nonlinear PMC targeting $\pi(\theta) = p(\theta | \mathbf{y})$.

$$\hat{p}_\ell^M(d\theta | \mathbf{y}) = \sum_{i=1}^M \bar{w}_\ell^{(i)} \delta_{\theta_\ell^{(i)}}(d\theta). \quad (10)$$

$$\hat{p}_\ell^M(d\mathbf{x} | \mathbf{y}) = \sum_{i=1}^M \bar{w}_\ell^{(i)} \delta_{\mathbf{x}_\ell^{(i)}}(d\mathbf{x}). \quad (12)$$

The choice of a Gaussian approximation of the proposal $q_{\ell+1}(\theta)$ in step 6 is arbitrary. We have adopted it here for simplicity. Other families of pdfs can be used without modifying the rest of the algorithm.

4.2. NPMC Targeting $p(\theta, \mathbf{x} | \mathbf{y})$. The NPMC method proposed in [16] may be readily applied to the approximation of the full joint posterior $p(\theta, \mathbf{x} | \mathbf{y})$, in a manner equivalent to the pMCMC algorithm. We consider a sampling mechanism of the form $q(\theta) \hat{p}^J(\mathbf{x} | \mathbf{y}, \theta)$, where samples $\theta^{(i)}$ are again generated from the latest proposal $q(\theta)$ and $\mathbf{x}^{(i)}$ are drawn from the SMC approximation $\hat{p}^J(\mathbf{x} | \mathbf{y}, \theta^{(i)})$ obtained via particle filtering (the iteration index has been omitted for simplicity). Then, the standard, unnormalized IW associated with the pair $(\theta^{(i)}, \mathbf{x}^{(i)})$ is computed as

$$\begin{aligned} w^{(i)*} &= \frac{\hat{p}^J(\theta^{(i)}, \mathbf{x}^{(i)} | \mathbf{y})}{q(\theta^{(i)}) \hat{p}^J(\mathbf{x}^{(i)} | \mathbf{y}, \theta^{(i)})} \\ &\propto \frac{\hat{p}^J(\mathbf{x}^{(i)}, \mathbf{y} | \theta^{(i)}) p(\theta^{(i)})}{q(\theta^{(i)}) \hat{p}^J(\mathbf{x}^{(i)} | \mathbf{y}, \theta^{(i)})} \\ &\propto \frac{\hat{p}^J(\mathbf{y} | \theta^{(i)}) p(\theta^{(i)})}{q(\theta^{(i)})} \end{aligned} \quad (11)$$

and is independent of \mathbf{x} . This reveals that, when samples $\mathbf{x}_\ell^{(i)}$ are drawn from $\hat{p}^J(d\mathbf{x} | \mathbf{y}, \theta)$, the algorithm yields a discrete approximation of the posterior distribution of the unobserved populations \mathbf{x} constructed as

Even though the proposed NPMC and the pMCMC require very similar computations for each pair of samples of $\{\theta, \mathbf{x}\}$ and thus have an equivalent computational cost, the NPMC has a set of important advantages with respect to its MCMC counterpart. PMC methods provide independent sets of samples at all iterations and do not require a burn-in period. On the other hand, the nonlinearity applied in the NPMC mitigates weight degeneracy, which is the main problem arising in conventional IS based methods, dramatically increasing its efficiency in high dimensional problems. As a consequence, we claim that the total number of samples, ML (and thus, the computational complexity), required by the NPMC can be significantly lower than that of pMCMC, I . Additionally, contrary to pMCMC, which requires a careful choice of the proposal tuning parameter, the proposed method does not require the accurate fitting of any parameters.

The NPMC method processes a set of M i.i.d. samples at each iteration, requiring a low number of iterations (around 10 for the type of problems addressed here) for convergence to the target distribution. The bulk of the computational cost of NPMC, as well as of pMCMC, is the SMC approximation of the likelihood function. In the pMCMC algorithm the samples $\theta^{(i)}$ are processed sequentially (one after the other), and this process cannot be parallelized. On the contrary, at each iteration ℓ of the NPMC method, the process of drawing M samples $\theta_\ell^{(i)}$ and computing the associated IWs $w_\ell^{(i)*}$ can be performed independently for each sample i . Thus, steps 1, 2, and 3 can be easily parallelized, reducing the total execution time up to that of a single sample $\theta_\ell^{(i)}$. On the other hand, steps 4, 5, and 6 require the complete set of samples and

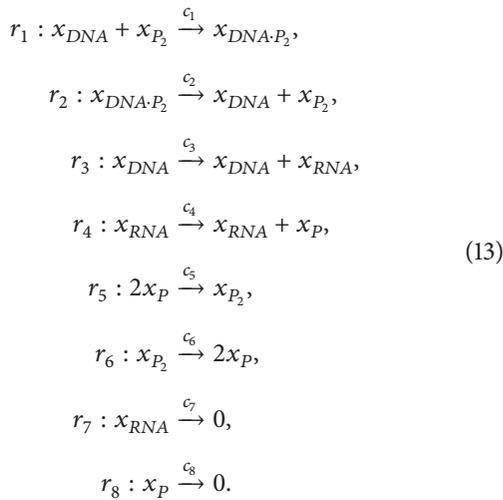
weights $\{\theta_\ell^{(i)}, w_\ell^{(i)*}\}_{i=1}^M$ and must be performed in a centralized manner. However, these computations have a negligible cost in comparison with a single likelihood approximation. Thus, the parallelization of the NPMC method can allow for a reduction in execution time up to a factor $\approx 1/M$. Note that we refer here to the parallelization of the PMC method and not of the SMC filter used to approximate the likelihood function.

An extensive numerical comparison of pMCMC versus NPMC for a complex SKM that can be interpreted as a generalization of the paradigmatic Lotka-Volterra model is presented in Section 5.

5. Example: An Autoregulatory Feedback Network

In this section, we compare the performance of the pMCMC and the NPMC methods when applied to the problem of approximating the posterior distributions of the log-rate parameters $p(\theta \mid \mathbf{y})$ and the populations $p(\mathbf{x} \mid \mathbf{y})$ in a complex SKM given some observed data \mathbf{y} . This model is significantly more involved than the standard predator-prey system. It has been introduced in [7] and further analyzed in [1, 5] for biochemical processes. In particular, it has also been considered as a model for representing the mechanisms for autoregulation in prokaryotes. The model is minimal in terms of the level of details included and offers a simplistic view of the mechanisms involved in gene autoregulation. However, it contains many of the interesting features of general autoregulatory feedback networks and it does provide sufficient detail to capture the network dynamics.

5.1. Prokaryotic Autoregulatory Model. The prokaryotic autoregulatory model is a SKM that involves $V = 5$ chemical species and $K = 8$ reaction equations, r_1, \dots, r_K , given by [7]



We construct the V -dimensional vector containing the population of all species at time instant t as $\mathbf{x}(t) = [x_{RNA}(t), x_P(t), x_{P_2}(t), x_{DNA \cdot P_2}(t), x_{DNA}(t)]^\top$. Thus, we obtain a stoichiometry matrix of the form

$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 2 & 0 & -1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{14}$$

and the hazard vector is given by

$$\mathbf{h}(t) = \left[c_1 x_{DNA} x_{P_2}, c_2 x_{DNA \cdot P_2}, c_3 x_{DNA}, c_4 x_{RNA}, c_5 \cdot \frac{x_P(x_P - 1)}{2}, c_6 x_{P_2}, c_7 x_{RNA}, c_8 x_P \right]^\top, \tag{15}$$

where the time dependence of the population of each species is omitted for notational simplicity.

This model involves a conservation law given by the relation $x_{DNA \cdot P_2} + x_{DNA} = C$, where C is the number of copies of this gene in the genome. We could use this relation to remove $x_{DNA \cdot P_2}$ from the model, replacing any occurrences of the latter in the hazard function with $C - x_{DNA}$, but in this paper we abide by the notation in equation (15). Further details of this model can be found in [1].

5.2. Simulation Setup. We have selected most of the simulation parameters following [5]. The true vector of rate parameters which we aim to estimate has been set to

$$\mathbf{c} = [0.1, 0.7, 0.35, 0.2, 0.1, 0.9, 0.3, 0.1]^\top, \tag{16}$$

which yields log-transformed rate parameters

$$\theta = -[2.30, 0.36, 1.05, 1.61, 2.30, 0.10, 1.20, 2.30]^\top. \tag{17}$$

The initial populations and the conservation constant have been set to $\mathbf{x}_0 = [x_1(0), \dots, x_V(0)]^\top = [8, 8, 8, 5, 5]^\top$ and $C = 10$, respectively. In all the simulations in this paper we have performed exact sampling from the stochastic model with the Gillespie algorithm to obtain the likelihood approximation via particle filtering. The number of particles for the SMC approximation $\hat{p}^J(\mathbf{x} \mid \theta, \mathbf{y})$ has been set to $J = 100$ for all the simulations (both the pMCMC and the NPMC algorithms converge (as $I \rightarrow \infty$ and $M \rightarrow \infty$, respectively) even for fixed J [6, 17]. We have tested numerically that for $J > 100$ performance hardly improves, while running times obviously become larger).

Independent uniform priors $\mathcal{U}(\theta_k; -7, 2)$ have been taken for each $\theta_k = \log(c_k)$. Opposite to [5], the initial populations \mathbf{x}_0 are assumed unknown for the inference algorithm, which employs independent Poisson priors $p(x_\nu(0)) = \mathcal{P}(x_\nu(0); \lambda_\nu)$, with the λ_ν parameters set to the true initial populations, that is, $\lambda_\nu = x_\nu(0)$, $\nu = 1, \dots, V$. Following [5], the conservation constant C is assumed to be known in the simulations.

We consider two different observation scenarios. In the complete observation (CO) scenario we assume that all species x_ν , $\nu = 1, \dots, V$, are observed at regular time intervals of length $\Delta = 1$ and corrupted by Gaussian

noise with variance $\sigma^2 = 4$ (assumed to be known). Thus, the observation matrix is of the form $\mathbf{M} = \mathbf{I}_V$ and the observations are given by

$$\mathbf{y}_n = \mathbf{x}_n + \mathbf{w}_n, \quad n = 1, \dots, N. \quad (18)$$

In the CO case the complete vector of observations $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_N^\top]^\top$ has dimension $VN \times 1$.

In the partial observation scenario (PO) only a linear combination of the proteins $x_P + 2x_{P_2}$ is observed, also contaminated by Gaussian noise, i.e., the observation matrix is given by $\mathbf{M} = [0, 1, 2, 0, 0]$ (with dimension $1 \times V$) and the observations are generated as

$$y_n = x_{2,n} + 2x_{3,n} + w_n, \quad \text{where } w_n \sim \mathcal{N}_1(w_n; 0, \sigma^2). \quad (19)$$

In the PO case, a vector of scalar observations with dimension $N \times 1$ is constructed as $\mathbf{y} = [y_1, \dots, y_N]^\top$.

We remark that the assumption of the observation noise terms (either \mathbf{w}_n or w_n) displaying a Gaussian distribution is not needed to apply neither the pMCMC method nor the NPMC algorithm. Both techniques can be applied as long as the joint likelihood $p(\mathbf{y}_n | \mathbf{x}_n, \boldsymbol{\theta})$ of the state and the parameters can be evaluated, numerically and point-wise (either exactly or approximately; see [22]). In particular, it is possible to plug a Gaussian-process estimate of the likelihood [12] into the NPMC algorithm while running the rest of the algorithm in the same way as described in Section 4 (and still rely on the theoretical guarantees introduced in [22]).

5.3. Performance Evaluation. To evaluate the performance of the pMCMC and the NPMC methods we compute, in all the simulation runs, the mean square error (MSE) attained by the sample set that approximates the marginal posterior of $\boldsymbol{\theta}$, generated by both schemes.

For the pMCMC method, we compute the MSE of each parameter θ_k based on the M -size final output (after removing the burn-in period and thinning), as

$$\text{MSE}_k = \frac{1}{M} \sum_{i=1}^M (\theta_k^{(i)} - \theta_k)^2, \quad k \in \{1, \dots, K\}. \quad (20)$$

For the NPMC method, we compute the MSE associated with each parameter θ_k , $k = 1, \dots, K$, based on the unweighted sample set at the ℓ -th iteration $\{\tilde{\theta}_\ell^{(i)}\}_{i=1}^M$, $\ell = 1, \dots, L$, as

$$\text{MSE}_{\ell,k} = \frac{1}{M} \sum_{i=1}^M (\tilde{\theta}_{\ell,k}^{(i)} - \theta_k)^2 = (\mu_{\ell,k} - \theta_k)^2 + \sigma_{\ell,k}^2, \quad (21)$$

where $\mu_{\ell,k}$ is the k -th component of the mean vector $\boldsymbol{\mu}_\ell$ and the variance term $\sigma_{\ell,k}^2$ is the (k, k) component of matrix $\boldsymbol{\Sigma}_\ell$.

We have chosen the MSE as a figure of merit because the observations are generated synthetically and, therefore, we have access to ground truth values for the rate parameters that can be used to compute MSEs. A direct comparison in terms of the posterior distributions (e.g., using the Kullback-Leibler divergence or the total variation distance) is not possible

because the posterior distribution of $\boldsymbol{\theta}$ conditional on the observations \mathbf{y} is intractable.

However, the MSE cannot be computed in real problems, where the true parameters θ_k are unknown. To monitor the stability and the efficiency of the two sampling schemes based on the generated sample alone, we resort to the so-called normalized effective sample size (NESS), which is often defined differently for MCMC and IS schemes [23].

In the MCMC literature, the NESS gives the relative size of an i.i.d. (independent and identically distributed) sample with the same variance as the current sample and thus indicates the loss in efficiency due to the use of a Markov chain [23]. For pMCMC we compute the NESS from the final chain (after removing the burn-in period and thinning) as

$$M^{\text{neff}} = \frac{1}{1 + 2 \sum_{j=1}^{\infty} \hat{\rho}(j)}, \quad (22)$$

where $\hat{\rho}(j) = \text{corr}(\boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(j)})$ is the average autocorrelation function (ACF) at lag j . For the computation of the NESS, we truncate j when $\hat{\rho}(j) < 0.1$.

For IS methods, the NESS may be interpreted as the relative size of a sample generated from the target distribution with the same variance as the current sample. Even when high values of the NESS do not guarantee a low approximation error, the NESS is often used as an indicator of the numerical stability of the algorithm [24]. It cannot be evaluated exactly but we may compute an approximation of the NESS at each iteration of the NPMC scheme based on the set of TIWs as

$$M_\ell^{\text{neff}} = \frac{1}{M \sum_{i=1}^M (\tilde{w}_\ell^{(i)})^2}, \quad \ell = 1, \dots, L. \quad (23)$$

In the next subsection we present, together with other numerical results, some comparisons involving the NESS for the pMCMC and PMC methods. The comparison should be taken with some caution because, although the interpretation of the NESS is the same for both techniques, the estimators have a different form and they are derived in a different way. Nevertheless, we believe that the NESS is a good numerical indicator of degeneracy for the two schemes, because it directly reflects poor mixing (high correlation) of the Markov chain and concentration of the weight in importance samplers, and we plot it with that purpose. See [25] for more details on the NESS and related statistics.

5.4. Simulation Results. We consider two simulation scenarios in which a different number of parameters is estimated.

5.4.1. Estimation of a Single Rate Parameter θ_1 . In this section we present numerical results regarding the approximation of the posterior distribution $p(\theta_1, \mathbf{x} | \boldsymbol{\theta}_{\setminus 1}, \mathbf{y})$ of a single rate parameter $\theta_1 = \log c_1$ and the populations \mathbf{x} , when the rest of parameters $\boldsymbol{\theta}_{\setminus 1} = [\theta_2, \dots, \theta_K]^\top$, are assumed known.

We compare the pMCMC and the NPMC methods in this simple scenario in order to show that the two algorithms perform almost equivalently in low dimensional problems. This is in a clear contrast with the results presented in

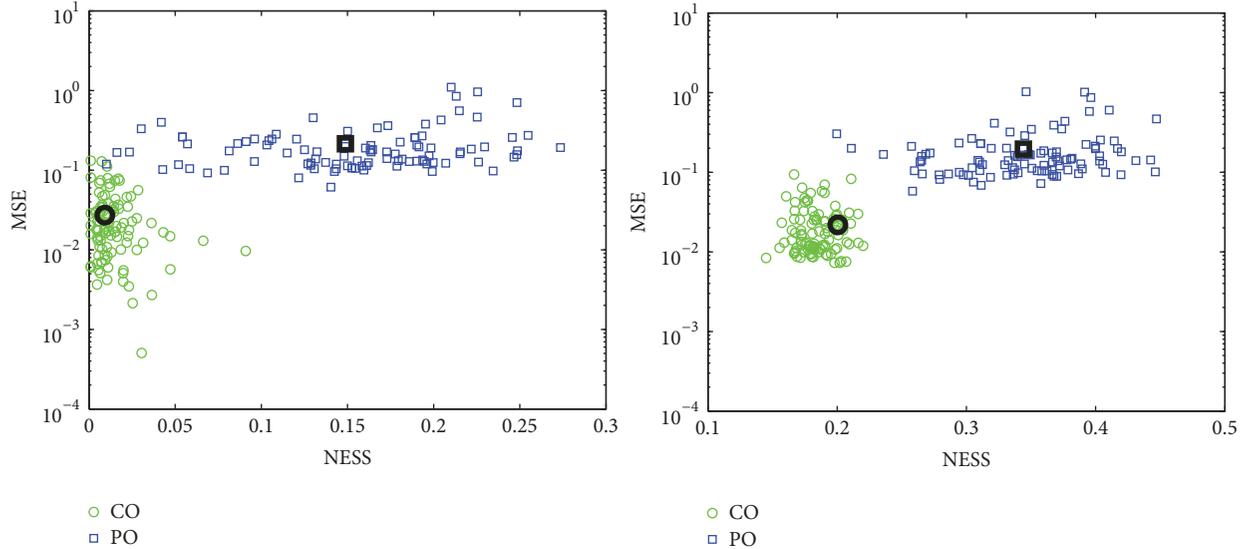


FIGURE 1: Performance of the pMCMC (*left*) and the NPMC (*right*) methods for the estimation of a unique rate parameter θ_1 : MSE (in logarithmic scale) obtained from the final output versus the NESS for each simulation run in the CO and the PO scenario. The big circles (CO) and squares (PO) represent simulation runs with a final mean MSE close to the global average.

Section 5.4.2, which show that the NPMC method can be more efficient in higher dimensional settings.

We have performed $P = 100$ independent simulation runs of the pMCMC and the NPMC schemes in the CO and the PO scenarios, with independent population and observation vectors in each simulation $p = 1, \dots, P$. Both in the CO and the PO cases, the same true population trajectories $\mathbf{x}^{(p)}$ were used, but the observations in the CO scenario, $\mathbf{y}_{CO}^{(p)}$, and in the PO scenario, $\mathbf{y}_{PO}^{(p)}$, differ. The number of observation times has been set to $N = 100$ and exactly the same data has been used for NPMC and pMCMC.

Following [5], as a proposal pdf $q(\theta^* | \theta)$ in the pMCMC scheme we consider a Gaussian random walk update with variance γ^2 . We have performed simulations with different γ values and the best results were obtained with $\gamma^2 = 1$, which are presented here. A total number of $I = 10^4$ iterations have been run in each simulation. A final sample of size $M = 10^3$ has been obtained from each Markov chain by discarding a burn-in period of 10^3 samples and thinning the output by a factor of 9.

In the NPMC scheme, the number of iterations has been set to $L = 10$, the number of samples per iteration is $M = 10^3$ and the *clipping* parameter is $M_T = 100$. In this way, the computational effort of the two methods is approximately the same, as they both generate $I = ML = 10^4$ samples in the space of θ .

In Figure 1 the final MSE obtained by the pMCMC (*left*) and the NPMC (*right*) algorithms for each simulation run is depicted versus the final NESS, in the CO and the PO scenarios. Note that the NESS is computed differently for pMCMC and NPMC. It can be observed that both algorithms perform similarly in this case, with an equivalent computational cost. Both algorithms attain on average lower MSE values in the CO scenario, as expected. However, the

NESS also takes lower values in the CO case, which indicates a worse mixing of the Markov chains in the pMCMC algorithm and also higher degeneracy in the NPMC algorithm.

In Figure 2 the evolution of the NESS (*left*) and the MSE (*right*) along the iterations of the NPMC algorithm is represented, for the CO and the PO scenarios. It can be observed that both measures attain a steady value by the 5-th iteration, both in the CO and the PO case, which suggests that actually less iterations are sufficient for this problem. Again, we observe that in the CO scenario both the NESS and the MSE reach lower values.

Figure 3 (*left*) plots the average ACF of the final pMCMC sample, after removing the burn-in period and thinning the Markov chain by a factor of 9. Particularly high correlations are present in the CO case, leading to a poor NESS. Related to the ACF, the average sample acceptance probability in the pMCMC scheme in the PO scenario is 0.091, while in the CO scenario it is only 0.0034, which means that 910 samples are accepted out of $I = 10^4$ in the CO case and only 34 in the CO case.

In Figure 3 (*right*) the final pdf estimates $\hat{p}(\theta_1 | \theta_{\setminus 1}, \mathbf{y})$ of the average simulation runs represented as big circles and crosses in Figure 1 are represented in the CO and the PO scenario, for the pMCMC and the NPMC schemes. For the pMCMC method we have built a Gaussian approximation of the posterior density $p(\theta_1 | \theta_{\setminus 1}, \mathbf{y})$ based on the final MCMC sample $\{\theta_1^{(i)}\}_{i=1}^M$. For the NPMC method, this approximation corresponds to the proposal pdf for the next iteration $L + 1$, i.e., $\hat{p}(\theta_1 | \theta_{\setminus 1}, \mathbf{y}) = q_{L+1}(\theta_1) = \mathcal{N}(\theta_1; \mu_{L,1}, \sigma_{L,1}^2)$, where the mean and variance terms $\mu_{L,1}$ and $\sigma_{L,1}^2$ are computed as in (*). It can be observed in Figure 3 (*right*) that very similar results are obtained by both algorithms in this scenario. The final MSE values obtained by the pMCMC and the NPMC methods, averaged over $P = 100$ simulation runs, are shown

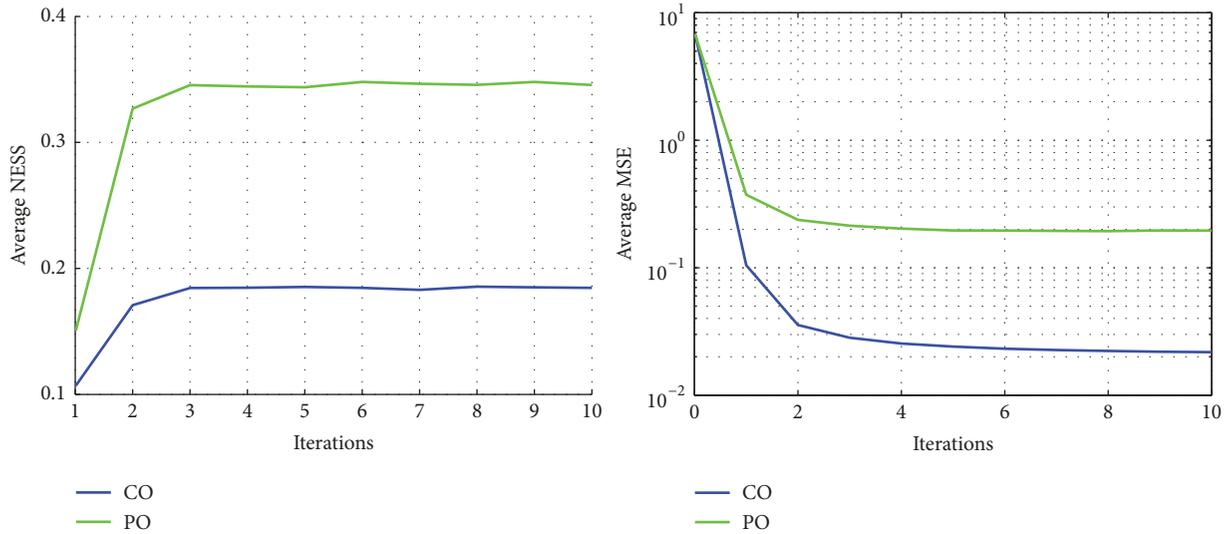


FIGURE 2: Evolution along the iterations of the NPMC algorithm of the average NESS (left) and MSE (right) in the CO and PO scenarios, estimating a single parameter θ_1 .

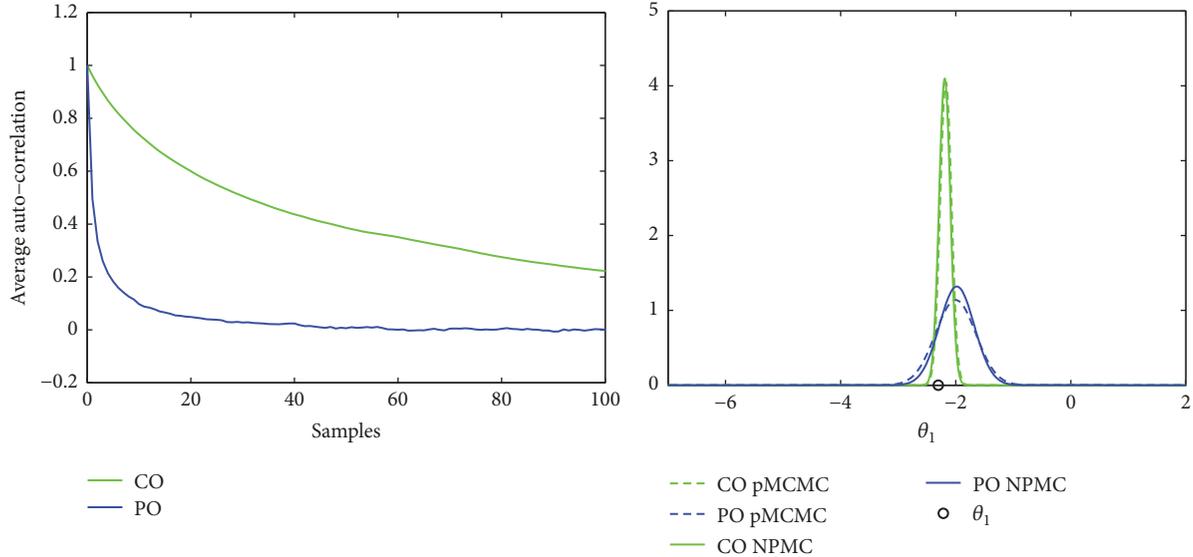


FIGURE 3: Left. Average ACF based on the final sample of size $M = 10^3$ of the pMCMC scheme in the CO and the PO scenarios, averaged over $P = 100$ simulation runs. Right. Marginal posterior pdf estimates $\hat{p}(\theta_1 | \theta_{\setminus 1}, \mathbf{y})$ of an average simulation run, for pMCMC and NPMC in the CO and PO scenarios. The true value θ_1 is also shown.

TABLE 1: Final mean and standard deviation (std) values of the MSE for θ_1 in the CO and PO scenarios, for pMCMC and NPMC. The prior values are included for comparison.

		mean MSE	std MSE
Prior		6.789	0
PO	pMCMC	0.215	0.171
	NPMC	0.195	0.170
CO	pMCMC	0.027	0.026
	NPMC	0.022	0.016

in Table 1, together with the MSE corresponding to the prior distribution.

Figure 4 depicts the posterior mean of the populations, $\hat{\mathbf{x}} = E_{\hat{p}(\mathbf{x}|\mathbf{y})}[\mathbf{x}]$, obtained with pMCMC (left) as $\hat{\mathbf{x}} = (1/M) \sum_{i=1}^M \mathbf{x}^{(i)}$ and with NPMC (right) as $\hat{\mathbf{x}} = \sum_{i=1}^M \bar{w}_L^{(i)} \mathbf{x}_L^{(i)}$ in the PO scenario. The results correspond to the particular simulation runs (different for pMCMC and NPMC) identified with big squares in Figure 1 and whose posterior approximations, $\hat{p}(\theta_1 | \theta_{\setminus 1}, \mathbf{y})$, are shown in Figure 3 (right). It can be observed that, in the PO scenario, the trend of the population of all the species is reasonably identified, even though only a linear combination of the proteins is observed. In the CO scenario the populations of all species are accurately estimated and are not shown for conciseness. Note that the populations of all species are very low, which suggests

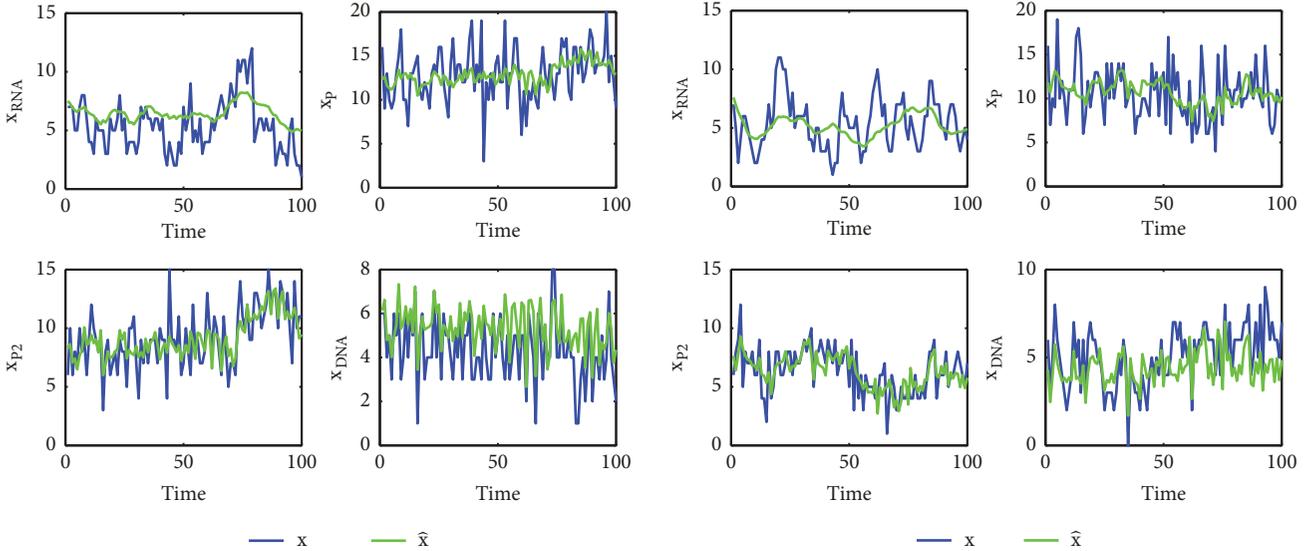


FIGURE 4: Posterior mean, $\hat{x} = E_{\hat{p}(x|y)}[x]$, of the populations obtained in a single simulation run of pMCMC (*left*) and NPMC (*right*) in the PO scenario (only a linear combination of the proteins is observed, corrupted by noise).

that the diffusion approximation may perform poorly in this scenario.

The results presented in this section reveal a very similar performance of the two methods in this simple scenario, which suggests that the parameters of pMCMC have been properly tuned. Also in terms of computational complexity pMCMC and NPMC perform very similarly. The execution time per 10^3 samples (one NPMC iteration and 10^3 pMCMC iterations) for the pMCMC scheme is 312 seconds, while for NPMC it is 325 seconds, both in the CO and in the PO cases, on a 3-GHz Intel Core 2 Duo CPU, with 4 GB of RAM. The stochastic forward simulation of the prokaryotic model with the Gillespie algorithm has been implemented in C, and the rest of the code in Matlab R2014b. Out of the execution time per iteration of the NPMC method, only 0.06 second correspond to batch steps 4, 5, and 6, and the rest of operations can be easily parallelized, as previously discussed.

However, the pMCMC method does not allow for an easy parallelization, provides a set of highly correlated samples (especially in the CO scenario) and requires the setting of the proposal variance γ^2 , the burn-in and the thinning parameters, which may not be straightforward, and determines the performance of the algorithm. On the contrary, the NPMC scheme provides uncorrelated sets of samples at each iteration and does not require the precise fitting of any parameters. Additionally, the computer simulations suggest that the convergence of the NPMC algorithm may be assessed observing the evolution of the NESS, which usually reaches a steady value simultaneously with the MSE.

5.4.2. Estimation of All the Parameters θ_k , $k = 1, \dots, K$. In this section we present simulation results to evaluate the performance of the pMCMC and the NPMC schemes in the approximation of the posterior distribution of the rate parameters and the populations of all species, $p(\theta, \mathbf{x} \mid \mathbf{y})$,

assuming that all the rate parameters are unknown, again in the CO and the PO scenarios.

In this case, $N = 200$ observation times are assumed for all the simulations. Again, $P = 100$ independent simulation runs of each algorithm have been performed. The NPMC scheme has been run for $L = 15$ iterations, with $M = 10^3$ samples per iteration and *clipping* parameter $M_T = 100$. The pMCMC scheme has been run with $I = 15 \times 10^3$ iterations in each simulation run, a burn-in period of 10^3 iterations, and thinning the output by a factor of 14. With this setup the computational effort is approximately the same in the two schemes. The variance of the random walk proposal has again been set to $\gamma^2 = 1$, since this value seems to yield the best performance.

In Figure 5 the MSE (in logarithmic scale), averaged over the parameters θ_k , attained by the pMCMC (*left*) and the NPMC (*right*) algorithms, is represented versus the NESS, in the CO and PO scenarios. Simulation runs which attained a final MSE close to the global average value are indicated with big circles (CO) and squares (PO) on both plots. It can be observed that the pMCMC method performs similarly in both scenarios, in terms of MSE and NESS, yielding poor results in both cases. On the contrary, the NPMC method provides significantly better MSE results in the CO scenario, where a larger amount of information is available. The NPMC method does not present degradation due to the high degeneracy occurring in the CO scenario.

Figure 6 depicts the evolution along the iterations of the NESS (*left*) and the MSE (*right*) averaged over $P = 100$ independent simulation runs for the NPMC algorithm. Both indices converge to a steady value in a low number of iterations also in this complex scenario. As expected, a significantly higher final MSE is attained in the extremely data-poor PO scenario.

In Figure 7 (*left*) the average ACF attained by the pMCMC in the CO and the PO cases is represented. Even after thinning

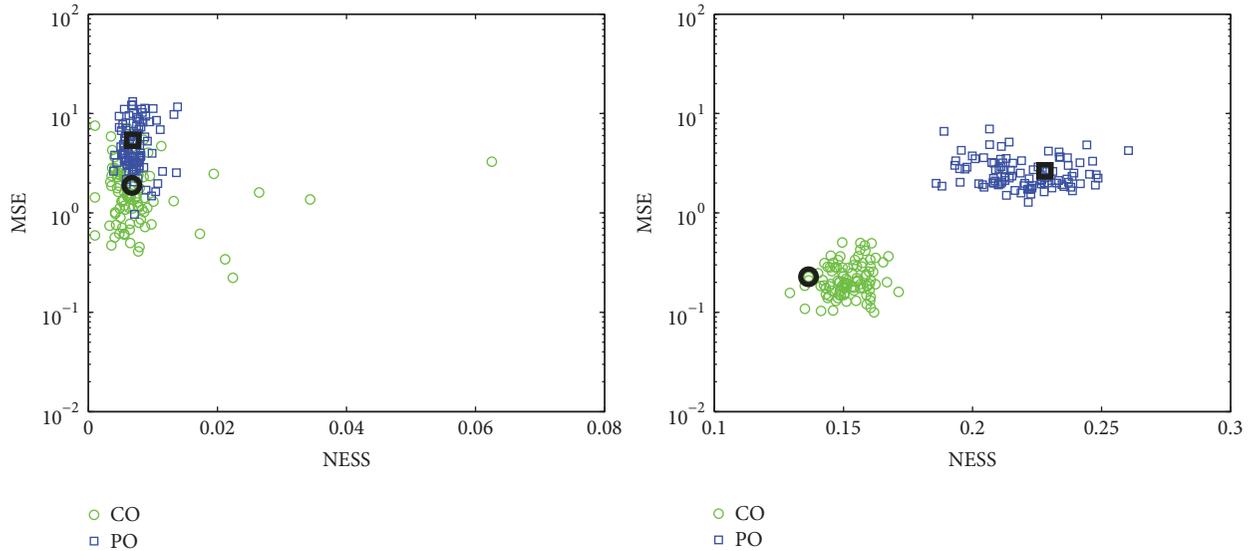


FIGURE 5: Performance of the pMCMC (*left*) and the NPMC (*right*) methods for the estimation of the whole set of rate parameters θ : MSE (in logarithmic scale) versus the final NESS, for each simulation run in the CO and the PO scenario. The big circles (CO) and squares (PO) represent simulation runs with a final mean MSE close to the global average.

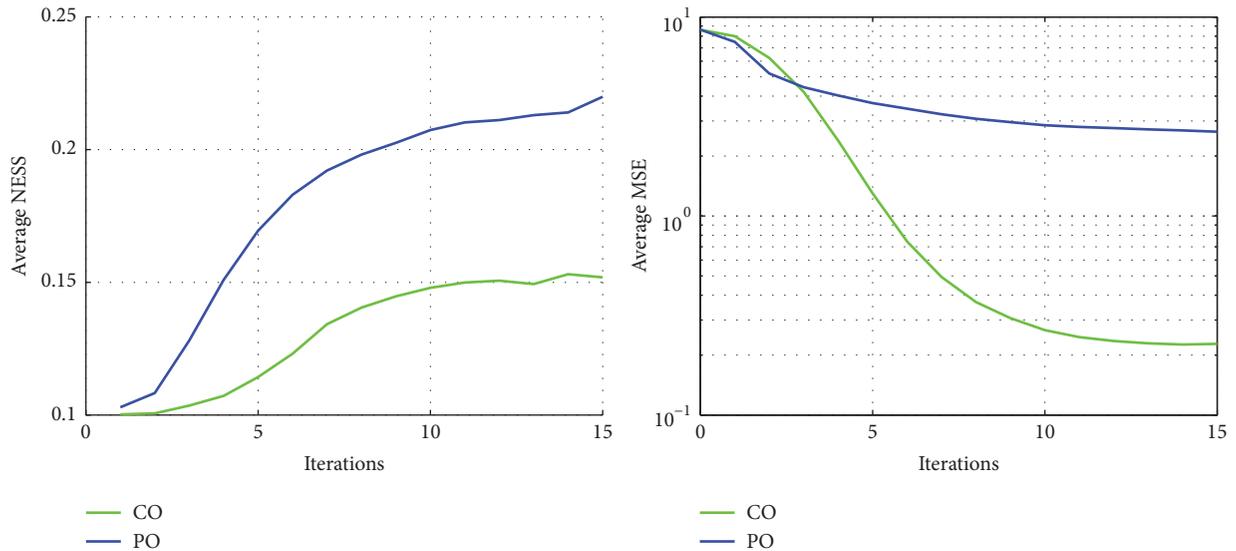


FIGURE 6: Evolution along the NPMC iterations of the average NESS (*left*) and MSE (*right*) in the CO and the PO scenario.

the output, the sample correlation is extremely high in both scenarios, which leads to a very low NESS. The acceptance rate is also very low and very long chains are required to obtain reasonable results. In the PO scenario ≈ 44 samples are accepted on average in a simulation run of $I = 15 \times 10^3$ samples (acceptance rate 0.0029). In the CO case, only ≈ 23 samples are accepted on average (rate 0.0015).

Figure 7 (*right*) depicts the final Markov chain provided by the pMCMC method (after removing the burn-in period and thinning the output) in the average simulation run represented with a big square in Figure 5 (*left*). It can be observed that the mixing of the chain is very poor, with a total number of accepted samples of 46 (close to the

average). Many other simulations, in both the PO and the CO scenarios, provide even lower number of accepted samples and thus very inconsistent results.

Figure 8 depicts the final Gaussian approximations of the marginal posteriors $p(\theta_k | \mathbf{y})$, $k = 1, \dots, 8$, obtained by the pMCMC and the NPMC methods, in the CO and PO scenarios, for the average simulation runs represented as big circles and squares in Figure 5. We can observe that the NPMC method provides a significantly better approximation of the log-rate parameters in the CO scenario, where a larger amount of data is available, which is also clear from Figure 5 (*right*). However, the pMCMC on average performs similarly in both scenarios, due to the low efficiency of the pMCMC

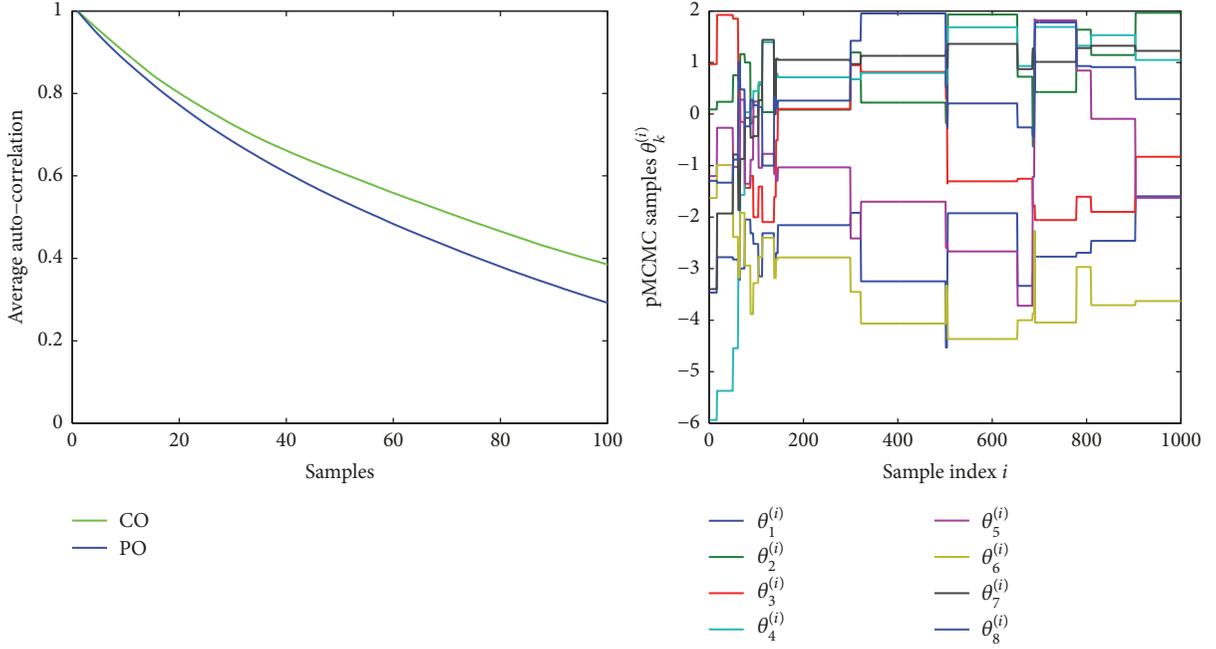


FIGURE 7: *Left.* Autocorrelations based on the final sample of size 10^3 of the pMCMC scheme in the CO and the PO scenarios, averaged over $P = 100$ simulation runs. *Right.* Markov chain provided by the pMCMC method in the PO scenario, corresponding to the average simulation run depicted with a big square in Figure 5 (*left*).

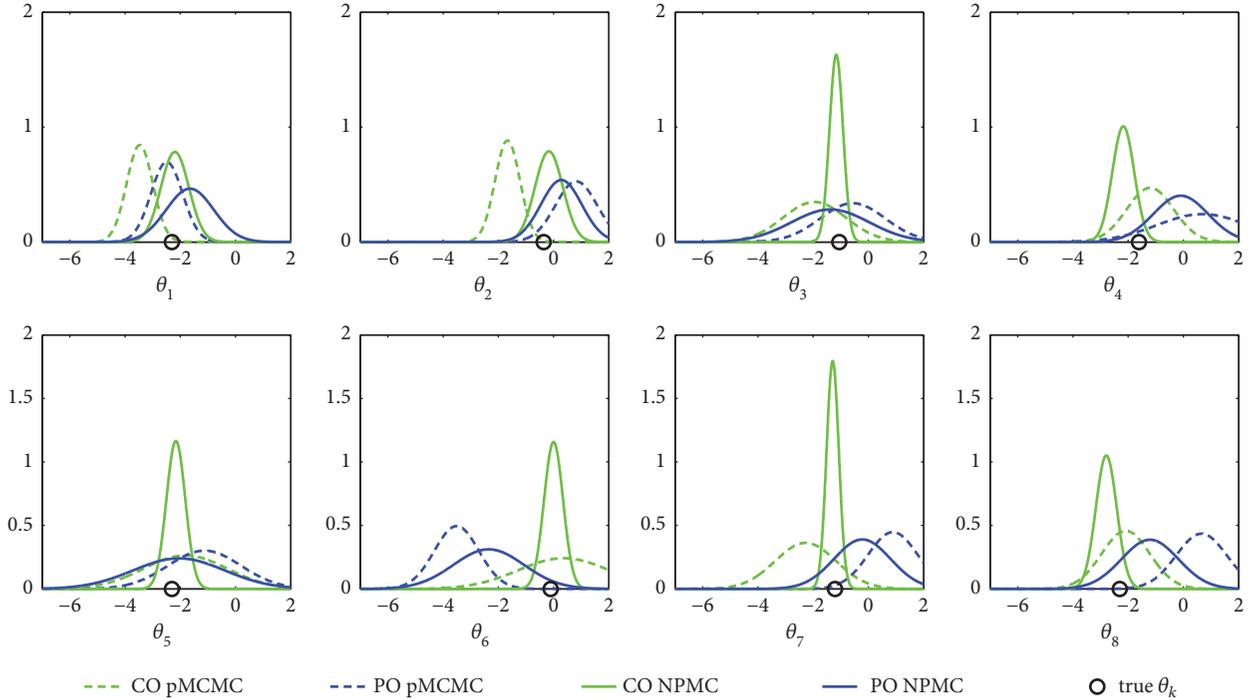


FIGURE 8: Marginal posterior pdf approximations of each parameter $\hat{p}(\theta_k | \mathbf{y})$, $k = 1, \dots, K$, attained in an average simulation run by the pMCMC and the NPMC, in the CO and in the PO case.

sampling scheme when the dimension of the problem (either K or N) increases.

In Table 2 the MSE of each parameter θ_k averaged over $P = 100$ independent simulation runs is shown, as obtained with the pMCMC and the NPMC schemes, for

the CO and the PO experiments. In the CO case, NPMC provides homogeneous results for all parameters. On the contrary, in the PO case, some of the parameters (specially θ_5 and θ_6) are significantly poorly estimated, presenting a final MSE close to the initial value (which corresponds to the

TABLE 2: Final MSE for the parameters θ_k , $k = 1, \dots, K$ in the CO and PO experiments, averaged over the simulation runs. The last two columns correspond to the mean and standard deviation (std) values of the global MSE (averaged over the parameters). The prior values are included for comparison.

		θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	mean MSE	std MSE
Prior		6.789	11.344	8.853	7.543	6.789	12.484	8.430	6.789	8.628	0
PO	pMCMC	3.412	3.319	5.543	3.200	7.059	8.929	6.799	4.371	5.329	2.926
	NPMC	1.246	1.011	2.214	1.490	4.073	7.015	2.311	1.856	2.652	1.020
CO	pMCMC	2.899	2.958	1.676	1.572	1.604	1.547	1.573	1.468	1.912	1.476
	NPMC	0.305	0.302	0.162	0.167	0.280	0.280	0.156	0.168	0.228	0.091

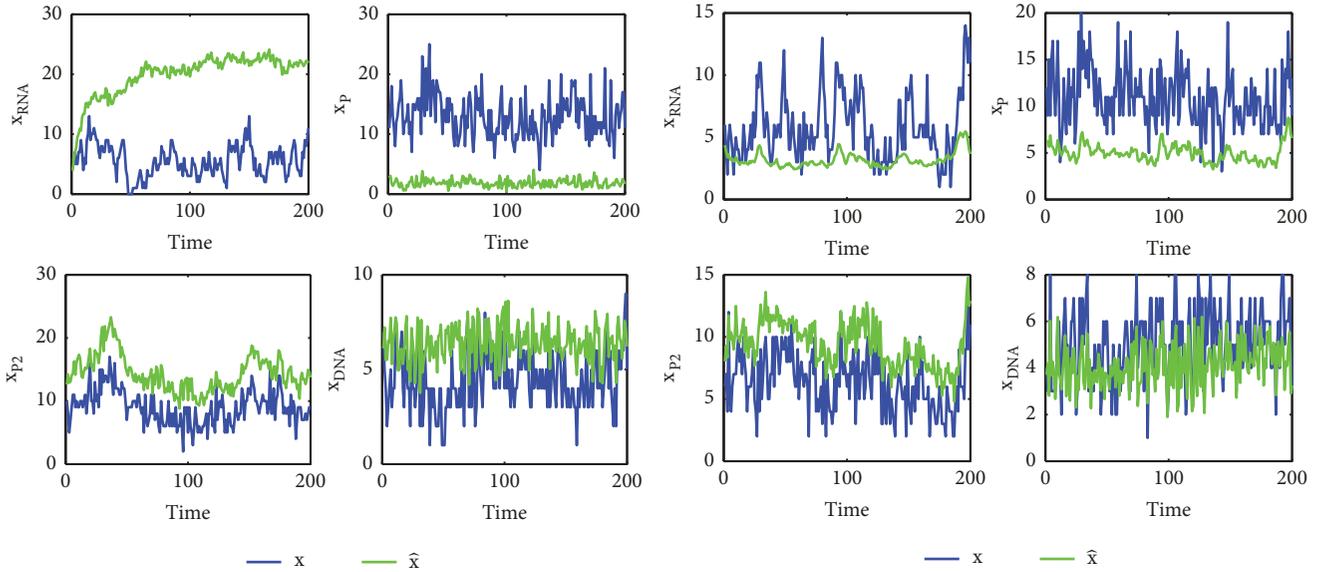


FIGURE 9: Posterior mean $\hat{\mathbf{x}} = E_{p(\mathbf{x}|\mathbf{y})}[\mathbf{x}]$ of the populations of all species obtained in the average simulation run of the pMCMC (left) and the NPMC (right) schemes, in the PO scenario.

prior knowledge). The pMCMC scheme presents significantly higher MSE values than NPMC in both observation scenarios and for all parameters θ_k .

Figure 9 depicts the population posterior mean $\hat{\mathbf{x}} = E_{p(\mathbf{x}|\mathbf{y})}[\mathbf{x}]$ corresponding to the average simulation runs of the pMCMC and the NPMC methods in the PO scenario, represented as big squares in Figure 5. Again, the NPMC method provides more accurate estimates of the unobserved populations than the pMCMC method, especially for x_{RNA} . In the CO scenario both methods provide good approximations of the populations of all species.

6. Conclusion

We have investigated the use of Monte Carlo-based Bayesian computation methods for approximating posterior distributions of the parameters and the species populations in stochastic kinetic models. Specifically, we have applied both particle Markov chain Monte Carlo (pMCMC) and nonlinear population Monte Carlo (NPMC) methods, which rely on different sampling and approximation schemes. Both pMCMC and NPMC methods resort to a sequential Monte Carlo approximation of the posterior populations to estimate the unknown interaction parameters. However, the pMCMC

constructs a Markov chain of correlated parameter samples while the NPMC algorithm is based on an importance sampling scheme with nonlinearly transformed weights to avoid degeneracy and the numerical problems typically arising in importance samplers when applied to high dimensional problems.

We have compared the performance of the two schemes for a challenging autoregulatory feedback network with 5 species and 8 unknown interaction rate parameters. Both methods have been applied without resorting to approximations of the underlying jump process (e.g., the diffusion approximation). We have shown how the NPMC algorithm outperforms the pMCMC method and requires only a moderate computational cost. Besides, the proposed method has a set of important features, common to all PMC schemes, as the sample independence, ease of parallelization, and compared to MCMC schemes, there is no need for convergence (burn-in) periods.

Let us note that there is room for improvement of both algorithms in practical applications. To be specific, the pMCMC scheme we have assessed is constructed around a random walk Metropolis algorithm, but other MCMC schemes can be adopted; hence it is possible to further optimize its performance. The same is true for the NPMC

Initialization Set $t = 0$, $r = 0$; select the interval length T and an initial population vector $\mathbf{x}(0)$.

Iterative steps:

1. Compute the instantaneous rates $h_k(t)$ and $h_0(t) = \sum_{k=1}^K h_k(t)$. The probability of reaction k is $p_k(t) = h_k(t)/h_0(t)$.
2. Generate two random numbers $s_1, s_2 \sim \mathcal{U}[0, 1]$.
3. Compute the waiting time to the next reaction as $\tau(t) = (1/h_0(t)) \ln(1/s_1)$.
4. Select the reaction type according to s_2 and the probability of each reaction $p_k(t)$.
5. Update the time index $t = t + \tau(t)$, and the reaction index $r = r + 1$.
6. Adjust the populations of the species $\mathbf{x}(t)$ according to the reaction occurred.
7. If $t < T$ go to step 1.

ALGORITHM 3: Gillespie algorithm [18].

Initialization ($n = 0$): Draw a collection of J samples $\{\mathbf{x}_0^{(j)}\}_{j=1}^J \sim p(\mathbf{x}_0)$.

Recursive step ($n = 1, \dots, N$):

1. Draw $\{\mathbf{x}_n^{(j)}\}_{j=1}^J \sim p(\mathbf{x}_n | \mathbf{x}_{n-1}^{(j)}, \boldsymbol{\theta})$ using the Gillespie algorithm (or a diffusion approximation).
2. Construct $\mathbf{x}_{1:n}^{(j)} = [\mathbf{x}_{1:n-1}^{(j)\top}, \mathbf{x}_n^{(j)\top}]^\top$.
3. Compute normalized IWs $\omega_n^{(j)*} = p(\mathbf{y}_n | \mathbf{x}_n^{(j)})$, $\omega_n^{(j)} = \omega_n^{(j)*} / \sum_{l=1}^J \omega_n^{(l)*}$, $j = 1, \dots, J$.
4. Resample J times with replacement from $\{\mathbf{x}_{1:n}^{(j)}\}_{j=1}^J$ according to the weights $\{\omega_n^{(j)}\}_{j=1}^J$ to yield $\{\tilde{\mathbf{x}}_{1:n}^{(j)}\}_{j=1}^J$.

ALGORITHM 4: Standard particle filter [19].

method, however. In this case, the step to optimize in practice is the choice of the importance function. In our comparison we have used multivariate Gaussian proposals (because they are straightforward to fit) but other, more efficient proposals can indeed be used within the proposed framework. Also note that a computationally simpler approach to inference in some SKMs is the use of sequential Monte Carlo approximate Bayesian computation (SMC-ABC) techniques [26, 27]. In the SMC-ABC framework the state-space model (for \mathbf{x}_n) is only used for forward simulation. There is no attempt to approximate the likelihood of $\boldsymbol{\theta}$, which is replaced by some simple summary statistic. SMC-ABC methods are attractive when computational cost is a major concern, but they can be expected to deliver an inferior performance (as they do not fully exploit the probabilistic model, when it is available).

Appendix

A. Gillespie Algorithm

The Gillespie algorithm [18], which is displayed in Algorithm 3, allows generating exact forward simulations of arbitrary SKMs, by drawing samples from the transition density $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{c})$, $n = 1, \dots, N$, given a set of rate parameters \mathbf{c} and an initial population \mathbf{x}_0 . The algorithm can be run up to a number of reactions or for a given time interval T .

B. Sequential Monte Carlo Approximation of

$p(\mathbf{x} | \boldsymbol{\theta}, \mathbf{y})$ and $p(\mathbf{y} | \boldsymbol{\theta})$

In this appendix we provide details on the approximation of the posterior $p(\mathbf{x} | \boldsymbol{\theta}, \mathbf{y})$ and the likelihood $p(\mathbf{y} | \boldsymbol{\theta})$. For a

given vector of log-rate parameters $\boldsymbol{\theta}$, the standard particle filter (see, e.g., [19]) displayed in Algorithm 4 is run.

An approximation of the posterior $p(\mathbf{x} | \boldsymbol{\theta}, \mathbf{y}) d\mathbf{x}$ may be constructed from the final set of samples $\mathbf{x}_{1:N}^{(j)} = \mathbf{x}^{(j)}$ and weights $\omega_N^{(j)}$ as the discrete random measure

$$\tilde{p}^J(d\mathbf{x} | \boldsymbol{\theta}, \mathbf{y}) = \sum_{j=1}^J \omega_N^{(j)} \delta_{\mathbf{x}^{(j)}}(d\mathbf{x}). \quad (\text{B.1})$$

The likelihood $p(\mathbf{y} | \boldsymbol{\theta})$ can be approximated in turn as

$$\tilde{p}^J(\mathbf{y} | \boldsymbol{\theta}) = \prod_{n=1}^N \frac{1}{J} \sum_{j=1}^J p(\mathbf{y}_n | \mathbf{x}_n^{(j)}). \quad (\text{B.2})$$

In order to obtain a sample from the approximation $\tilde{p}^J(d\mathbf{x} | \boldsymbol{\theta}, \mathbf{y})$ in the pMCMC or the NPMC schemes, we just draw a sample out of the set $\{\mathbf{x}^{(j)}\}_{j=1}^J$ according to their IWs $\omega_N^{(j)}$.

Data Availability

All used data are synthetic. The code is freely available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research has been partially supported by the Spanish Ministry of Economy and Competitiveness (Projects

TEC2015-69868-C2-1-R ADVENTURE and TEC2017-86921-C2-1-R CAIMAN). Inés P. Mariño also acknowledges support from the grant of the Ministry of Education and Science of the Russian Federation Agreement no. 074-02-2018-330.

References

- [1] D. J. Wilkinson, *Stochastic Modelling for Systems Biology*, vol. 44, CRC press, 2006.
- [2] R. J. Boys, D. J. Wilkinson, and T. B. Kirkwood, “Bayesian inference for a discretely observed stochastic kinetic model,” *Statistics and Computing*, vol. 18, no. 2, pp. 125–135, 2008.
- [3] P. Milner, C. S. Gillespie, and D. J. Wilkinson, “Moment closure based parameter inference of stochastic kinetic models,” *Statistics and Computing*, vol. 23, no. 2, pp. 1–9, 2013.
- [4] D. J. Wilkinson, “Parameter inference for stochastic kinetic models of bacterial gene regulation: a Bayesian approach to systems biology,” in *Bayesian statistics*, vol. 9, pp. 679–706, Oxford Univ. Press, Oxford, UK, 2011.
- [5] A. Golightly and D. J. Wilkinson, “Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo,” *Interface Focus*, vol. 1, no. 6, pp. 807–820, 2011.
- [6] C. Andrieu, A. Doucet, and R. Holenstein, “Particle Markov chain Monte Carlo methods,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.
- [7] A. Golightly and D. J. Wilkinson, “Bayesian inference for stochastic kinetic models using a diffusion approximation,” *Biometrics: Journal of the International Biometric Society*, vol. 61, no. 3, pp. 781–788, 2005.
- [8] M. Komorowski, B. Finkenstädt, C. V. Harper, and D. A. Rand, “Bayesian inference of biochemical kinetic parameters using the linear noise approximation,” *BMC Bioinformatics*, vol. 10, no. 1, p. 343, 2009.
- [9] A. Golightly, D. A. Henderson, and C. Sherlock, “Efficient particle MCMC for exact inference in stochastic biochemical network models through approximation of expensive likelihoods,” *Statistics and Computing*, vol. 25, no. 5, pp. 1039–1055, 2012.
- [10] V. Stathopoulos and M. A. Girolami, “Markov chain Monte Carlo inference for Markov jump processes via the linear noise approximation,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical & Engineering Sciences*, vol. 371, no. 1984, Article ID 20110541, 21 pages, 2013.
- [11] A. Golightly, D. A. Henderson, and C. Sherlock, “Delayed acceptance particle MCMC for exact inference in stochastic kinetic models,” *Statistics and Computing*, vol. 25, no. 5, pp. 1039–1055, 2015.
- [12] C. C. Drovandi, M. T. Moores, and R. J. Boys, “Accelerating pseudo-marginal MCMC using Gaussian processes,” *Computational Statistics & Data Analysis*, vol. 118, pp. 1–17, 2018.
- [13] O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert, “Population Monte Carlo,” *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 907–929, 2004.
- [14] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Míguez, and P. M. Djuric, “Adaptive Importance Sampling: The past, the present, and the future,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 60–79, 2017.
- [15] T. Bengtsson, P. Bickel, and B. Li, “Curse-of-dimensionality revisited: collapse of the particle filter in very large scale systems,” *Probability and statistics: Essay in honour of David A. Freedman*, vol. 2, pp. 316–334, 2008.
- [16] E. Koblents and J. Miguez, “A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models,” *Statistics and Computing*, pp. 1–19, 2013.
- [17] J. Míguez, I. P. Mariño, and M. A. Vázquez, “Analysis of a nonlinear importance sampling scheme for Bayesian parameter estimation in state-space models,” *Signal Processing*, vol. 142, pp. 281–291, 2018.
- [18] D. T. Gillespie, “Exact stochastic simulation of coupled chemical reactions,” *The Journal of Physical Chemistry C*, vol. 81, no. 25, pp. 2340–2361, 1977.
- [19] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*, Springer Verlag, New York, NY, USA, 2001.
- [20] V. Volterra, “Fluctuations in the abundance of a species considered mathematically,” *Nature*, vol. 118, no. 2972, pp. 558–560, 1926.
- [21] C. Sherlock, A. H. Thiery, G. O. Roberts, and J. S. Rosenthal, “On the efficiency of pseudo-marginal random walk Metropolis algorithms,” *The Annals of Statistics*, vol. 43, no. 1, pp. 238–275, 2015.
- [22] E. Koblents, J. Miguez, M. A. Rodriguez, and A. M. Schmidt, “A nonlinear population Monte Carlo scheme for the Bayesian estimation of parameters of α -stable distributions,” *Computational Statistics & Data Analysis*, vol. 95, pp. 57–74, 2016.
- [23] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2nd edition, 2004.
- [24] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [25] L. Martino, V. Elvira, and F. Louzada, “Effective sample size for importance sampling based on discrepancy measures,” *Signal Processing*, vol. 131, pp. 386–401, 2017.
- [26] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. H. Stumpf, “Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems,” *Journal of the Royal Society Interface*, vol. 6, no. 31, pp. 187–202, 2009.
- [27] K. Csilléry, M. G. B. Blum, O. E. Gaggiotti, and O. François, “Approximate Bayesian Computation (ABC) in practice,” *Trends in Ecology & Evolution*, vol. 25, no. 7, pp. 410–418, 2010.

