

Scalable inference for Markov processes with intractable likelihoods

Jamie Owen · Darren J. Wilkinson · Colin S. Gillespie

Received: 15 March 2014 / Accepted: 16 October 2014 / Published online: 1 November 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract Bayesian inference for Markov processes has become increasingly relevant in recent years. Problems of this type often have intractable likelihoods and prior knowledge about model rate parameters is often poor. Markov Chain Monte Carlo (MCMC) techniques can lead to exact inference in such models but in practice can suffer performance issues including long burn-in periods and poor mixing. On the other hand approximate Bayesian computation techniques can allow rapid exploration of a large parameter space but yield only approximate posterior distributions. Here we consider the combined use of approximate Bayesian computation and MCMC techniques for improved computational efficiency while retaining exact inference on parallel hardware.

Keywords ABC · Particle MCMC · Markov processes · Intractable likelihood

1 Introduction

Stochastic kinetic models describe the probabilistic evolution of a dynamical system. Motivated by the need to incorporate intrinsic stochasticity in the underlying mechanics of the systems, stochastic models are increasingly used to model a wide range of real-world problems, particularly, but not limited to, computational systems biology, predator–prey population models, and single species population development (Kitano 2002; Boys et al. 2008; Gillespie and Golightly 2010). Sys-

tems are governed by a network of reactions which change the state of the system by a discrete amount, and hence are most naturally represented by a continuous time Markov process on a discrete state space. Typically, exact realisations from the model are obtained using the direct method (Gillespie 1977). There are a number of faster, but approximate algorithms, such as the diffusion approximation, moment closure, or hybrid simulation strategies (Gillespie 2000, 2009; Salis and Kaznessis 2005).

Our goal is to perform statistical inference for the parameters that govern the dynamics, where data are partially missing and prior knowledge of reaction rates may be poor. Likelihood functions for such problems are rarely analytically tractable but it is possible to leverage particle Markov chain Monte Carlo (pMCMC) methods to perform exact inference (Golightly and Wilkinson 2011; Andrieu et al. 2010). Typically, pMCMC algorithms exhibit a high computational cost, since at each iteration we run a sequential Monte Carlo (SMC) filter, which requires multiple forward simulations. This cost can become overly prohibitive if convergence is slow.

Approximate Bayesian computation (ABC) techniques, allow posterior inference to be made for problems where evaluation of the likelihood function is unavailable, relying heavily on the ability to simulate from the model (Tavare et al. 1997; Pritchard et al. 1999; Beaumont et al. 2002). An approximation to the true posterior distribution of interest is made via samples of parameter vectors, θ , that yield simulated data deemed “close” to the observed data. Here the definition of close is controlled by a tolerance, ϵ , and distance function $\rho(\cdot)$, hence retained samples are from $\pi(\theta | \rho(\mathcal{D}, \mathcal{D}^*) \leq \epsilon)$. The simple rejection algorithm typically leads to high numbers of proposed parameter vectors being rejected for meaningful tolerance values (Pritchard et al. 1999). Further developments within this framework lead

R-package: <https://github.com/csgillespie/abcpmcmc>.

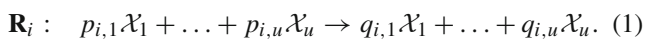
J. Owen (✉) · D. J. Wilkinson · C. S. Gillespie
School of Mathematics & Statistics, Newcastle University,
Newcastle upon Tyne, UK
e-mail: j.r.owen@ncl.ac.uk

to MCMC schemes which avoid calculation of the likelihood, as well as sequential Monte Carlo algorithms, which typically exhibit better acceptance rates than the simple rejection sampler (Marjoram et al. 2003; Del Moral et al. 2006; Sisson et al. 2007; Toni et al. 2009). Such techniques have successfully been applied to stochastic kinetic models for approximate inference (Drovandi and Pettitt 2011; Fearnhead and Prangle 2012).

In this article, we explore a scenario where we desire exact posterior inference for model rate parameters, while exploiting parallel hardware. By coupling scalable, approximate and exact techniques, we are able to increase computational efficiency of posterior sampling via parallel particle MCMC and improve convergence properties of chains that suffer expensive burn-in periods to further increase efficiency.

2 Stochastic kinetic models

Consider a network of reactions involving a set of u species $\mathcal{X}_1, \dots, \mathcal{X}_u$ and v reactions $\mathbf{R}_1, \dots, \mathbf{R}_v$ where each reaction \mathbf{R}_i is given by



Let P be the $v \times u$ matrix of pre-reaction coefficients of reactants $p_{i,j}$, similarly Q , the $v \times u$ matrix of post-reaction coefficients of products. The stoichiometry matrix, defined as

$$S = (Q - P)', \quad (2)$$

is a useful way to encode the structure of the reaction network. Also define X_t to be the vector $(X_{1,t}, \dots, X_{u,t})$ denoting the number of species \mathcal{X} present at time t .

Each reaction \mathbf{R}_i is assumed to have an associated rate constant, θ_i , and a hazard function $h_i(X_t, \theta_i)$ which gives the overall propensity for a type i reaction to occur. The form of $h_i(X_t, \theta_i)$ in many cases can be considered as arising from interactions between components in a well mixed population. This leads to reactions following the law of mass action kinetics where the hazard function for a reaction of type i takes the form

$$h_i = \theta_i \prod_{j=1}^u \binom{X_{j,t}}{p_{i,j}}. \quad (3)$$

If $\theta = (\theta_1, \dots, \theta_v)$ and $h(X_t, \theta) = (h_1(X_t, \theta), \dots, h_v(X_t, \theta))$ then values for θ and X_0 complete full specification of this Markov process.

Algorithm 1 The Direct method

1. Set $t = 0$. Initialise the rate constants θ and initial states X_0 .
2. Calculate the hazard functions $h(X_t, \theta)$ and $h_0(X_t, \theta) = \sum_i^v h_i(X_t, \theta_i)$.
3. Set $t = t + \delta t$ where

$$\delta t \sim \text{Exp}(h_0(X_t, \theta)).$$
4. Simulate the reaction index $j \in (1, \dots, v)$ with probabilities p_j

$$p_j = \frac{h_j(X_t, \theta)}{h_0(X_t, \theta)}.$$
5. Set $X_{t+\delta t} = X_t + S[j]$ where $S[j]$ is the j^{th} column of the stoichiometry matrix S .
6. If $t < T$ return to 2.

2.1 The direct method

Exact simulations of the evolution of such a system can be obtained via the Direct method (Gillespie 1977). Algorithm 1 describes the procedure for exact forward simulation of a stochastic kinetic model, given its stoichiometry matrix, S , a vector of reaction rates, θ , an initial state vector X_0 and an associated hazard function $h(X_t, \theta)$. Reactions simulated to occur via the Direct method incorporate the discrete and stochastic nature of the system since each reaction, which increments or decrements species levels by discrete amounts, are chosen probabilistically.

Less computationally expensive simulation algorithms such as the chemical Langevin equation (CLE) relax the restriction imposed by the discrete state space but retain the stochasticity of the underlying mechanics, giving approximate realisations of the progression of the species (Gillespie 2000). For the purpose of exact inference and this article we consider only exact realisations of the reaction network via the direct method. For a more in depth background into stochastic kinetic modelling see Wilkinson (2011).

We anticipate data, $\mathcal{D} = (d_0, d_1, \dots, d_T)$, to be a collection of noisy, possibly partial observations of the system $X = (X_0, X_1, \dots, X_T)$ at discrete time intervals.

3 Methods for Bayesian inference in intractable likelihood problems

Methods for likelihood-free Bayesian inference have become popular due to the increasing use of models in which the likelihood function is either deemed too computationally expensive to compute, or analytically unavailable. In particular, approximate Bayesian computation (ABC), and particle filtering methods have proven to be two classes of algorithms in which one can proceed with inference in a scenario where evaluation of the likelihood is unavailable, commonly termed likelihood-free inference.

Algorithm 2 ABC rejection sampler

1. Generate a candidate parameter vector $\theta^* \sim \pi(\theta)$.
2. Simulate a candidate data set $\mathcal{D}^* \sim \pi(\mathcal{D}|\theta^*)$.
3. Calculate a measure of distance between the candidate data, \mathcal{D}^* , and the observed data \mathcal{D} , $\rho(\mathcal{D}, \mathcal{D}^*)$.
4. Accept θ^* if $\rho(\cdot) < \epsilon$ for some predetermined, fixed, ϵ .
5. Go to 1.

3.1 Approximate Bayesian computation

The goal of approximate Bayesian computation techniques is to obtain a collection of samples from the posterior distribution $\pi(\theta|\mathcal{D})$ where the likelihood function $\pi(\mathcal{D}|\theta)$ is unavailable, due to either high computational cost or analytical intractability. Typically we rely on the assumption that simulation from the model given a set of parameters, θ , is relatively straightforward, as in Algorithm 1. Given a set of proposed parameter vectors we would ideally keep any such vectors which yield data simulations that are equal to the observations that we have. In reality however, we will typically match the simulated data to the observations perfectly with low probability. To avoid rejecting all proposals, we instead keep parameters that yield simulations deemed sufficiently close to the observations. We characterise a set of simulated data, \mathcal{D}^* as sufficiently close if, for a given metric, $\rho(\cdot)$, the distance between \mathcal{D}^* and observed data \mathcal{D} is below a tolerance threshold, ϵ . The simple rejection sampler is described in Algorithm 2.

Rather than leading to a sample from the exact posterior distribution, the samples instead have the distribution $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) < \epsilon)$. As $\epsilon \rightarrow 0$ this tends to the true posterior distribution if $\rho(\cdot)$ is a properly defined metric on sufficient statistics. Conversely as $\epsilon \rightarrow \infty$ one recovers the prior distribution on the parameters. Since it is usually the case that sufficient statistics are not available in this type of problem, further approximation can be made by choosing a set of summary statistics, $S(\cdot)$, which are not sufficient, but it is hoped describe the data well. A set of summary statistics is often used over the observed data in problems in which the dimension of data is large. This is to combat the well-known curse of dimensionality that leads to intolerably poor acceptance rates in this type of scheme. In this case we replace step 3 of Algorithm 2 with a step that first calculates $S(\mathcal{D}^*)$ then assigns distance according to $\rho(S(\mathcal{D}), S(\mathcal{D}^*))$. The criteria for forming a good choice of a set of informative summary statistics is beyond the scope of this article and we refer the reader to the following paper which summarises the current research in this area, Blum et al. (2013). Throughout this article we use a Euclidean metric on the simulated data points as we found that this gave sufficient acceptance rates. If a large value of ϵ is chosen, a large number of proposed candidates are accepted, but little is learnt about the

Algorithm 3 Sequential ABC

1. Initialise $\epsilon_0 > \epsilon_1 > \dots > \epsilon_T > 0$ and set the population indicator, $t = 0$.
2. Set particle indicator, $i = 1$.
3. If $t = 0$, sample $\theta^{**} \sim \pi(\theta)$
 Else sample θ^* from the previous population $\{\theta_{t-1}^{(i)}\}$ with weights w_{t-1} and perturb to obtain $\theta^{**} \sim K_t(\theta|\theta^*)$
 If $\pi(\theta^{**}) = 0$, return to 3.
 Simulate a candidate dataset $x^* \sim f(x|\theta^{**})$
 If $d(x_0, x^*) \geq \epsilon_t$, return to 3.
4. Set $\theta_t^{(i)} = \theta^{**}$ and calculate weight for particle $\theta_t^{(i)}$, $w_t^{(i)}$

$$w_t^{(i)} = \begin{cases} 1, & \text{if } t = 0 \\ \frac{\pi(\theta_t^{(i)})}{\sum_{j=1}^N w_{t-1}^{(j)} K_t(\theta_t^{(j)}|\theta_t^{(i)})}, & \text{if } t > 0 \end{cases}$$
 If $i < N$ set $i = i + 1$, go to 3
5. Normalise the weights, if $t < T$, set $t = t + 1$ and go to 2.

posterior distribution. Choosing a small tolerance yields a much better approximation to $\pi(\theta|\mathcal{D})$ but at the expense of poorer acceptance rates and hence increased computational cost. This rejection based approach has been included in a number of sampling techniques.

Within the context of this article we consider a sequential approach to ABC inference. An example of such a scheme, based on importance sampling, is described in Algorithm 3, see Toni et al. (2009) for details.

Algorithm 3 as described here allows us choice of an essentially arbitrary perturbation kernel $K_t(\cdot)$ for preventing degeneracy of particles, and the sequence of tolerances $\epsilon_0, \dots, \epsilon_T$. In practice the choice of $K_t(\cdot)$ will have an important effect on the overall efficiency of the algorithm. A perturbation which gives small moves will typically allow acceptance rates to remain higher, whilst exploring the space slowly. Conversely making large moves allows better exploration of the parameter space, but usually at the cost of reduced acceptance rates. We shall use the ‘optimal’ choice of a random walk perturbation kernel for this algorithm as detailed in Filippi et al. (2013). For a multivariate Gaussian random walk kernel at population t , we choose $\Sigma^{(t)}$ as

$$\Sigma^{(t)} \approx \sum_{i=1}^N \sum_{k=1}^{N_0} \omega^{(i,t-1)} \tilde{\omega}^{(k)} (\tilde{\theta}^{(k)} - \theta^{(i,t-1)}) (\tilde{\theta}^{(k)} - \theta^{(i,t-1)})' \tag{4}$$

where N is the number of weighted samples we have from the previous population $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) < \epsilon_{t-1})$ with weights ω , N_0 is the number of those samples, $\tilde{\theta}$, that satisfy $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) < \epsilon_t)$ with associated normalised weights $\tilde{\omega}$. This builds on the work of Beaumont et al. (2009).

The choice of sequence of tolerances $\epsilon_0, \dots, \epsilon_T$ also has a large effect on the efficiency of the algorithm. A sequence which decreases too slowly will yield high acceptance rates but convergence to the posterior will be slow. Whereas a

Algorithm 4 Metropolis Hastings

1. Initialise with a random starting value $\theta \sim \pi(\theta)$.
2. Propose a move to a new candidate $\theta^* \sim q(\theta^*|\theta)$.
3. Accept the move with probability

$$\min \left\{ 1, \frac{\pi(\mathcal{D}|\theta^*)\pi(\theta^*)q(\theta|\theta^*)}{\pi(\mathcal{D}|\theta)\pi(\theta)q(\theta^*|\theta)} \right\}, \quad (5)$$

else remain at θ .

4. Return to 2.

rapidly decreasing sequence typically yields poor acceptance rates. We will, throughout this article, use an adaptive choice of ϵ_t determined by taking a quantile of the distances at $(t-1)$. It should be noted that [Silk et al. \(2013\)](#) observe that there can be issues with convergence when using an adaptive tolerance based on quantiles. However the authors are unaware of a generally accepted solution to this problem.

3.1.1 ABC methods in parallel

Non MCMC-based ABC techniques, such as those described in Algorithms 2 and 3 are often amenable to parallelisation. Since all proposed parameters are independent samples from the same distribution, and the acceptance step for each does not depend on the previously accepted particle, the bulk of computation for a sequential ABC scheme can be run in parallel, greatly reducing overall CPU time required to obtain a sample from the target. Some thread communication is necessary as we move from one distribution in the sequence to the next, something that would be avoided if using the simple rejection sampler, but typically this is small in comparison to the work done in forward simulation.

Coding of a parallel ABC algorithm adds little complexity over a non-parallelised version. This is particularly true for the simple rejection sampler where one effectively just runs a sampler on each of N processors and no communication between threads is needed.

3.2 Markov chain Monte Carlo

Suppose interest lies in $\pi(\theta|\mathcal{D})$, and that we wish to construct a MCMC algorithm whose stationary distribution is exactly this posterior. Using an appropriate proposal function $q(\theta^*|\theta)$ we can construct a Metropolis Hastings algorithm to do this as described in Algorithm 4. This can often be impractical due to unavailability of the likelihood term, $\pi(\mathcal{D}|\theta)$.

In practice one normally chooses a proposal function $q(\theta^*|\theta)$ such that proposed moves from θ to θ^* are distributed symmetrically about θ giving rise to random walk Metropolis Hastings. Typical choices include a uniform $\mathcal{U}(\theta - \sigma, \theta + \sigma)$ or Gaussian $\mathcal{N}(\theta, \Sigma)$ distribution. In this article we shall con-

sider Gaussian innovations for random walk proposals. It has been shown that under various assumptions about the target, the optimal scaling for a random walk Metropolis algorithm using a Gaussian proposal, Σ_q , for a d dimensional distribution is

$$\Sigma_q = \frac{(2.38)^2}{d} \Sigma, \quad (6)$$

where Σ is the posterior covariance ([Roberts et al. 1997](#); [Roberts and Rosenthal 2001](#)). Typically, however, the posterior covariance is unavailable.

3.2.1 Parallel MCMC

MCMC algorithms are somewhat less amenable to parallelisation than rejection sampling algorithms. Essentially there are two options to be considered, parallelisation of a single MCMC chain or the construction of parallel chains. For an in depth discussion see [Wilkinson \(2006\)](#). Parallelisation of a single chain in many scenarios is somewhat difficult due to the inherently iterative nature of a Markov chain sampler and will not be discussed in detail here. Running parallel chains, however, is straightforward.

3.2.2 Parallel chains

In practice, chains initialised with an arbitrary starting point will be, after an appropriate burn-in period, essentially sampling from the target. In the context of a serial implementation there is still some debate over whether it is better to run a single long chain, or many shorter chains. The benefits of the single chain are that any burn in period is only suffered once, whereas the argument for multiple shorter chains is that one may better diagnose convergence to the stationary distribution. The argument changes when considering parallel implementation. Indeed, if burn-in periods are relatively short, running independent chains on separate processors can be a very time efficient way of learning about the distribution of interest.

Burn-in is still a potential limiter on the scaling of the performance to be had when employing parallel chains with the number of processors. The greater the burn-in period of a chain, the more time each processor has to waste computing samples that will eventually be thrown away. The theoretical speed up calculation given N processors to obtain n stored samples with a burn-in period b is

$$\text{Speed-up}(N) = \frac{b+n}{b+\frac{n}{N}}, \quad (7)$$

which is clearly limited for any $b > 0$, as $N \rightarrow \infty$ ([Wilkinson 2006](#)). A “perfect” parallelisation of multiple chains then is one in which we have no burn-in period, i.e initialise each

Algorithm 5 The boot–strap particle filter

Let \mathbf{X}_t^* denote a set of N particles, $\{(x_t^i, \pi_t^i) : i = 1, \dots, N\}$. The filter assumes fixed parameters and so we drop the θ from our notation.

1. Initialise at $t = 0$ and draw a set of N independent samples, $\mathbf{X}_t^* \sim \pi(X_0)$.
2. Sample a set of indices for candidates for forward simulation, I_t^i according to the weights π_t .
3. Simulate forward from the model the chosen paths, $x_{t+1}^i \sim \pi(x_{t+1}^i | x_t^{I_t^i})$.
4. Calculate weights, $w_{t+1}^i = p(d_{t+1} | x_{t+1}^i)$, and normalise to set $\pi_{t+1}^i = \frac{w_{t+1}^i}{\sum_{j=1}^N w_{t+1}^j}$.
5. Set $t := t + 1$.
6. If $t < T$, sample a new set of particles $\mathbf{X}_t^* \sim \pi(\mathbf{X}_t | D_{0:t})$ and return to 2.

Define $\hat{p}(d_t | D_{1:t-1}) = \frac{1}{N} \sum_{i=1}^N w_t^i$, then $\hat{p}(D_{1:t}) = \prod_{l=1}^t \hat{p}(d_l | D_{1:l-1})$.

chain with an independent draw from the target. The performance gain when run on N processors is then of factor N . Clearly this “perfect” situation will not be possible to implement in practice since we typically are unable to sample from the target.

3.2.3 Particle MCMC

Re-consider the Metropolis Hastings MCMC algorithm from Sect. 3.2. The ability to construct such a scheme relies on the evaluation of the likelihood function $\pi(D|\theta)$ among others. Crucially, the problems of interest discussed in this article are such that the likelihood term $\pi(D|\theta)$ is unavailable. However Andrieu and Roberts (2009) proposed a pseudo marginal MCMC approach to this problem. In the acceptance ratio of the MCMC scheme, replace the intractable likelihood, $\pi(D|\theta)$, with a non–negative Monte Carlo estimate. It can be shown that, provided $E[\hat{\pi}(D|\theta)] = \pi(D|\theta)$, the stationary distribution of the resulting Markov chain is exactly the distribution of interest. In the context of inference for state space models, it is natural to make use of sequential Monte Carlo techniques through a bootstrap particle filter, (Doucet et al. 2001), to obtain estimates of the likelihood $\hat{\pi}(D|\theta)$. The bootstrap particle filter for application to Markov processes is described in Algorithm 5.

Substituting the estimate of $\hat{\pi}(D|\theta)$ in place of the likelihood function yields a MH algorithm with acceptance probability

$$\min \left\{ 1, \frac{\hat{\pi}(D|\theta^*)\pi(\theta^*)q(\theta|\theta^*)}{\hat{\pi}(D|\theta)\pi(\theta)q(\theta^*|\theta)} \right\}. \tag{8}$$

It turns out that the particle filter’s estimate of marginal likelihood is unbiased, giving rise to an “exact approximate” pseudo-marginal MCMC algorithm. This use of a SMC sampler within a pseudo-marginal algorithm is a special case of the particle marginal Metropolis Hastings (PMMH) algo-

rithm described in Andrieu et al. (2010), which in general can be used to target the full joint posterior on the state and rate $\pi(\theta, X|\mathcal{D})$. In this article we consider only the pseudo-marginal special case. As discussed in section 3.2.1 this type of scheme does not lend itself well to parallelisation unless we have the opportunity to initialise chains with a sample from the posterior distribution $\pi(\theta|\mathcal{D})$. We recognise that rather sophisticated parallel particle filter algorithms which have the potential for increasing the speed of moves within a single chain are available. However the speed up is limited by necessary communication and synchronisation between processors and typically do not scale well.

The pseudo–marginal MCMC scheme requires specification of a number of particles, N , to be used in the boot–strap filter. It was noted, (Andrieu and Roberts 2009), that the efficiency of the scheme decreases as the variance of the estimated marginal likelihood increases. This can be overcome by increasing the value of N , albeit at an increased computational cost. An optimal choice of N was the subject of Pitt et al. (2012) and Doucet et al. (2012). The former suggest that N should be chosen such that the variance in the estimated log–likelihood is around 1 while the latter show that the efficiency penalty is small for values between 0.25 and 2.25.

3.3 Combined use of ABC and pMCMC

Since it is typically not possible to initialise a MCMC chain with a draw from the desired target, we propose an approach to parallel MCMC by choosing initial parameter vectors according to samples from an approximate posterior distribution. The intuition is that if we have a reasonable approximation to the target of interest, samples from the approximation will closely match those from $\pi(\theta|\mathcal{D})$. Because of this we expect that after a very short burn–in we are sampling from the desired target. Hence we are approaching the scenario of near perfect parallelisation of MCMC. Clearly the better the approximation the shorter the burn-in for each chain will be.

If we first run an ABC scheme targeting an approximation to this distribution, we can exploit parallel hardware to eliminate a large region of prior parameter space very quickly. Then take a set of N independent samples from $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) < \epsilon)$ to initialise N independent, parallel, pMCMC chains each targeting the exact posterior distribution $\pi(\theta|\mathcal{D})$.

We can in some sense consider this process of obtaining a sample from an ABC posterior as an artificial burn–in period. Crucially however, since the ABC algorithm yields a large set of samples from $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) \leq \epsilon)$ the computational price of performing the artificial burn–in has only to be paid once and can be parallelised. With a set of samples we can initialise an arbitrary number of MCMC chains each with an

Algorithm 6 Hybrid ABC pMCMC

1. Run an ABC algorithm targeting $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) \leq \epsilon_T)$.
2. Initialise multiple MCMC chain with a sample $\theta_0 \sim \pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) \leq \epsilon_T)$ and set $i = 1$.
3. Propose $\theta^* \sim q(\theta^*|\theta)$ for $q(\cdot)$ a Gaussian random walk kernel, $\mu_q = \theta, \Sigma_q = \frac{(2.38)^2}{d} \Sigma_{ABC}$.
4. Approximate $\pi(\mathcal{D}|\theta^*)$ via a bootstrap particle filter, $\hat{\pi}(\mathcal{D}|\theta^*)$ where the number of particles is chosen as in (10).
5. With probability

$$\min \left\{ 1, \frac{\hat{\pi}(\mathcal{D}|\theta^*)\pi(\theta^*)q(\theta|\theta^*)}{\hat{\pi}(\mathcal{D}|\theta)\pi(\theta)q(\theta^*|\theta)} \right\}$$

set $\theta^{(i)} = \theta^*$ else set $\theta^{(i)} = \theta^{(i-1)}$.

6. Set $i := i + 1$ and return to 3.

independent parameter vector θ_0 which comes from a distribution close to the target $\pi(\theta|\mathcal{D})$.

Since ABC is being used only for initialisation, it is not used as a prior for the MCMC algorithm, nor does it form part of the proposal. Hence, the ABC approximation does not affect the exactness of the pMCMC target.

3.4 Random walk pMCMC using ABC

We now refer back to the optimal choice of Gaussian random walk kernel of Roberts and Rosenthal (2001) mentioned in Sec. 3.2. Since we hope that we start with a reasonable approximation to the true posterior, we likewise consider that the covariance of a sample from $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) < \epsilon)$ denoted here Σ_{ABC} , will be close to the covariance of true posterior samples. This allows us to use the approximate posterior covariance structure as an informative tool for calibrating the random walk innovations. In most MCMC applications, since the true posterior variance is unknown, Σ_q requires tuning, typically from pilot runs. In the case of using Σ_{ABC} we hope that we can also remove the necessity to tune Σ_q . In practice we take

$$\Sigma_q = \frac{(2.38)^2}{d} \Sigma_{ABC}. \tag{9}$$

In addition we use our approximation to automatically tune the number of particles required for the particle filter. In practice we take the average of the approximate distribution, $\bar{\theta}_{ABC}$ to calculate the number of particles required to satisfy

$$\text{Var}(\log(\hat{\pi}(\mathcal{D}|\bar{\theta}_{ABC}))) \simeq 2, \tag{10}$$

in line with Sherlock et al. (2013). We do this by running the particle filter a number of times with varying numbers of particles until the condition is satisfied. The hybrid ABC pMCMC algorithm is outlined in Algorithm 6.

4 Applications

4.1 The Lotka–Volterra system

The Lotka–Volterra predator–prey system is a simple stochastic kinetic model (Lotka 1952; Volterra 1926). The system is characterised by a set of three reactions detailing interactions between two species; prey X and predators Y :



The reactions \mathbf{R}_1 , \mathbf{R}_2 and \mathbf{R}_3 can be thought of as prey birth, an interaction resulting in a prey death and predator birth, and predator death respectively. Whilst this is a relatively simple example of a model in this context, it highlights many of the difficulties associated with more complex systems. We summarise the system by its stoichiometry matrix, S and hazard function $h(\mathbf{X}_t, \theta)$:

$$S = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}, \quad h(\mathbf{X}_t, \theta) = (\theta_1 X_t, \theta_2 X_t Y_t, \theta_3 Y_t). \tag{12}$$

Figure 1(a) is a set of synthetic, noisy observations of the two species simulated using the Gillespie algorithm with true parameter values $\log(\theta) = (0, -5.30, -0.51)$ together with a Gaussian measurement error structure, $d_{j,t} \sim \mathcal{N}(X_{j,t}, 10^2)$.

4.1.1 pMCMC for Lotka–Volterra

Exact parameter inference is possible for this system using a pMCMC scheme; provided the chain is initialised near the posterior mode (Wilkinson 2011). However under poor initialisation, a pMCMC scheme will perform very badly.

Suppose we have little prior knowledge on the rate parameters,

$$\log(\theta_i) \sim \mathcal{U}(-8, 8), \quad i = 1, 2, 3. \tag{13}$$

We take a prior distribution for the initial state X_0 on each individual species as Poisson distributions with rate parameter equal to the true initial conditions,

$$x_{1,0} \sim \text{Pois}(50), \quad x_{2,0} \sim \text{Pois}(100). \tag{14}$$

Further we assume that the variance, σ^2 , of the Gaussian measurement error is known.

Using a Gaussian random walk proposal on $\log(\theta)$, $q(\log(\theta^*)|\log(\theta))$ we construct a MH algorithm using a bootstrap particle filter, targeting $\pi(\theta|\mathcal{D})$ as in Algorithm 4.

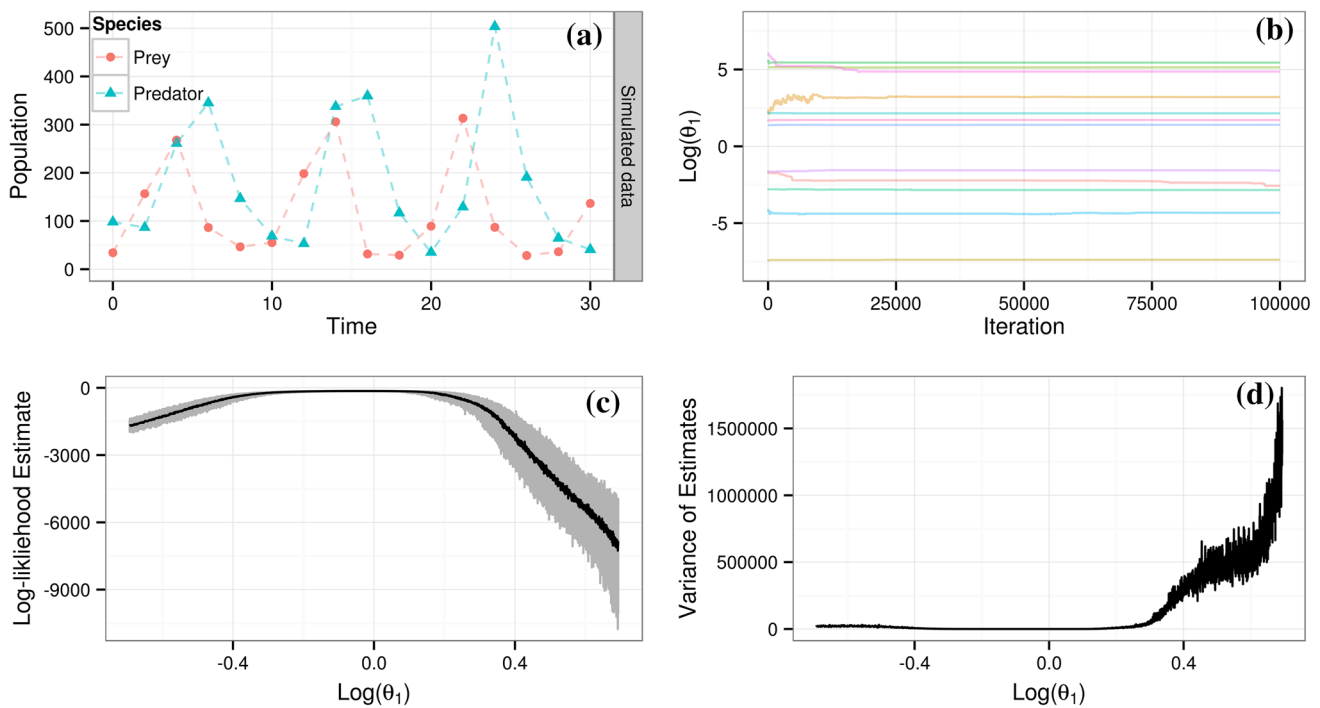


Fig. 1 Investigating computational issues with pMCMC for the Lotka–Volterra model defined in Sect. 4.1. **a** The true underlying synthetic data set. Species are observed at discrete time points and corrupted with $N(0, 10^2)$ noise. **b** Twelve trace plots of θ_1 from pMCMC chains initialised with random draws from the prior (see expression 13). The

chains fail to explore the space. **c** shows the median and 95% interval for estimates of the log-likelihood from the particle filter for varying θ_1 close to the true value, for θ_2 and θ_3 fixed at the true values. **d** shows that the variance of log-likelihood estimates increases away from the true values

Figure 1(b) shows that when initialising the chain using random draws from the weakly informative prior, the chains do not explore the space, failing to converge. Further investigation shows why this is happening. Figure 1c, d show that away from the true values, the variance of the log-likelihood estimates from the bootstrap particle filter increases sharply. Figure 1 (c) shows the 95% interval of log-likelihood estimates over 100 runs of a particle filter using 150 particles. Figure 1 (d) shows that even close to the true value, $\theta_1 = 1$, the variance of the log-likelihood estimates increases dramatically with a small move. In both Fig. 1c, d the other two parameters were kept fixed at the true values. When we are in a region with negligible likelihood the chain has a tendency to stick as a result of the variability in the likelihood estimates. On top of this, small proposed moves can lead to large variation in the estimated likelihood, leading to poor exploration of the tails in the posterior distribution. Whilst we are guaranteed to eventually converge to the stationary distribution, the required computational cost, without carefully thought out initialisation, could be very high. We note that this is not a failure of the theory or algorithm, but a consequence of the sensitivity to initialisation of parameter values experienced in this type of model.

We therefore apply the proposed ABC initialisation for a “perfect” parallel pMCMC scheme.

4.1.2 Results for the Lotka–Volterra model using a hybrid approach

We use a sequence of seven distributions $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) < \epsilon_t)$, $t = 0, \dots, 6$ to obtain our approximation to the posterior. For each population t we take ϵ_t as the 0.3 quantile of the distribution of distances from the samples at $t - 1$ and propose candidates until we reach a sample of 1,000 for each t . At each stage we perform the forward simulation of the model given proposed parameter values in parallel. Figure 2 (a) shows summaries for log of the reaction 1 rate parameter $\log(\theta_1)$ for each of the distributions in the series through the sequential ABC. The distributions quickly remove a large region of space that, had we sampled from the prior distribution to initialise the chain, are likely to have been poor starting points. The scheme converges around the true value $\log(\theta_1) = 0$. Given the sample from $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) < \epsilon_6)$ we initialise eight MCMC chains on separate processors with random draws. The results in Fig. 2(b, c, d) are then, the 20,000 pooled samples from the eight independent parallel chains, each of which has been thinned by a factor of 100 to give 2,500 samples. Each chain is sampling from the same stationary distribution as seen in the trace plot for θ_1 , Fig. 2 (c), and mixing is good, Fig. 2 (b). Further the true parameter values, $\log(\theta) = (0, -5.30, -0.51)$, used to simulate the

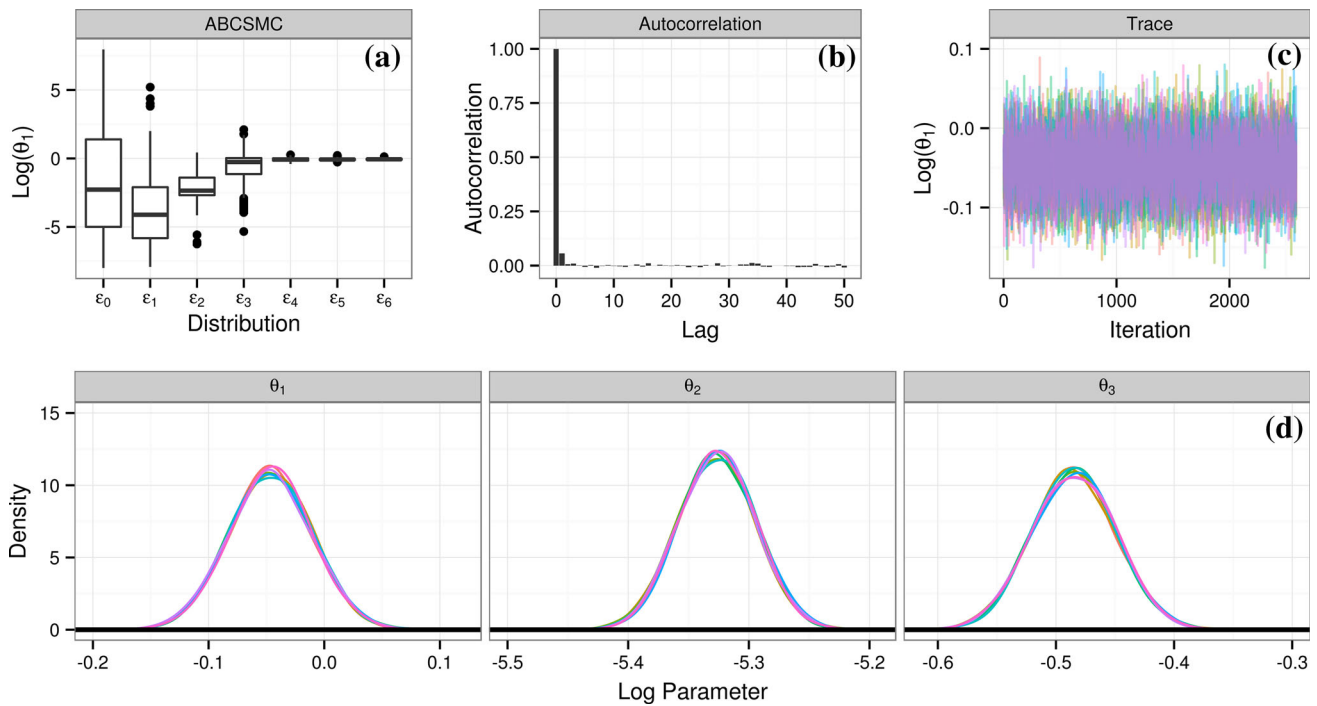


Fig. 2 Analysis of results for the synthetic data for the Lotka–Volterra model. **a** are successive distributions of $\log(\theta_1)$ in the sequential ABC scheme, Algorithm 3. **b** shows autocorrelations for chain 1, representative of each of the parallel chains, and **c** are traces of eight parallel MCMC chains for $\log(\theta_1)$. Note that each chain is sampling from

the same stationary distribution and mixing appears good. **d** are the posterior densities for $\log(\theta)$, each chain leads to a posterior density plot that is very close to that of every other chain. True values $\log(\theta) = (0.0, -5.30, -0.51)$ are well identified

data are well identified within the posterior densities, Fig. 2 (d).

4.2 Real-data problem: aphid model

Next we consider a model of aphid population dynamics as proposed in Matis et al. (2007). The system can be represented by the following reactions:



and summarised in terms of its stoichiometry matrix S , and hazard function $h(\mathbf{X}_t, \theta)$,

$$S = \begin{pmatrix} 1 & -1 \\ 1 & 0 \end{pmatrix}, \quad h(\mathbf{X}_t, \theta) = (\lambda N_t, \mu N_t C_t), \tag{16}$$

$$\mathbf{X}_0 = (N_0, C_0), \quad \theta = (\lambda, \mu).$$

N_t and C_t are the numbers of aphids alive at time t and the cumulative number of aphids that have lived up until time t respectively. In the first reaction, when a new aphid is introduced into the system we get an increase in both the current number of aphids and the cumulative count. When an aphid is removed the number of aphids decreases by one but

the cumulative count remains the same. Aphid death is considered to be proportional not only to the number of aphids currently in the system, but also to the cumulative count, representing the idea that over time they are making their own environment less habitable.

Given initial conditions $\mathbf{X}_0 = (N_0, C_0)$ and a set rate of parameters θ , we can simulate from the model. However the difficulty in practice with the model presented here is that we will never have observations of the cumulative count. Observations then are noisy, discrete time measurements of just a single species of the system.

4.2.1 Aphid data

We now consider the data described in Matis et al. (2008) consisting of cotton-aphid counts for twenty seven treatment-block combinations. The treatments consisted of three nitrogen levels (blanket, variable and none), three irrigation levels (low, medium and high) and three blocks. The sampling times of the data are $t = 0, 1.14, 2.29, 3.57, 4.57$ weeks, or every seven to eight days. We restrict our investigation to a single treatment combination, three data sets with blanket nitrogen level and low irrigation. If we denote the block by

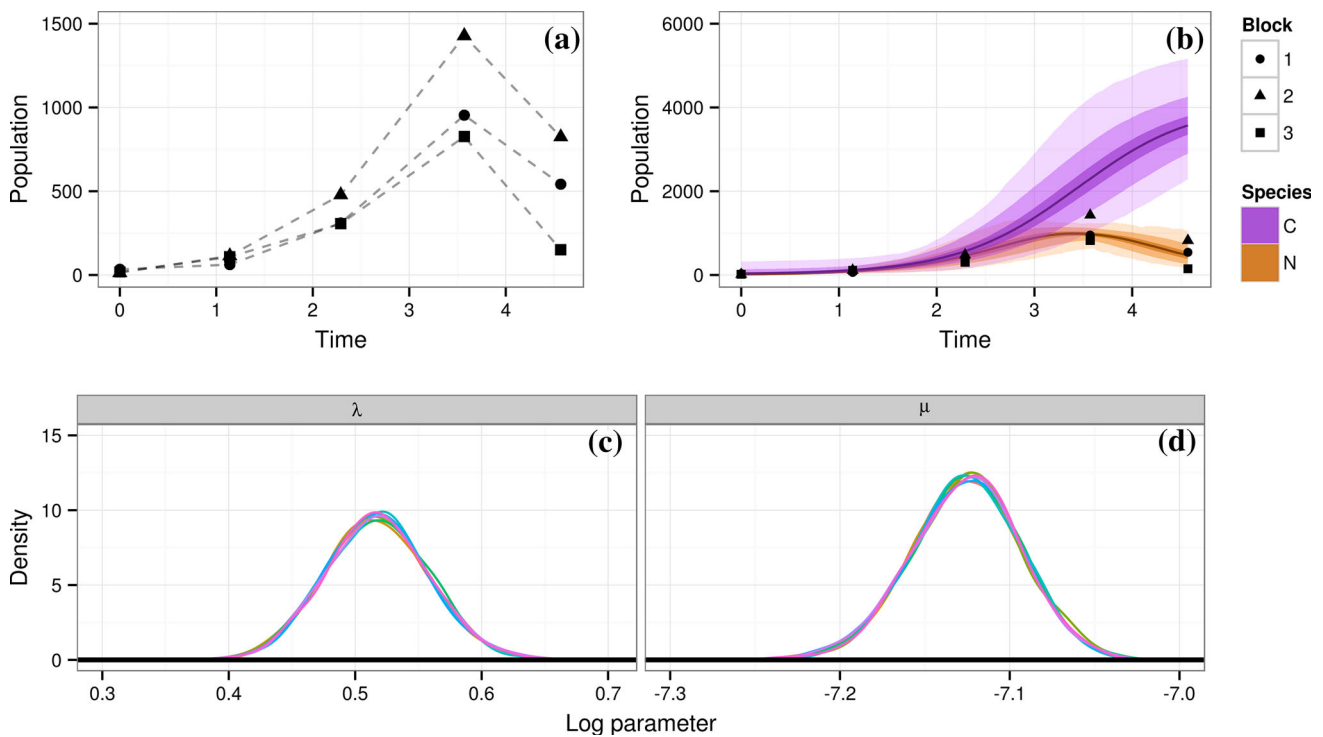


Fig. 3 Analysis of the real data for the aphid growth model. **a** There are the three data sets of aphid counts each with of five observations. **b** The posterior predictive model fit given a sample from the collection of

posterior densities. **c, d** output from each MCMC chain, highlight that we are sampling from the same stationary distribution

$i \in \{1, 2, 3\}$ then the data \mathcal{D}_i is the number of aphids, N , in block i at each time t . The data are plotted in Fig. 3(a).

We make the assumption that the counts are observed with error such that

$$d_t \sim \text{Pois}(x_t), \tag{17}$$

and use a set of weakly informative priors on the rate parameters θ

$$\log(\theta_i) \sim \mathcal{U}(-8, 8), \quad i = 1, 2. \tag{18}$$

We place a prior of the form

$$C_0 = N_0 + g, \quad g \sim \text{Geom}(0.03), \tag{19}$$

to reflect the fact that we are unable to measure C_0 .

We treat the three sets of observations as repeats of the same experiment. Likelihood estimates are obtained by running a particle filter for each of the three data sets and taking the product of the individual estimates. A full treatment of all twenty-seven data sets using a fixed effects model can be found in Gillespie and Golightly (2010). We consider the initial aphid counts to be the true values, as in Gillespie and Golightly (2010), on the basis that there should be no error in counting such small populations.

4.2.2 Results for the aphid growth model

We use the same criteria for the choice of ϵ_t for the ABC section of the inference as with the Lotka–Volterra model in 4.1.2, namely the 0.3 quantile of the distribution of distances. A sequence of five distributions gives us 1000 samples from $\pi(\theta|\rho(\mathcal{D}, \mathcal{D}^*) < \epsilon_4)$ which we use to initialise eight parallel chains. We record 20,000 samples from the exact target posterior $\pi(\theta|\mathcal{D})$ after appropriate thinning. Figure 3(c and d) shows the analysis of the MCMC chains. Again we find that each chain is sampling from the same target and posterior densities are very close from all eight chains. Figure 3(b) shows posterior predictive quantiles given a sample from the pooled posterior distribution. Model fit appears to be reasonable. The results are consistent with those seen in Gillespie and Golightly (2010) where they assume that observations are made without error and make use of an approximate simulation algorithm for realisations of the model.

4.3 Gene expression

Finally, we consider a simple gene regulation model characterised by three species (DNA, mRNA, denoted R, and protein, P) and four reactions. The reactions represent transcription, mRNA degradation, translation and protein degradation

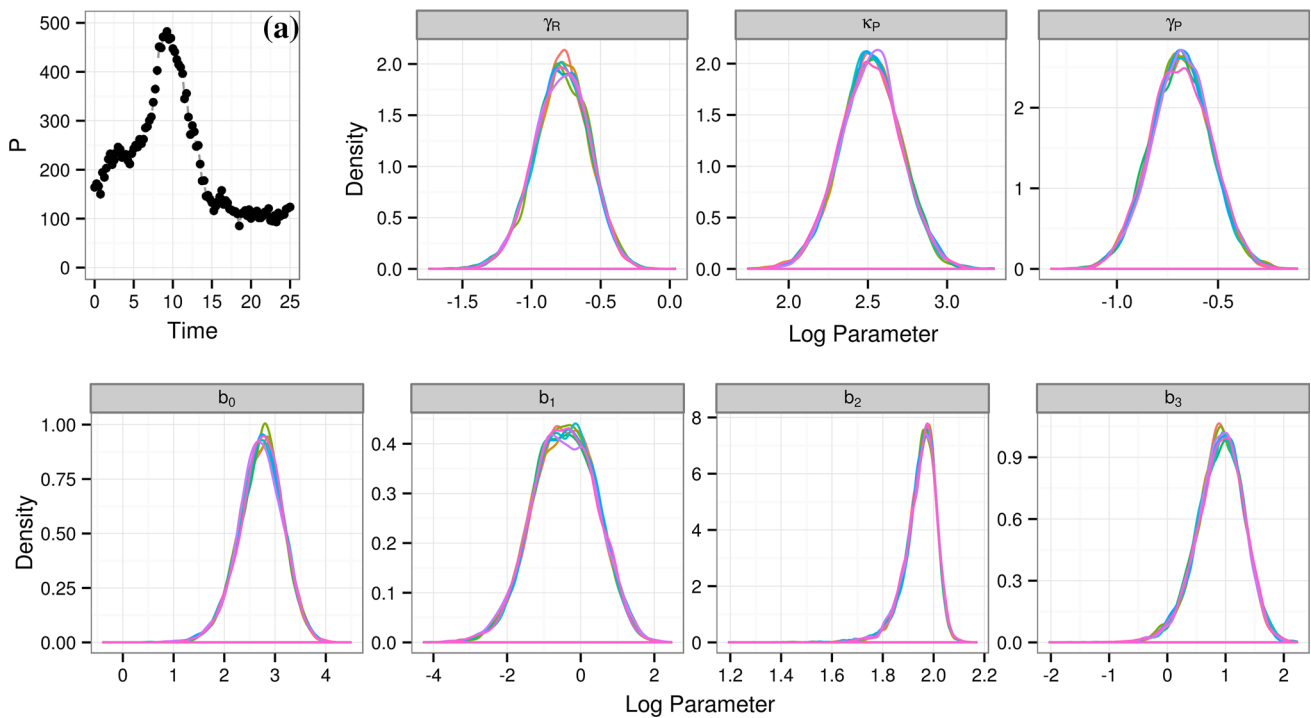
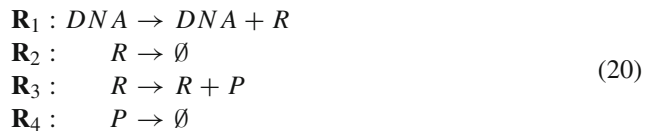


Fig. 4 **a** is the noisy pseudo-data for the Protein levels in the model. The other plots show the individual densities from pMCMC chains after appropriate thinning having been initialised via an ABC run as

described in Sect. 3.4. The plots clearly show that each of the chains are in agreement with regard to sampling from the stationary distribution

respectively. The system has been analysed by Komorowski et al. (2009) and Golightly et al. (2014) among others:



with stoichiometry matrix \mathbf{S} , and hazard function $h(\mathbf{X}_t, \theta)$

$$\mathbf{S} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \quad h(\mathbf{X}_t, \theta) = (\kappa_{R,t}, \gamma_R R_t, \kappa_P R_t, \gamma_P P_t)
 \tag{21}$$

where $\mathbf{X}_t = (R_t, P_t)$ and $\theta = (\gamma_R, \kappa_P, \gamma_P, b_0, b_1, b_2, b_3)$ where we note that, as in Komorowski et al. (2009), we take $\kappa_{R,t}$ to be the time dependent transcription rate. Specifically,

$$\kappa_{R,t} = b_0 \exp(-b_1(t - b_2)^2) + b_3
 \tag{22}$$

such that the transcription rate increases for $t < b_2$ and tends towards a baseline value, b_3 , for $t > b_2$. As above the goal is inference on the unknown parameter vector, θ . In keeping with inference in Komorowski et al. (2009) we create a data poor scenario, 100 observations of synthetic data simulated given initial conditions $X_0 = (10, 150)$ and parameter values (0.44, 0.52, 10, 15, 0.4, 7, 3) corrupted with mea-

surement error, $Y_t \sim \mathcal{N}(X_t, I\sigma^2)$, $\sigma = 10$, with observations on the mRNA discarded. The data is shown in Fig. 4(a).

We follow Komorowski et al. (2009) by assuming the same prior distributions, including informative priors for the degradation rates to ensure identifiability. Specifically

$$\begin{aligned}
 \gamma_R &\sim \Gamma(19.36, 44) & \gamma_P &\sim \Gamma(27.04, 52) \\
 \kappa_P &\sim \text{Exp}(0.01) & b_0 &\sim \text{Exp}(0.01) \\
 b_1 &\sim \text{Exp}(1.0) & b_2 &\sim \text{Exp}(0.1) \\
 b_3 &\sim \text{Exp}(0.01)
 \end{aligned}$$

where $\Gamma(a, b)$ is the gamma distribution with mean a/b and $\text{Exp}(a)$ is the Exponential distribution with mean $1/a$.

For simplicity we assume that both the initial state, $X_0 = (10, 150)$, and the measurement error standard deviation, $\sigma = 10$, are known.

4.3.1 Results for gene expression data

We follow the same procedure as with the two examples above. Using a sequential ABC run to obtain a sample of 1,000 parameter vectors distributed according to the approximate posterior. We then use eight random draws from the final ABC sample to initialise the parallel pMCMC chains with tuning parameters chosen as described in Sect. 3.4. The posterior densities, a sample of 4000 from each chain having

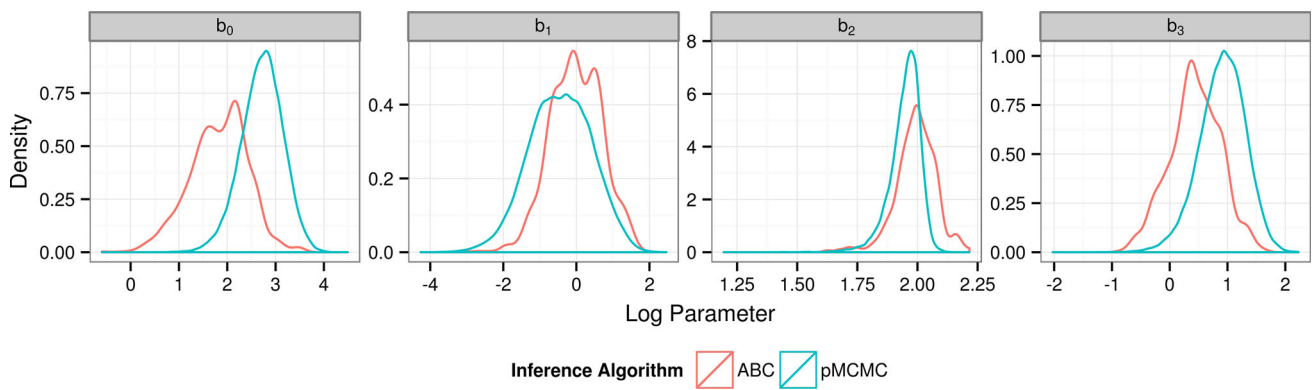


Fig. 5 A comparison between the final sample using the ABC SMC algorithm and the pMCMC for the gene regulation model for the four parameters in the time dependent hazard. The plot shows that their is

a distinct difference between the two posterior samples. Plots for the other three parameters show similar but are omitted here

been subjected to appropriate thinning, are shown in Fig. 4. It is clear that each of the chains is sampling from the same target giving us confidence in the resulting densities. The posteriors obtained are consistent with those in Golightly et al. (2014) and true parameter values are well identified. Figure 5 shows that the sample from the final iteration of the sequential ABC algorithm is markedly different from that in the pMCMC algorithm. There is a measurable improvement in using this type of scheme over using solely ABC in this way. We characterise the difference shown here as an improvement due to the fact that we know that pMCMC is asymptotically exact.

5 Discussion

We have proposed an approach to inference for Markov processes that is asymptotically exact and combines the relative strengths of ABC and pMCMC methodology to increase computational efficiency through use of parallel hardware. Through use of an approximation to the posterior distribution of interest, obtained via a sequential ABC algorithm which is easy to parallelise, we can set up a parallel implementation of pMCMC which has numerous desirable properties. By enabling the construction of independent parallel chains initialised close to the stationary distribution, this enables fast convergence and sampling from an exact posterior distribution that scales well with available computational resources. Throughout our analyses we have made use of parallel computation, however we believe that the proposed approach will also be of interest in situations where parallel hardware is not available, as it still addresses the pMCMC initialisation and tuning problem. Algorithmic tuning parameters required for pMCMC, such as the variance of Gaussian random walk proposals and numbers of particles for the particle filter can be chosen without the need for additional pilot runs, as a consequence of having a sample from an ABC posterior. In

addition, independent parallel chains allow verification of convergence and the computational saving in burn-in times extends to repeat MCMC analyses.

We have demonstrated this approach by applying it to three stochastic kinetic models. With the Lotka–Volterra predator prey system, a relatively simple model in which both species can be observed, we highlighted clear issues with practical implementation of a pseudo-marginal approach in a scenario in which prior information on reaction rate parameters is poor. This issue can be alleviated by first obtaining a sample from an approximation to the posterior, then using it to guide an exact pMCMC scheme. The approach discussed performed similarly well in the application to a set of real data for a model for aphid growth dynamics in which one of the species in the system can never be observed, where we again imposed weak prior conditions on the rate parameters governing the system and had access to repeat data. Finally we applied the scheme to a gene regulation model in which we had partially observed data and rate parameters were not all time homogeneous. The analyses of results show that we can verify that we are sampling from the same target distribution adding to our belief that we have converged to the true posterior of interest in each case.

Acknowledgments The authors gratefully acknowledge the referees for their detailed and constructive comments. JO was supported by an EPSRC studentship.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

Andrieu, C., Roberts, G.O.: The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Stat.* **37**(2), 697–725 (2009)

- Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods. *J. Royal Stat. Soc.* **72**(3), 269–342 (2010)
- Beaumont, M.A., Zhang, W., Balding, D.J.: Approximate Bayesian computation in population genetics. *Genetics* **162**(4), 2025–2035 (2002)
- Beaumont, M.A., Cornuet, J.M., Marin, J.M., Robert, C.P.: Adaptive approximate Bayesian computation. *Biometrika* **96**(4), 983–990 (2009)
- Blum, M.G., Nunes, M.A., Prangle, D., Sisson, S.A., et al.: A comparative review of dimension reduction methods in approximate bayesian computation. *Stat. Sci.* **28**(2), 189–208 (2013)
- Boys, R.J., Wilkinson, D.J., Kirkwood, T.B.: Bayesian inference for a discretely observed stochastic kinetic model. *Stat. Comput.* **18**(2), 125–135 (2008)
- Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. *J. Royal Stat. Soc.* **68**(3), 411–436 (2006)
- Doucet, A., de Freitas, N., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer, New York (2001)
- Doucet, A., Pitt, M., Kohn, R.: Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator (2012). arXiv:12101871.
- Drovandi, C.C., Pettitt, A.N.: Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. *Biometrics* **67**(1), 225–233 (2011)
- Fearnhead, P., Prangle, D.: Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *J. Royal Stat. Soc.* **74**(3), 419–474 (2012)
- Filippi, S., Barnes, C.P., Cornebise, J., Stumpf, M.P.: On optimality of kernels for approximate bayesian computation using sequential monte carlo. *Stat. Appl. Genet. Mol. Biol.* **12**(1), 87–107 (2013)
- Gillespie, C.S.: Moment-closure approximations for mass-action models. *IET Syst. Biol.* **3**(1), 52–58 (2009). doi:[10.1049/iet-syb:20070031](https://doi.org/10.1049/iet-syb:20070031)
- Gillespie, C.S., Golightly, A.: Bayesian inference for generalized stochastic population growth models with application to aphids. *J. Royal Stat. Soc.* **59**(2), 341–357 (2010)
- Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* **81**(25), 2340–2361 (1977)
- Gillespie, D.T.: The chemical Langevin equation. *J. Chem. Phys.* **113**(1), 297–306 (2000)
- Golightly, A., Wilkinson, D.J.: Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus* **1**(6), 807–820 (2011)
- Golightly, A., Henderson, D., Sherlock, C.: Delayed acceptance particle mcmc for exact inference in stochastic kinetic models. *Stat. Comput.* pp 1–17 (2014). doi:[10.1007/s11222-014-9469-x](https://doi.org/10.1007/s11222-014-9469-x)
- Kitano, H.: Computational systems biology. *Nature* **420**(6912), 206–210 (2002)
- Komorowski, M., Finkenstädt, B., Harper, C.V., Rand, D.A.: Bayesian inference of biochemical kinetic parameters using the linear noise approximation. *BMC Bioinform* **10**(1), 343 (2009)
- Lotka, A.J.: Elements of Physical Biology. Williams & Wilkins Baltimore, Baltimore (1925)
- Marjoram, P., Molitor, J., Plagnol, V., Tavaré, S.: Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* **100**(26):15,324–15,328 (2003)
- Matis, J.H., Kiffe, T.R., Matis, T.I., Stevenson, D.E.: Stochastic modeling of aphid population growth with nonlinear, power-law dynamics. *Math. Biosci.* **208**(2), 469–494 (2007)
- Matis, T.I., Parajulee, M.N., Matis, J.H., Shrestha, R.B.: A mechanistic model based analysis of cotton aphid population dynamics data. *Agric. For. Entomol.* **10**(4), 355–362 (2008)
- Pitt, M.K., RdS, Silva: On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *J. Econ.* **171**(2), 134–151 (2012)
- Pritchard, J.K., Seielstad, M.T., Perez-Lezaun, A., Feldman, M.W.: Population growth of human y chromosomes: a study of y chromosome microsatellites. *Mol. Biol. Evol.* **16**(12), 1791–1798 (1999)
- Roberts, G.O., Rosenthal, J.S.: Optimal scaling for various Metropolis-Hastings algorithms. *Stat. Sci.* **16**(4), 351–367 (2001)
- Roberts, G.O., Gelman, A., Gilks, W.R.: Weak convergence and optimal scaling of random walk metropolis algorithms. *Ann. Appl. Probab.* **7**(1), 110–120 (1997)
- Salis, H., Kaznessis, Y.: Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. *J. Chem. Phys.* **122**(054), 103 (2005)
- Sherlock, C., Thiery, A.H., Roberts, G.O., Rosenthal, J.S.: On the efficiency of pseudo-marginal random walk Metropolis algorithms. arXiv:13097209 (2013)
- Silk, D., Filippi, S., Stumpf, M.P.: Optimizing threshold-schedules for sequential approximate Bayesian computation: applications to molecular systems. *Stat Appl Genet Mol Biol* **12**(5), 603–618 (2013)
- Sisson, S., Fan, Y., Tanaka, M.M.: Sequential Monte Carlo without likelihoods. *Proc. Natl. Acad. Sci.* **104**(6), 1760–1765 (2007)
- Tavare, S., Balding, D.J., Griffiths, R., Donnelly, P.: Inferring coalescence times from dna sequence data. *Genetics* **145**(2), 505–518 (1997)
- Toni, T., Welch, D., Strelkowa, N., Ipsen, A., Stumpf, M.P.: Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *J. Royal Soc. Interface* **6**(31), 187–202 (2009)
- Volterra, V.: Fluctuations in the abundance of a species considered mathematically. *Nature* **118**, 558–560 (1926)
- Wilkinson, D.J.: Parallel Bayesian computation, Statistics Textbooks and Monographs, vol 184. MARCEL DEKKER AG (2006)
- Wilkinson DJ (2011) Stochastic modelling for systems biology, Chapman & Hall/CRC mathematical biology and medicine series, vol 44, 2nd edn. CRC Press