

# Efficient SMC<sup>2</sup> schemes for stochastic kinetic models

Andrew Golightly<sup>1</sup> · Theodore Kypraios<sup>2</sup>

Received: 10 April 2017 / Accepted: 2 November 2017  
© The Author(s) 2017. This article is an open access publication

**Abstract** Fitting stochastic kinetic models represented by Markov jump processes within the Bayesian paradigm is complicated by the intractability of the observed-data likelihood. There has therefore been considerable attention given to the design of pseudo-marginal Markov chain Monte Carlo algorithms for such models. However, these methods are typically computationally intensive, often require careful tuning and must be restarted from scratch upon receipt of new observations. Sequential Monte Carlo (SMC) methods on the other hand aim to efficiently reuse posterior samples at each time point. Despite their appeal, applying SMC schemes in scenarios with both dynamic states and static parameters is made difficult by the problem of particle degeneracy. A principled approach for overcoming this problem is to move each parameter particle through a Metropolis-Hastings kernel that leaves the target invariant. This rejuvenation step is key to a recently proposed SMC<sup>2</sup> algorithm, which can be seen as the pseudo-marginal analogue of an idealised scheme known as iterated batch importance sampling. Computing the parameter weights in SMC<sup>2</sup> requires running a particle filter over dynamic states to unbiasedly estimate the intractable observed-data likelihood up to the current time point. In this paper, we propose to use an auxiliary particle filter inside the SMC<sup>2</sup> scheme. Our method uses two recently proposed constructs for sampling conditioned jump processes, and we find that the resulting inference schemes typically require fewer state particles than when using a simple bootstrap filter. Using two applications, we compare the performance

of the proposed approach with various competing methods, including two global MCMC schemes.

**Keywords** Auxiliary particle filter (APF) · Bayesian inference · Markov jump process (MJP) · Sequential Monte Carlo (SMC) · Stochastic kinetic model (SKM)

## 1 Introduction

Markov jump processes (MJPs) are routinely used to describe the dynamics of discrete-valued processes evolving continuously in time. Application areas include (but are not limited to) systems biology (Golightly and Wilkinson 2005; Wilkinson 2012), predator–prey interaction (Ferm et al. 2008; Boys et al. 2008) and epidemiology (Lin and Ludkovski 2013; McKinley et al. 2014). Here, we focus on the MJP representation of a stochastic kinetic model (SKM), whereby transitions of species in a reaction network are described probabilistically via an instantaneous reaction rate or hazard, which depends on the current system state and a set of rate constants, with the latter typically the object of inference.

Owing to the intractability of the observed-data likelihood, Bayesian inference for SKMs is typically performed via Markov chain Monte Carlo (MCMC). Early attempts based on data augmentation were used by Gibson and Renshaw (1998) (see also O’Neill and Roberts (1999)) in the context of epidemiology, and by Boys et al. (2008) for more general reaction networks. Unfortunately, such methods can suffer from poor mixing due to dependence between the parameters and latent states to be imputed. Recently proposed pseudo-marginal MCMC schemes, e.g. particle MCMC (pMCMC) (Andrieu et al. 2010), offer a promising alternative and have been successfully applied in both the epidemiology (McKinley et al. 2014) and systems biology (Golightly and Wilkinson

✉ Andrew Golightly  
andrew.golightly@ncl.ac.uk

<sup>1</sup> School of Mathematics and Statistics, Newcastle University, Newcastle upon Tyne, UK

<sup>2</sup> School of Mathematical Sciences, University of Nottingham, Nottingham, UK

2015) literature. However, these ‘global’ inference schemes require careful selection and tuning of proposal mechanisms and must be restarted from scratch upon receipt of new observations or when assimilating information from multiple data sets. Moreover, the efficiency of such schemes depends heavily on the mechanism used to update the latent jump process.

We therefore consider sequential Monte Carlo (SMC) schemes which recycle posterior samples from one time point to the next through simple reweighting and resampling steps (see e.g. Doucet et al. (2001) for an introduction and Jacob (2015) for a recent review). The main drawback of SMC in scenarios with both dynamic states and static parameters is particle degeneracy: that is, when the number of distinct particles decreases over time. *Ad-hoc* approaches for overcoming this problem include the use of jittering each static parameter particle before propagation to the next time point (Gordon et al. 1993; Liu and West 2001). In special cases when the distribution of parameters given all latent states is tractable, this structure can be exploited to give a particle filter that uses conditional sufficient statistics to rejuvenate parameter samples (Storvik 2002; Fearnhead 2002). A related approach is the particle learning (PL) method of (Carvalho et al. 2010) which combines the use of conditional sufficient statistics with an auxiliary particle filter (Pitt and Shephard 1999). As discussed in Chopin et al. (2010) however, PL does not completely overcome the degeneracy issue. Chopin (2002) proposed a particle filter for static models (the so-called iterated batch importance sampling (IBIS) algorithm) that weights parameter particles by the observed-data likelihood contributions at each time point. Particle degeneracy is mitigated via a resample-move step (Gilks and Berzuini 2001), which ‘moves’ each parameter particle through a Metropolis-Hastings kernel that leaves the target invariant. This step can be executed subject to the fulfilment of some degeneracy criterion e.g. small effective sample size. Unfortunately, intractability of the observed-data likelihood precludes the use of IBIS for the class of models considered here.

The focus of this paper, therefore, is on the pseudo-marginal analogue of IBIS, which replaces the idealised particle weights with estimates obtained by running an SMC scheme over dynamic states for each parameter particle. The nested use of particle filters in this way results in an algorithm known as SMC<sup>2</sup> (Chopin et al. 2013). The resample-move step is accomplished by moving each parameter particle through a pMCMC kernel. The algorithm allows for choosing the number of state particles dynamically, by monitoring the acceptance rate of the resample-move step. Furthermore, the output of the algorithm can be used to estimate the model evidence at virtually no additional computational cost. This feature is particularly useful in the context of model selection, for example, when choosing between competing reaction networks based on a given data set.

The simplest implementation of SMC<sup>2</sup> uses a bootstrap filter over dynamic states in both the reweighting and move steps. However, this is likely to be particularly inefficient unless the noise in the measurement error process dominates the intrinsic stochasticity in the MJP. In this case, highly variable estimates of the observed-data likelihood will lead to small effective sample sizes, increasing the rate at which the resample-move step is triggered. Moreover, use of a bootstrap filter-driven pMCMC kernel is also likely to be highly inefficient, requiring many state particles to maintain a reasonable acceptance rate. In the special case of no measurement error, Drovandi and McCutchan (2016) use the alive particle filter of Del Moral et al. (2015) to drive an SMC<sup>2</sup> scheme.

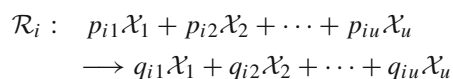
Our contribution is the development of an auxiliary particle filter for use inside the SMC<sup>2</sup> scheme. Our method uses two recently proposed constructs for sampling conditioned jump processes and can be applied in scenarios when only observations on a subset of system components are available. Moreover, observations may be subject to additive Gaussian error. We find that the proposed approach typically requires fewer state particles than when using a simple bootstrap filter. Using two applications and both real and synthetic data, we compare the performance of the proposed approach with various competing methods, including alive SMC<sup>2</sup>.

The remainder of this paper is organised as follows. In Sect. 2, a brief review of the Markov process representation of a reaction network is presented. Section 3 outlines the structure of the problem before presenting details of the auxiliary particle filter and its use inside SMC<sup>2</sup>. The methodology is used in a number of applications in Sect. 4 before conclusions are drawn in Sect. 5.

## 2 Stochastic kinetic models

We give here a brief introduction to stochastic kinetic models and refer the reader to Wilkinson (2012) for an in-depth treatment.

Consider a reaction network involving  $u$  species  $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_u$  and  $v$  reactions  $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_v$ , with each reaction denoted by  $\mathcal{R}_i$  and written as



where stoichiometric coefficients  $p_{ij}$  and  $q_{ij}$  are non-negative integers. When a type  $i$  reaction does occur, the system state changes discretely, via the  $i$ th row of the so-called net effect matrix  $A$ , a  $v \times u$  matrix with  $(i, j)$ th element given by  $q_{ij} - p_{ij}$ . In what follows, for notational convenience, we work with the stoichiometry matrix defined as  $S = A'$ . Let  $X_{j,t}$  denote the (discrete) number of species  $\mathcal{X}_j$  at time  $t$ , and let  $X_t$  be the  $u$ -vector

$X_t = (X_{1,t}, X_{2,t}, \dots, X_{u,t})'$ . The time evolution of  $X_t$  can be described by a vector of rates (or hazards) of the reactions together with the stoichiometry matrix which describes the effect of each reaction on the state. We therefore define a rate function  $h_i(X_t, c_i)$ , giving the overall hazard of a type  $i$  reaction occurring, and we let this depend explicitly on the reaction rate constant  $c_i$ , as well as the state of the system at time  $t$ . We model the system with a Markov jump process (MJP), so that for an infinitesimal time increment  $dt$ , the probability of a type  $i$  reaction occurring in the time interval  $(t, t + dt]$  is  $h_i(X_t, c_i)dt$ . Under the standard assumption of mass action kinetics, the hazard function for a particular reaction of type  $i$  takes the form of the rate constant multiplied by a product of binomial coefficients expressing the number of ways in which the reaction can occur, that is

$$h_i(X_t, c_i) = c_i \prod_{j=1}^u \binom{X_{j,t}}{p_{ij}}$$

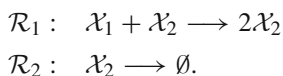
Values for  $c = (c_1, c_2, \dots, c_v)'$  and the initial system state  $X_0 = x_0$  complete specification of the Markov process. Although the transition probability associated with this process is rarely analytically tractable (except in some simple cases) generating exact realisations of the MJP is straightforward. This is due to the fact that if the current time and state of the system are  $t$  and  $X_t$  respectively, then the time to the next event will be exponential with rate parameter

$$h_0(X_t, c) = \sum_{i=1}^v h_i(X_t, c_i),$$

and the event will be a reaction of type  $\mathcal{R}_i$  with probability  $h_i(X_t, c_i)/h_0(X_t, c)$  independently of the inter-event time. This simulation method is typically referred to as *Gillespie's direct method* in the stochastic kinetics literature, after Gillespie (1977).

### 2.1 Example 1: a stochastic epidemic model

The Susceptible–Infected–Removed (SIR) epidemic model (see e.g. Andersson and Britton 2000) describes the evolution of two species (susceptibles  $\mathcal{X}_1$  and infectives  $\mathcal{X}_2$ ) via two reaction channels which correspond to an infection of a susceptible individual and a removal of an infective individual:



The stoichiometry matrix is given by

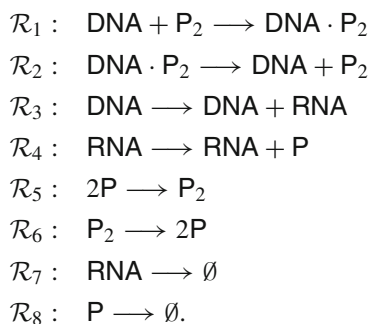
$$S = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}$$

and the associated hazard function is

$$h(X_t, c) = (c_1 X_{1,t} X_{2,t}, c_2 X_{2,t})'$$

### 2.2 Example 2: prokaryotic autoregulation

A commonly used mechanism for autoregulation in prokaryotes which has been well-studied and modelled is a negative feedback mechanism whereby dimers of a protein repress its own transcription (e.g. Arkin et al. 1998). A simplified model for such a prokaryotic autoregulation, based on this mechanism of dimers of a protein coded for by a gene repressing its own transcription, can be found in Golightly and Wilkinson (2005) (see also Golightly and Wilkinson (2011)). The full set of reactions in this simplified model are



Note that this model contains a conservation law, so that the total number  $k$  of  $\text{DNA} \cdot P_2$  and  $\text{DNA}$  is fixed for all time. Denoting the number of molecules of  $\text{RNA}$ ,  $P$ ,  $P_2$  and  $\text{DNA}$  as  $X_1$ ,  $X_2$ ,  $X_3$  and  $X_4$  respectively, gives the stoichiometry matrix

$$S = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 2 & 0 & -1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and associated hazard function

$$h(X, c) = (c_1 X_4 X_3, c_2(k - X_4), c_3 X_4, c_4 X_1, c_5 X_2(X_2 - 1)/2, c_6 X_3, c_7 X_1, c_8 X_2)'$$

where we have dropped  $t$  to ease the notation.

## 3 Sequential Bayesian inference

### 3.1 Setup

Suppose that the Markov jump process is not observed directly, but observations (on a regular grid)  $y_t, t = 1, 2, \dots$  are available and assumed conditionally independent (given

the latent jump process) with conditional probability distribution obtained via the observation equation

$$Y_t = P'X_t + \varepsilon_t, \quad \varepsilon_t \sim N(0, \Sigma), \quad t = 1, 2, \dots \quad (1)$$

Here,  $Y_t$  is taken to be a length- $p$  vector,  $P$  is a constant matrix of dimension  $u \times p$  and  $\varepsilon_t$  is a length- $p$  Gaussian random vector. The density  $p(y_t|x_t)$  linking the observed and latent processes satisfies

$$p(y_t|y_{1:t-1}, x_{[1,t]}, c) = p(y_t|x_t, c)$$

where  $x_{[1,t-1]}$  denotes the MJP over an interval  $[1, t-1]$ .

We assume that primary interest lies in the recursive exploration of the marginal posteriors  $p(c|y_{1:t})$ ,  $t = 1, \dots, T$ . Upon ascribing a prior density  $p(c)$  to the parameters, Bayes theorem gives

$$p(c|y_{1:t}) \propto p(c)p(y_{1:t}|c) \quad (2)$$

$$\propto p(c|y_{1:t-1})p(y_t|y_{1:t-1}, c) \quad (3)$$

which immediately suggests a sequential importance sampling scheme that repeatedly reweights a set of  $N_c$  parameter samples (known as ‘particles’ in this context) by the observed-data (or ‘marginal’) likelihood contributions  $p(y_t|y_{1:t-1}, c)$ . This approach is used in the iterated batch importance sampling (IBIS) algorithm of Chopin (2002), together with MCMC steps for rejuvenating parameter samples in order to circumvent particle degeneracy. Although each observed-data likelihood contribution is typically intractable, progress can be made by substituting a non-negative estimate of  $p(y_t|y_{1:t-1}, c)$ . In order for the resulting algorithm to target the correct posterior, these estimates should be constructed so that the observed-data likelihood up to the current time point,  $p(y_{1:t}|c)$ , can be unbiasedly estimated. This task can be achieved by running a particle filter with  $N_x$  particles targeting  $p(x_t|y_{1:t}, c)$  for each  $c$ -particle. Particle MCMC steps are then occasionally used to rejuvenate the sample. This approach was proposed and theoretically justified by Chopin et al. (2013) who term the resulting algorithm SMC<sup>2</sup> due to the use of nested filters. The simplest implementation of the algorithm runs a bootstrap particle filter (e.g. Gordon et al. 1993) for each  $c$ -particle, which only requires the ability to forward-simulate the MJP and evaluate  $p(y_t|x_t, c)$ . Despite the appeal of this simple approach, the resulting estimates of the observed-data likelihood contributions can have high variance, unless the observations are not particularly informative, limiting the efficiency of the SMC<sup>2</sup> scheme. This is due to the collapse of the bootstrap particle filter, which results from very few state trajectories having reasonable weight. The problem is exacerbated in the case of no measurement error, where only state trajectories that ‘hit’ observations are assigned

a non-zero weight. Drovandi and McCutchan (2016) use the alive particle filter of Del Moral et al. (2015) (see also Appendix A.1) to avoid this problem. Unfortunately, this approach can be extremely computationally expensive, since it repeatedly generates simulations of the jump process until a predetermined number of hits are obtained. In what follows, therefore, we use an auxiliary particle filter (for which the bootstrap filter can be viewed as a special case) to efficiently estimate each  $p(y_t|y_{1:t-1}, c)$ . We describe the auxiliary particle filter in the next section before describing its use inside an SMC<sup>2</sup> scheme.

### 3.2 Auxiliary particle filter

The aim of the particle filter is to recursively approximate the sequence of filtering densities  $p(x_t|y_{1:t}, c)$ . To this end, suppose that at time  $t-1$ , a weighted sample  $\{x_{t-1}^i, w_{t-1,c}^i\}_{i=1}^{N_x}$  is available, and is approximately distributed according to  $p(x_{t-1}|y_{1:t-1}, c)$ . Note that although the predictive  $p(x_{(t-1,t)}|y_{1:t-1}, c)$  is typically intractable, the weighted sample from the previous time point can be used to give the approximation  $\hat{p}(x_{(t-1,t)}|y_{1:t-1}, c) \propto \sum_{i=1}^{N_x} p(x_{(t-1,t)}|x_{t-1}^i, c)w_{t-1,c}^i$ . Hence, upon receipt of a new datum  $y_t$ , the particle filter constructs the approximate posterior

$$\hat{p}(x_{(t-1,t)}|y_{1:t}, c) \propto p(y_t|x_t, c) \sum_{i=1}^{N_x} p(x_{(t-1,t)}|x_{t-1}^i, c) w_{t-1,c}^i \quad (4)$$

from which draws can be generated using (for example) importance resampling. A simple strategy is to use  $\hat{p}(x_{(t-1,t)}|y_{1:t-1}, c)$  as a proposal mechanism, which is straightforward to sample from by picking a particle  $x_{t-1}^i$  with probability  $w_{t-1,c}^i$  and simulating according to  $p(x_{(t-1,t)}|x_{t-1}^i, c)$  using Gillespie’s direct method (see Sect. 2). The state  $x_t^i$  can be stored along with the new (unnormalised) weight  $\tilde{w}_{t,c}^i = p(y_t|x_t^i, c)$ . Resampling (with replacement) amongst the particles using the weights as probabilities gives a sample approximately distributed according to (4). Repeating this procedure for each time point gives the bootstrap particle filter of Gordon et al. (1993).

The auxiliary particle filter (APF) of Pitt and Shephard (1999) (see also Pitt et al. 2012) can be seen as a generalisation of the bootstrap filter. The APF is constructed by noting that

$$\begin{aligned} p(y_t|x_t, c)p(x_{(t-1,t)}|x_{t-1}, c) \\ = p(y_t|x_{t-1}, c)p(x_{(t-1,t)}|x_{t-1}, y_t, c) \end{aligned}$$

which immediately suggests an importance resampling strategy that initially preweights each  $x_{t-1}^i$  particle by

$\tilde{w}_{t-1|t,c} = p(y_t|x_{t-1}^i, c)w_{t-1,c}^i$  and propagates according to  $p(x_{(t-1,t]}|x_{t-1}^i, y_t, c)$ . The new (unnormalised) weight is  $\tilde{w}_{t,c}^i = 1$ , giving the fully adapted form of the APF (Pitt and Shephard 2001). In practice,  $p(y_t|x_{t-1}, c)$  and  $p(x_{(t-1,t]}|x_{t-1}, y_t, c)$  are intractable and approximations  $g(y_t|x_{t-1}, c)$  and  $g(x_{(t-1,t]}|x_{t-1}, y_t, c)$  must be sought, giving the APF described in Algorithm 1. Note that taking  $g(y_t|x_{t-1}, c) = 1$  and  $g(x_{(t-1,t]}|x_{t-1}, y_t, c) = p(x_{(t-1,t]}|x_{t-1}, c)$  admits the bootstrap particle filter as a special case.

Following Pitt et al. (2012), we use the output of the APF to estimate  $p(y_t|y_{1:t-1}, c)$  with the quantity

$$\hat{p}(y_t|y_{1:t-1}, c) = \left( \sum_{i=1}^{N_x} \frac{\tilde{w}_{t,c}^i}{N_x} \right) \left( \sum_{i=1}^{N_x} \tilde{w}_{t-1|t,c} \right).$$

Crucially, Pitt et al. (2012) (see also Del Moral 2004) show that

$$\hat{p}(y_{1:T}|c) = \hat{p}(y_1|c) \prod_{t=2}^T \hat{p}(y_t|y_{1:t-1}, c)$$

is an unbiased estimator of  $p(y_{1:T}|c)$ . Justification of the use of  $\hat{p}(y_t|y_{1:t-1}, c)$ , as given above, in an SMC<sup>2</sup> scheme then follows directly from Chopin et al. (2013).

### 3.2.1 Propagation: method 1

It remains that we can find suitable densities  $g(y_t|x_{t-1}, c)$  and  $g(x_{(t-1,t]}|x_{t-1}, y_t, c)$ . Focusing first on the latter, we use an approximation to the conditioned jump process proposed by Golightly and Wilkinson (2015). The method works by approximating the expected number of reaction events over an interval of interest conditional on the next observation. The resulting conditioned hazard is used in place of the unconditioned hazard in Gillespie’s direct method.

Consider an interval  $[t - 1, t]$  and suppose that we have simulated as far as time  $s \in [t - 1, t]$ . Let  $\Delta R_s$  denote the number of reaction events over the time  $t - s = \Delta s$ . We approximate  $\Delta R_s$  by assuming a constant reaction hazard over  $\Delta s$ . A Gaussian approximation to the corresponding Poisson distribution then gives

$$\Delta R_s \sim N(h(x_s, c)\Delta s, H(x_s, c)\Delta s)$$

where  $H(x_s, c) = \text{diag}\{h(x_s, c)\}$ . Under the Gaussian observation regime (1) we have that

$$Y_t|X_s = x_s \sim N(P'(x_s + S h(x_s, c)\Delta s), P'S H(x_s, c)S'P\Delta s + \Sigma). \tag{5}$$

Hence, the joint distribution of  $\Delta R_s$  and  $Y_t$  (conditional on  $x_s$ ) can then be obtained approximately as

### Algorithm 1 Auxiliary particle filter

1. Initialisation ( $t = 1$ ). For  $i = 1, 2, \dots, N_x$ :
  - (a) Sample  $x_1^i \sim p(\cdot)$ .
  - (b) Compute the weights  $\tilde{w}_{1,c}^i = p(y_1|x_1^i, c)$ ,  $w_{1,c}^i = \frac{\tilde{w}_{1,c}^i}{\sum_{j=1}^{N_x} \tilde{w}_{1,c}^j}$ .
  - (c) Compute the current estimate of observed-data likelihood  $\hat{p}(y_1|c) = \sum_{i=1}^{N_x} \tilde{w}_{1,c}^i/N_x$ .
2. For times  $t = 2, 3, \dots, T$  and  $i = 1, 2, \dots, N_x$ :
  - (a) Compute the preweights  $\tilde{w}_{t-1|t,c}^i = g(y_t|x_{t-1}^i, c)w_{t-1,c}^i$ ,  $w_{t-1|t,c}^i = \frac{\tilde{w}_{t-1|t,c}^i}{\sum_{j=1}^{N_x} \tilde{w}_{t-1|t,c}^j}$ .
  - (b) Sample the index  $a_{t-1}^i \sim \mathcal{M}(w_{t-1|t,c}^{1:N_x})$  of the ancestor of particle  $i$ , where  $\mathcal{M}(p^{1:n})$  denotes the multinomial distribution that assigns a probability  $p^i$  to outcome  $i \in \{1, \dots, n\}$ .
  - (c) Propagate. Sample  $x_{(t-1,t]}^i \sim g(\cdot|x_{t-1}^{a_{t-1}^i}, y_t, c)$ .
  - (d) Compute the weights

$$\tilde{w}_{t,c}^i = \frac{p(y_t|x_t^i, c)p(x_{(t-1,t]}^i|x_{t-1}^{a_{t-1}^i}, c)}{g(y_t|x_{t-1}^{a_{t-1}^i}, c)g(x_{(t-1,t]}^i|x_{t-1}^{a_{t-1}^i}, y_t, c)}$$

$$w_{t,c}^i = \frac{\tilde{w}_{t,c}^i}{\sum_{j=1}^{N_x} \tilde{w}_{t,c}^j}$$

- (e) Compute the current estimate of observed-data likelihood  $\hat{p}(y_{1:t}|c) = \hat{p}(y_{1:t-1}|c)\hat{p}(y_t|y_{1:t-1}, c)$  where

$$\hat{p}(y_t|y_{1:t-1}, c) = \left( \sum_{i=1}^{N_x} \frac{\tilde{w}_{t,c}^i}{N_x} \right) \left( \sum_{i=1}^{N_x} \tilde{w}_{t-1|t,c} \right).$$

$$\begin{pmatrix} \Delta R_s \\ Y_t \end{pmatrix} \sim N \left\{ \begin{pmatrix} h(x_s, c)\Delta s \\ P'(x_s + S h(x_s, c)\Delta s) \end{pmatrix}, \begin{pmatrix} H(x_s, c)\Delta s & H(x_s, c)S'P\Delta s \\ P'S H(x_s, c)\Delta s & P'S H(x_s, c)S'P\Delta s + \Sigma \end{pmatrix} \right\}.$$

Taking the expectation of  $\Delta R_s|Y_t = y_t$  and dividing the resulting expression by  $\Delta s$  gives an approximate conditioned hazard as

$$h^*(x_s, c|y_t) = h(x_s, c) + H(x_s, c)S'P(P'S H(x_s, c)S'P\Delta s + \Sigma)^{-1} \times (y_t - P'[x_s + S h(x_s, c)\Delta s]). \tag{6}$$

Although the conditioned hazard in (6) depends on the current time  $s$  in a nonlinear way, a simple implementation ignores this time dependence, giving exponential waiting times between reaction events. Hence, the construct can be used to generate realisations from an approximation to the true (but intractable) conditioned jump process by applying Gillespie’s direct method with  $h(x_s, c)$  replaced by  $h^*(x_s, c|y_t)$ .

To calculate the weights used in step 2(d) of Algorithm 1, we note that  $p(x_{(t-1,t]}|x_{t-1}, c)$  can be written explicitly by

considering the generation of all reaction times and types over  $(t - 1, t]$ . To this end, we let  $r_j$  denote the number of reaction events of type  $\mathcal{R}_j$ ,  $j = 1, \dots, v$ , and define  $n_r = \sum_{j=1}^v r_j$  as the total number of reaction events over the interval, which is obtained deterministically from the trajectory  $x_{(t-1,t]}$ . Reaction times (assumed to be in increasing order) and types are denoted by  $(\tau_i, v_i)$ ,  $i = 1, \dots, n_r$ ,  $v_i \in \{1, \dots, v\}$  and we take  $\tau_0 = t - 1$  and  $\tau_{n_r+1} = t$ . The so-called complete-data likelihood (Wilkinson 2012) over  $(t - 1, t]$  is then given by

$$p(x_{(t-1,t]}|x_{t-1}, c) = \left\{ \prod_{i=1}^{n_r} h_{v_i}(x_{\tau_{i-1}}, c_{v_i}) \right\} \times \exp \left\{ - \sum_{i=1}^{n_r} h_0(x_{\tau_i}, c) (\tau_{i+1} - \tau_i) \right\}$$

An expression for  $g(x_{(t-1,t]}|x_{t-1}, y_t, c)$  is obtained similarly. Hence, the weights we require take the form

$$\begin{aligned} \tilde{w}_{t,c} &= \frac{p(y_t|x_t, c)p(x_{(t-1,t]}|x_{t-1}, c)}{g(y_t|x_{t-1}, c)g(x_{(t-1,t]}|x_{t-1}, y_t, c)} \\ &= \frac{p(y_t|x_t, c)}{g(y_t|x_{t-1}, c)} \left\{ \prod_{i=1}^{n_r} \frac{h_{v_i}(x_{\tau_{i-1}}, c_{v_i})}{h_{v_i}^*(x_{\tau_{i-1}}, c_{v_i}|y_t)} \right\} \\ &\quad \times \exp \left\{ - \sum_{i=1}^{n_r} [h_0(x_{\tau_i}, c) - h_0^*(x_{\tau_i}, c|y_t)] [\tau_{i+1} - \tau_i] \right\}. \end{aligned} \quad (7)$$

### 3.2.2 Propagation: method 2

Fearnhead (2008) derives a conditioned hazard in the case of complete and noise-free observation of the MJP. Extending the method to the observation scenario given by (1) is straightforward. Consider again an interval  $[t - 1, t]$  and suppose that we have simulated as far as time  $s \in [t - 1, t]$ . For reaction  $\mathcal{R}_i$  let  $x' = x_s + S^{(i)}$ , where  $S^{(i)}$  denotes the  $i$ th column of the stoichiometry matrix so that  $x'$  is the state of the MJP after a single occurrence of  $\mathcal{R}_i$ . The conditioned hazard of  $\mathcal{R}_i$  satisfies

$$\begin{aligned} h_i(x_s, c|y_t) &= \lim_{\Delta s \rightarrow 0} \frac{\Pr(X_{s+\Delta s} = x'|X_s = x_s, y_t, c)}{\Delta s} \\ &= h_i(x_s, c_i) \lim_{\Delta s \rightarrow 0} \frac{p(y_t|X_{s+\Delta s} = x', c)}{p(y_t|X_s = x_s, c)} \\ &= h_i(x_s, c_i) \frac{p(y_t|X_s = x', c)}{p(y_t|X_s = x_s, c)}. \end{aligned}$$

Of course, in practice, the transition density  $p(y_t|x_s, c)$  is intractable and we therefore use the approximation in (5) to obtain an approximate conditioned hazard  $h_i^\dagger(x_s, c|y_t)$  and combined hazard  $h_0^\dagger(x_s, c|y_t)$ . Note that to calculate this approximate conditioned hazard, the density associated with

the approximation in (5) must be calculated  $v + 1$  times (once using  $x_s$  and for each  $x'$  obtained after the  $v$  possible transitions of the process). Although  $h_0^\dagger(x_s, c|y_t)$  is time dependent, the simple simulation approach described in Sect. 3.2.1 that ignores this time dependence can be easily implemented. The form of the weight required in step 2(d) of Algorithm 1 is given by Eq. 7 with  $h^*$  replaced by  $h^\dagger$ .

### 3.2.3 Prewriteight

Finally, note that the derivations of the conditioned hazards described above suggest a form for the preweight  $g(y_t|x_{t-1}, c)$ . Using the approximation in (5) with  $s = t - 1$  and assuming an inter-observation time of  $\Delta$  gives

$$g(y_t|x_{t-1}, c) = N(y_t; P'(x_{t-1} + S h(x_{t-1}, c)\Delta), P' S H(x_{t-1}, c) S' P \Delta + \Sigma) \quad (8)$$

where  $N(\cdot; m, V)$  denotes the multivariate Gaussian density with mean vector  $m$  and variance matrix  $V$ . In some scenarios, the density in (8) may have lighter tails than  $p(y_t|x_{t-1}, c)$ . In this case, some particles that are consistent with the next observation are likely to be pruned out. Although the problem can be alleviated by raising the density in (8) to a power (say  $1/\delta$  where  $\delta > 1$ ), this introduces an additional tuning parameter. We find that simply taking  $g(y_t|x_{t-1}, c) = 1$  is computationally convenient and works well in practice.

### 3.3 SMC<sup>2</sup> scheme

In this section, we provide a brief exposition of the SMC<sup>2</sup> scheme. The reader is referred to Chopin et al. (2013) for further details including a formal justification (see also Fulop and Li (2013) for a related algorithm and Jacob (2015) for a recent discussion).

Recall the target posterior at time  $t$ ,  $p(c|y_{1:t})$  given by (3). Suppose that a weighted sample  $\{c^k, \omega^k\}_{k=1}^{N_c}$  from  $p(c|y_{1:t})$  is available. The SMC<sup>2</sup> algorithm reweights each  $c$ -particle according to a non-negative estimate of  $p(y_t|y_{1:t-1}, c^k)$ , obtained from the output of a particle filter. We propose to use the auxiliary particle filter of Sect. 3.2. In order to use the APF in this way, we require storage of the state particles and associated weights at each time point  $t$  and for each parameter particle  $c^k$ . We denote the APF output at iteration  $t$  by  $\{x_{t,c^k}^{1:N_x}, w_{t,c^k}^{1:N_x}\}$ . To circumvent particle degeneracy, the SMC<sup>2</sup> scheme uses a resample-move step (see e.g. Gilks and Berzuini 2001) that firstly resamples parameter particles (and the associated states, weights and observed-data likelihoods  $p(y_{1:t}|c^k)$ ) and then moves each parameter sample through a particle Metropolis-Hastings kernel which leaves the target posterior invariant (Andrieu et al. 2010). The resample-move

step is only used if some degeneracy criterion is fulfilled. Typically, at each time  $t$ , the effective sample size (ESS) is computed as

$$ESS = 1 / \sum_{k=1}^{N_c} (\omega^k)^2$$

and the resample-move step is triggered if  $ESS < \gamma N_c$  for  $\gamma \in (0, 1)$  and a standard choice is  $\gamma = 0.5$ . A key feature of the SMC<sup>2</sup> scheme is that the current set of  $c$ -particles can be used in the design of the proposal density  $q(c^*|c)$ . For the applications in Sect. 4, we use an independent proposal so that  $q(c^*|c) = q(c^*)$ . As the rate constants must be strictly positive, we take

$$q(c^*) = \log N \left( c^*; \widehat{E}(\log(c)|y_{1:t}), \widehat{Var}(\log(c)|y_{1:t}) \right)$$

where  $\log N(\cdot; m, V)$  denotes the density associated with the exponential of a  $N(m, V)$  random variable.

The SMC<sup>2</sup> scheme with fixed  $N_x$  is given by Algorithm 2. It remains that the number of state particles is suitably chosen. Andrieu et al. (2010) show that  $N_x = O(t)$  to obtain a reasonable acceptance rate in the particle Metropolis-Hastings step. Therefore, Chopin et al. (2013) suggest an automatic method that allows  $N_x$  to increase over time. Essentially, the acceptance rate of the move step is monitored and if this rate falls below a given threshold,  $N_x$  is increased (e.g. by multiplying by 2). Suppose that at time  $t$  and for each  $c^k$ , we have  $\{x_{t,c^k}^{1:N_x}, w_{t,c^k}^{1:N_x}\}$  and observed-data likelihood  $\hat{p}_{N_x}(y_{1:t}|c^k)$ , where we have explicitly written the observed-data likelihood to depend on  $N_x$ . Let  $\tilde{N}_x$  denote the updated number of state particles. A generalised importance sampling strategy is used to swap the  $x$ -particles, their associated weights and the estimates of observed-data likelihood with new values obtained by running the APF with  $\tilde{N}_x$  state particles, for each  $c^k$ . Chopin et al. (2013) show that the weights associated with each parameter particle  $c^k$  should be multiplied by  $\hat{p}_{N_x}(y_{1:t}|c^k) / \hat{p}_{\tilde{N}_x}(y_{1:t}|c^k)$ . Fortunately, the frequency at which the potentially expensive resample-move step is executed reduces over time and the computational cost of the algorithm is  $O(N_c t^2)$  (rather than  $O(N_c t^3)$  if the resample-move step was triggered at every time point).

Finally, consider the evidence

$$p(y_{1:T}) = \prod_{t=1}^T p(y_t|y_{1:t-1}),$$

where we adopt the convention that  $p(y_1) = p(y_1|y_{1:0})$ . It is straightforward to estimate  $p(y_{1:T})$  using the output of the SMC<sup>2</sup> scheme, at virtually no additional computational cost. Each factor  $L_t = p(y_t|y_{1:t-1})$  in the product above is

**Algorithm 2** SMC<sup>2</sup> scheme

1. Initialisation. For  $k = 1, \dots, N_c$  sample  $c^k \sim p(\cdot)$  and set  $\tilde{\omega}^k = 1$ . For  $t = 1, \dots, T$ :
2. Sequential importance sampling. For  $k = 1, \dots, N_c$ :
  - (a) Perform iteration  $t$  of the auxiliary particle filter to obtain  $\{x_{t,c^k}^{1:N_x}, w_{t,c^k}^{1:N_x}\}$  and  $\hat{p}(y_t|y_{1:t-1}, c^k)$ . Note that  $\hat{p}(y_1|c^k) = \hat{p}(y_1|y_{1:0}, c^k)$ .
  - (b) Update and normalise the importance weights via

$$\tilde{\omega}^k := \tilde{\omega}^k \hat{p}(y_t|y_{1:t-1}, c^k), \quad \omega^k = \frac{\tilde{\omega}^k}{\sum_{j=1}^{N_c} \tilde{\omega}^j}$$

- (c) Update observed-data likelihood estimate via

$$\hat{p}(y_{1:t}|c^k) = \hat{p}(y_{1:t-1}|c^k) \hat{p}(y_t|y_{1:t-1}, c^k).$$

3. If  $ESS < \gamma N_c$  resample and move. For  $k = 1, \dots, N_c$ :

- (a) Sample indices  $a_k \sim \mathcal{M}(\omega^{1:N_c})$  and set  $\{c^k, \tilde{\omega}^k\} := \{c^{a_k}, 1\}$ ,  $\{x_{t,c^k}^{1:N_x}, w_{t,c^k}^{1:N_x}\} := \{x_{t,c^{a_k}}^{1:N_x}, w_{t,c^{a_k}}^{1:N_x}\}$  and  $\hat{p}(y_{1:t}|c^k) := \hat{p}(y_{1:t}|c^{a_k})$ .
- (b) Propose  $c^* \sim q(\cdot|c^k)$ . Perform iterations  $1, \dots, t$  of the auxiliary particle filter to obtain  $\hat{p}(y_{1:t}|c^*)$ . With probability

$$\min \left\{ 1, \frac{p(c^*) \hat{p}(y_{1:t}|c^*)}{p(c^k) \hat{p}(y_{1:t}|c^k)} \times \frac{q(c^k|c^*)}{q(c^*|c^k)} \right\}$$

$$\text{put } c^k := c^*, \{x_{t,c^k}^{1:N_x}, w_{t,c^k}^{1:N_x}\} := \{x_{t,c^*}^{1:N_x}, w_{t,c^*}^{1:N_x}\} \text{ and } \hat{p}(y_{1:t}|c^k) := \hat{p}(y_{1:t}|c^*).$$

estimated by

$$\hat{L}_t = \sum_{k=1}^{N_c} \omega^k \hat{p}(y_t|y_{1:t-1}, c^k). \tag{9}$$

**4 Applications**

To illustrate the methodology described in the previous sections, we consider two applications of increasing complexity. In Sect. 4.1, a Susceptible-Infected-Removed (SIR) epidemic model is fitted using real data; namely, the Abakaliki smallpox data set given in Bailey (1975). We compare the performance of SMC<sup>2</sup> schemes based on auxiliary, bootstrap and alive particle filters. Using synthetic data, we compare the best performing SMC<sup>2</sup> scheme with its particle MCMC counterpart and, additionally, a data augmentation scheme. In Sect. 4.2, we apply SMC<sup>2</sup> to infer the parameters governing a simple prokaryotic autoregulatory network using synthetic data. All algorithms are coded in C and were run on a desktop computer with an Intel Core i7-4770 processor and a 3.40GHz clock speed. The code is available at <http://www.mas.ncl.ac.uk/~nag48/smc2.zip>.

#### 4.1 Abakaliki smallpox data

We first consider the well-studied temporal data set obtained from a smallpox outbreak that took place in the small Nigerian village Abakaliki in 1967. Bailey (1975, p. 125) provides a complete set of 29 inter-removal times, measured in days. Table 1 shows the data here as the days on which the removal of individuals actually took place, with the first day set to be time 0. The outbreak resulted in 32 cases, 30 out of which corresponded to individuals who were members of a religious organisation whose 120 members refused to be vaccinated.

Numerous authors such as O'Neill and Roberts (1999), Fearnhead and Meligkotsidou (2004) and Boys and Giles (2007) amongst others have considered these data by focussing solely on the 30 cases amongst the population of 120, despite the fact that the original dataset (provided in a WHO report) contains far more information than the inter-removal times, such as the physical locations of the cases and the members of each household. A fully Bayesian analysis of this full dataset can be found in Stockdale et al. (2017), but here our purpose is to illustrate our methodology and therefore, we only consider the partial data set assuming that there have been 30 cases in a closed population of size 120.

We assume an SIR model (see Sect. 2.1) for the data with observations being equivalent to daily measurements of  $X_1 + X_2$  (as there is a fixed population size). In addition, and for simplicity, we assume that a single individual remained infective just after the first removal occurred. We analyse the data under the assumption of no measurement error, that is,  $P' = (1, 1)$  and  $\Sigma = 0$  in the observation equation (1).

We followed Fearnhead and Meligkotsidou (2004) by taking independent Gamma priors so that  $c_1 \sim Ga(10, 10^4)$  and  $c_2 \sim Ga(10, 10^2)$  a priori, where  $Ga(a, b)$  denotes a Gamma distribution with shape  $a$  and rate  $b$ . We applied

three different SMC<sup>2</sup> schemes based on the bootstrap, alive and auxiliary (with propagation methods 1 and 2) particle filters. In each case we took  $N_c = 5000$ , an ESS-threshold of  $\gamma = 50\%$  and an initial number of state particles of  $N_x = 10$ , except when using the bootstrap filter which required  $N_x = 100$  initially, to give output comparable to the other methods, in terms of accuracy (—see further discussion below). The value of  $N_x$  was doubled if the acceptance rate calculated in the resample-move step fell below 20%.

Table 2 and Figs. 1, 2 and 3 summarise the output of each SMC<sup>2</sup> scheme. We compare the accuracy of each scheme by reporting bias and root-mean-square error (RMSE) of the estimators of the marginal posterior means and standard deviations of  $\log(c_1)$  and  $\log(c_2)$ . These quantities are reported in Table 2 and were obtained by performing 100 independent runs of each scheme and comparing the aforementioned posterior estimators to reference values, obtained from a long run ( $3 \times 10^6$  iterations) of particle MCMC (pMCMC). For the pMCMC run, we used the auxiliary particle filter-driven scheme of Golightly and Wilkinson (2015) which uses Algorithm 1 and propagation method 1 at each MCMC iteration to compute  $\hat{p}(y_{1:T} | c^*)$  for a proposed value  $c^*$ . A comparison of SMC<sup>2</sup> and pMCMC is given in Sect. 4.1.1.

Inspection of Table 2 shows that all schemes give generally comparable output in terms of bias and RMSE, although we found that the bootstrap implementation was particularly sensitive to the initial choice of  $N_x$ , with relatively low values leading to noticeable biases in the marginal posterior mean estimators. Using 100 initial state particles seemed to alleviate this problem. We therefore use CPU cost as a proxy for overall efficiency. Interestingly, the alive SMC<sup>2</sup> scheme performs poorly in terms of CPU cost, despite requiring the smallest number of state particles. As can be seen from Fig. 1 (left panel), alive SMC<sup>2</sup> maintains a high effective sample

**Table 1** Abakaliki smallpox data

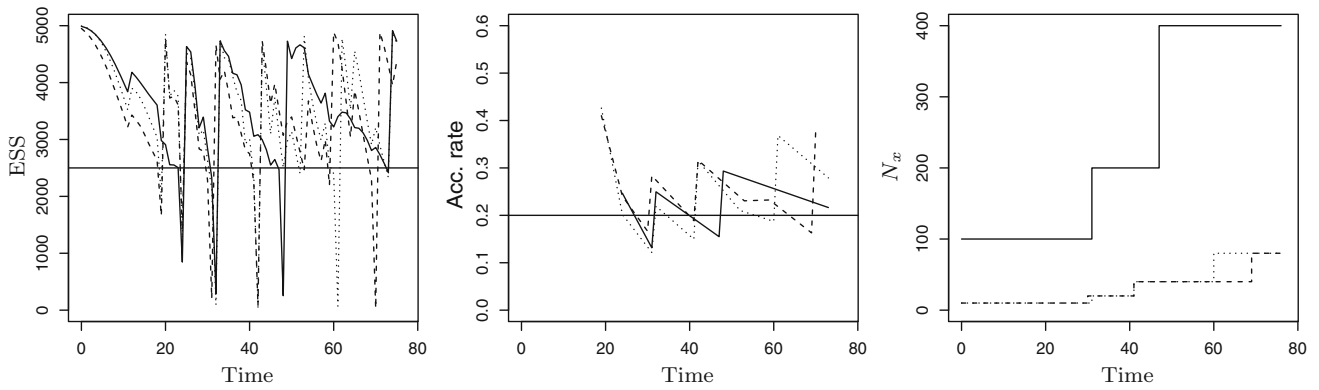
Day	1	14	21	23	26	27	31	36	39	41	43	48
No. of removals	1	1	1	1	3	1	1	1	1	2	2	1
Day	51	52	56	57	58	59	61	62	67	72	77	
No. of removals	1	1	2	1	1	1	2	1	2	1	1	

**Table 2** SIR epidemic model (Abakaliki data).  $N_x$  at time  $T$ , CPU time (in seconds), bias (and RMSE in parentheses) of estimators of the posterior expectations  $E(\log(c_1)|y_{1:T})$ ,  $E(\log(c_2)|y_{1:T})$  and standard deviations  $SD(\log(c_1)|y_{1:T})$ ,  $SD(\log(c_2)|y_{1:T})$

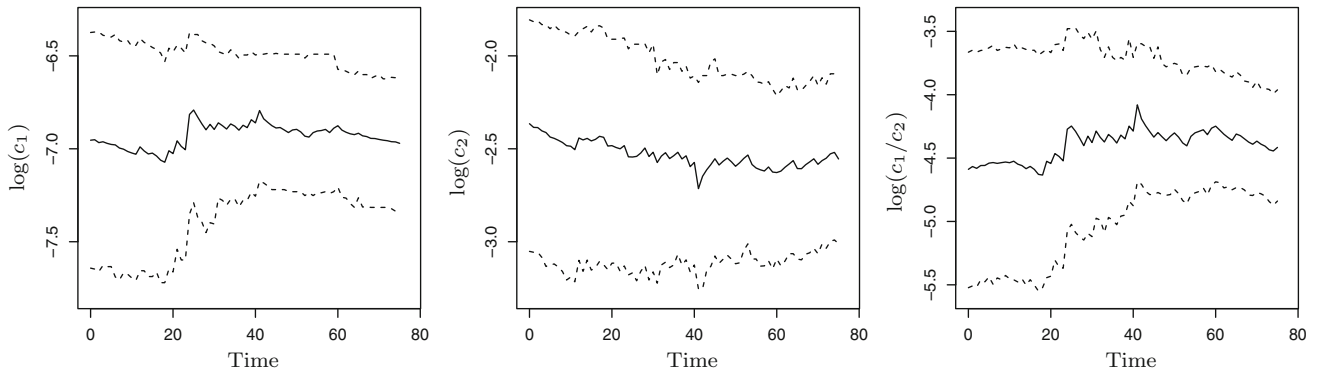
Filter	$N_x$	CPU (s)	Bias (RMSE)			
			$\widehat{E}(\log(c_1) y_{1:T})$	$\widehat{E}(\log(c_2) y_{1:T})$	$\widehat{SD}(\log(c_1) y_{1:T})$	$\widehat{SD}(\log(c_2) y_{1:T})$
Bootstrap	404	175	0.068 (0.022)	0.017 (0.023)	−0.026 (0.012)	−0.011 (0.017)
Alive	77	256	0.033 (0.026)	−0.006 (0.035)	−0.020 (0.014)	−0.008 (0.022)
Auxiliary (meth. 1)	81	45	0.041 (0.024)	−0.024 (0.028)	−0.024 (0.014)	−0.010 (0.016)
Auxiliary (meth. 2)	174	162	0.037 (0.028)	0.040 (0.034)	−0.022 (0.019)	−0.024 (0.020)

All results are obtained by averaging over 100 runs of each SMC<sup>2</sup> scheme

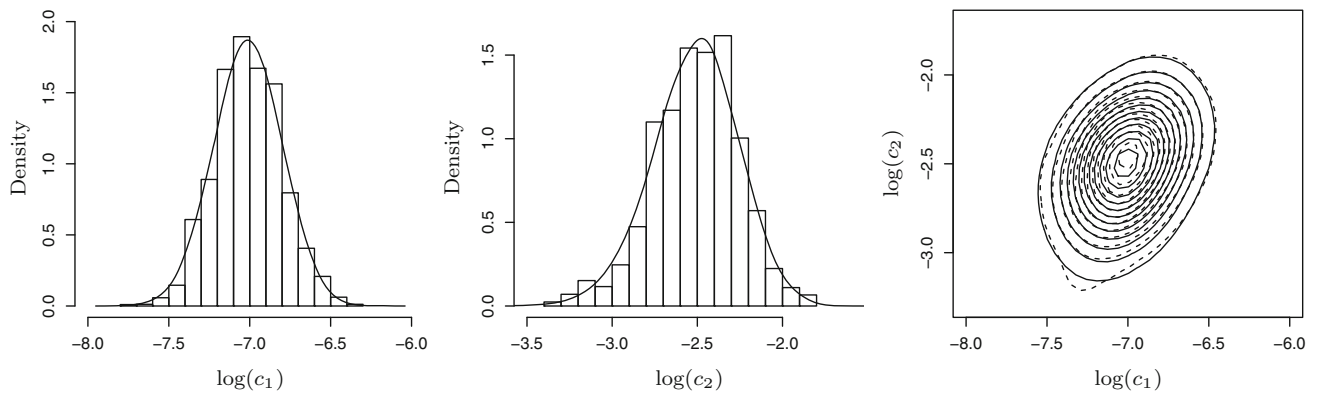




**Fig. 1** SIR epidemic model (Abakaliki data). Left panel: Effective sample size (ESS) against time. Middle panel: Acceptance rate against time. Right panel: Number of state particles  $N_x$  against time. Horizontal lines indicate the thresholds at which resampling and doubling of  $N_x$  take place. All results are based on a single typical run of an SMC<sup>2</sup> scheme using the bootstrap (solid line), alive (dashed line) and auxiliary method 1 (dotted line) particle filters. Auxiliary method 2 is omitted for ease of exposition



**Fig. 2** SIR epidemic model (Abakaliki data). Marginal posterior mean (solid line) and 95% credible interval (dashed lines) for  $\log(c_1)$  (left),  $\log(c_2)$  (middle) and  $\log(c_1/c_2)$  (right) based on the output of the auxiliary SMC<sup>2</sup> scheme



**Fig. 3** SIR epidemic model (Abakaliki data). Left and middle panels: marginal posterior distributions based on the output of the auxiliary SMC<sup>2</sup> scheme (histograms) and pMCMC scheme (kernel density estimates). Right panel: Contour plot of the joint posterior from the output of the auxiliary SMC<sup>2</sup> scheme (dashed lines) and pMCMC (solid lines)

size (ESS), rarely falling below the threshold that would trigger the resample-move step. In spite of this desirable behaviour, the scheme requires repeatedly forward simulating the process at each time point to obtain  $N_x$  matches,

resulting in a CPU cost that is almost 1.5 times larger than that obtained for the bootstrap driven scheme. Both auxiliary schemes outperform the bootstrap implementation, with method 1 by a factor of 3.9 in terms of CPU cost. Finally,

we note that the SMC<sup>2</sup> scheme allows for sequential learning of the rate constants as well as the basic reproduction number  $R_0 = c_1/c_2$ —see Fig. 2 showing marginal posterior means and 95% credible intervals against time. Figure 3 compares the output of an SMC<sup>2</sup> scheme with the output of a long run of pMCMC and demonstrates that accurate fully Bayesian inferences about the parameters are possible, even when using relatively few parameter particles.

#### 4.1.1 Comparison with MCMC

Here, we assess the utility of the auxiliary particle filter (method 1) driven SMC<sup>2</sup> scheme as an offline inference scheme by comparing its performance to that of two competing MCMC schemes, namely the particle MCMC scheme used by Golightly and Wilkinson (2015) and a data augmentation scheme first introduced by O’Neill and Roberts (1999) and Gibson and Renshaw (1998).

As discussed earlier, the likelihood of the observed data (i.e. removal times) is challenging to compute. The reason is that one has to integrate out all the possible configurations of infection times that are consistent with the data; in other words, those that do not result in the epidemic ceasing before the last removal time. One way to overcome this issue is to introduce the unobserved infection times as additional variables which will allow us to compute an augmented likelihood. Combining the augmented likelihood with prior distributions on the infection rate ( $c_1$ ) and removal rate ( $c_2$ ), we can then explore the joint posterior density of the infection times,  $c_1$  and  $c_2$  using a data-augmented Markov Chain Monte Carlo scheme (DA-MCMC).

A vanilla DA-MCMC algorithm consists of updating  $c_1$ ,  $c_2$  and the infection times from their corresponding full conditional (posterior) densities. It turns out that the full conditional densities for  $c_1$  and  $c_2$  have standard forms and can be updated using a Gibbs step; in fact, both full conditional densities are Gamma densities. The infection times are less straightforward to deal with because the full conditional distribution of each infection time is not of a standard form. However, they can be updated by using a Metropolis-Hastings step. This is done by proposing a new infection time and accepting that proposed infection times with some probability determined by the Metropolis-Hastings ratio. In particular, a new infection time for the  $j$ th individual,  $i_j^*$ , is proposed by drawing  $X \sim \text{Exp}(c_2)$  and setting  $i_j^* = r_j - X$  where  $r_j$  denotes the corresponding removal time of individual  $j$ .

To provide a challenging scenario, we assumed a fixed population size of  $n = 1000$ , an infection rate of  $c_1 = 0.0013$ , a removal rate of  $c_2 = 1$  and generated a synthetic data set consisting of 622 inter-removal times, equivalent to 622 measurements of  $X_1 + X_2$ . For simplicity, we assume that the initial condition  $x_0 = (n - 1, 1)'$  is known. We took

vague Exponential  $\text{Exp}(0.001)$  priors for each rate constant and performed 50 runs of (auxiliary) SMC<sup>2</sup>, pMCMC and MCMC-DA with the following settings.

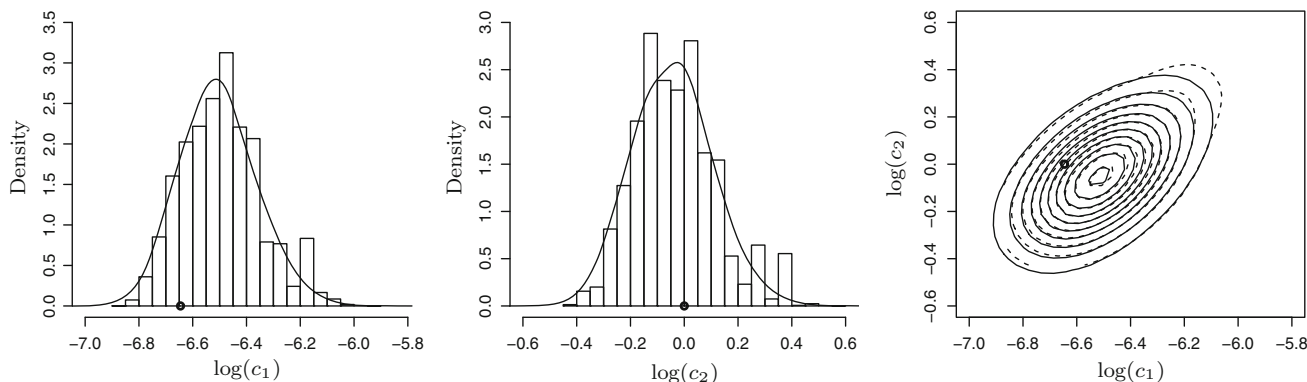
1. *SMC<sup>2</sup>* We took  $N_c = 5000$ , an ESS-threshold of  $\gamma = 50\%$  and an initial number of state particles of  $N_x = 100$ . The value of  $N_x$  was doubled if the acceptance rate calculated in the resample-move step fell below 20%. Note that initialising with a sample from the vague prior would result in very few parameter particles consistent with the first observation. This problem can be alleviated, for example, by partitioning the interval  $[0, 1]$  into  $m + 1$  equally spaced intermediate time points and targeting the tempered posteriors  $p(c)p(y_1|c)^{i/m}$ ,  $i = 0, 1, \dots, m$ . We adopted an alternative solution and performed 10000 pMCMC iterations using the first 10 observations (with  $N_x = 100$ ), thinned by a factor of 2 and then ran SMC<sup>2</sup> for the remaining 612 observations, having initialised with the pMCMC output.
2. *pMCMC* Following the practical advice of Sherlock et al. (2015), the number of state particles was chosen so that the variance of the estimator of the log-posterior at the posterior median (obtained from a pilot run) was around 2. This gave  $N_x = 1200$ . A random walk proposal was used for the log-parameters with the variance of the Gaussian innovations taken to be  $\widehat{\text{Var}}(\log(c)|y_{1:T})$  (estimated from a pilot run) and scaled to give an acceptance rate of around 10% – 15%. The same pilot run was used to obtain the estimate  $\widehat{E}(\log(c)|y_{1:T})$ , and the main monitoring runs were initialised using this value.
3. *MCMC-DA* It has been illustrated that in practice (Kypraios 2007), if the infection-time update step is repeated several times in each iteration of the MCMC algorithm then mixing can improve substantially. Denote the fraction of infection times to be update in each MCMC step by  $\delta$ . After running a number of short pilot runs with  $\delta \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ , we found that  $\delta = 0.5$  was optimal in terms of minimising autocorrelation time (defined below). The main monitoring runs then used  $\delta = 0.5$  and were initialised with the same values used for the pMCMC runs.

Note that the number of iterations of pMCMC and MCMC-DA performed for the 50 runs was determined by the CPU cost of each run of SMC<sup>2</sup>. Consequently, all results are reported for the same computational budget. The results are summarised in Table 3 and Fig. 4. From the latter, it is clear that the output of SMC<sup>2</sup> is comparable with that of pMCMC. The two competing MCMC schemes can be directly compared by computing autocorrelation time (ACT), sometimes referred to as inefficiency and can be interpreted as the factor by which the number of iterations ( $n_{\text{iters}}$ ) should be multiplied, to obtain the same level of precision as using  $n_{\text{iters}}$  iid

**Table 3** SIR epidemic model (synthetic data).  $N_x$  at time  $T$ , maximum autocorrelation time (mACT), bias (and RMSE in parentheses) of estimators of the posterior expectations  $E(\log(c_1)|y_{1:T})$ ,  $E(\log(c_2)|y_{1:T})$  and standard deviations  $SD(\log(c_1)|y_{1:T})$ ,  $SD(\log(c_2)|y_{1:T})$

Method	$N_x$	mACT	Bias (RMSE)			
			$\widehat{E}(\log(c_1) y_{1:T})$	$\widehat{E}(\log(c_2) y_{1:T})$	$\widehat{SD}(\log(c_1) y_{1:T})$	$\widehat{SD}(\log(c_2) y_{1:T})$
SMC <sup>2</sup>	1120	–	–0.004 (0.017)	–0.009 (0.016)	–0.009 (0.009)	–0.008 (0.009)
pMCMC	1200	29	0.001 (0.006)	0.001 (0.006)	–0.003 (0.003)	–0.001 (0.003)
MCMC-DA	–	232	–0.006 (0.017)	–0.002 (0.017)	0.036 (0.014)	0.037 (0.015)

All results are obtained by averaging over 50 runs of each scheme. Note that mACT for MCMC-DA has been scaled to correspond to the average number of iterations of pMCMC



**Fig. 4** SIR epidemic model (synthetic data). Left and middle panels: marginal posterior distributions based on the output of the auxiliary SMC<sup>2</sup> scheme (histograms) and pMCMC scheme (kernel density estimates)

Right panel: Contour plot of the joint posterior from the output of the auxiliary SMC<sup>2</sup> scheme (dashed lines) and pMCMC (solid lines). The true values of  $\log(c_1)$  and  $\log(c_2)$  are indicated

posterior draws. The ACT for a particular series of parameter values is given by

$$1 + 2 \sum_{k=1}^{\infty} \rho_k$$

where  $\rho_k$  is the autocorrelation function for the series at lag  $k$ . The ACT can be estimated using the R package CODA (Plummer et al. 2006).

The MCMC-DA scheme is relatively cheap, with the CPU budget affording runs of around  $10^6$  iterations on average. By comparison, the pMCMC scheme typically used around  $2.5 \times 10^4$  iterations. However, the mixing of MCMC-DA is very poor, due to the dependence between the parameter values and the imputed infection times. For pMCMC, a joint update of the parameters and latent infection times is used (thereby side-stepping the issue of high correlation between the two) and mixing is much improved. Consequently, for MCMC-DA, the maximum (over each parameter series) ACT is around 8 times larger than that for pMCMC (after matching iteration numbers). Not surprisingly, estimators of the marginal posterior means and standard deviations for the log rate constants based on MCMC-DA exhibit biases and root-mean-square errors that are significantly larger than those

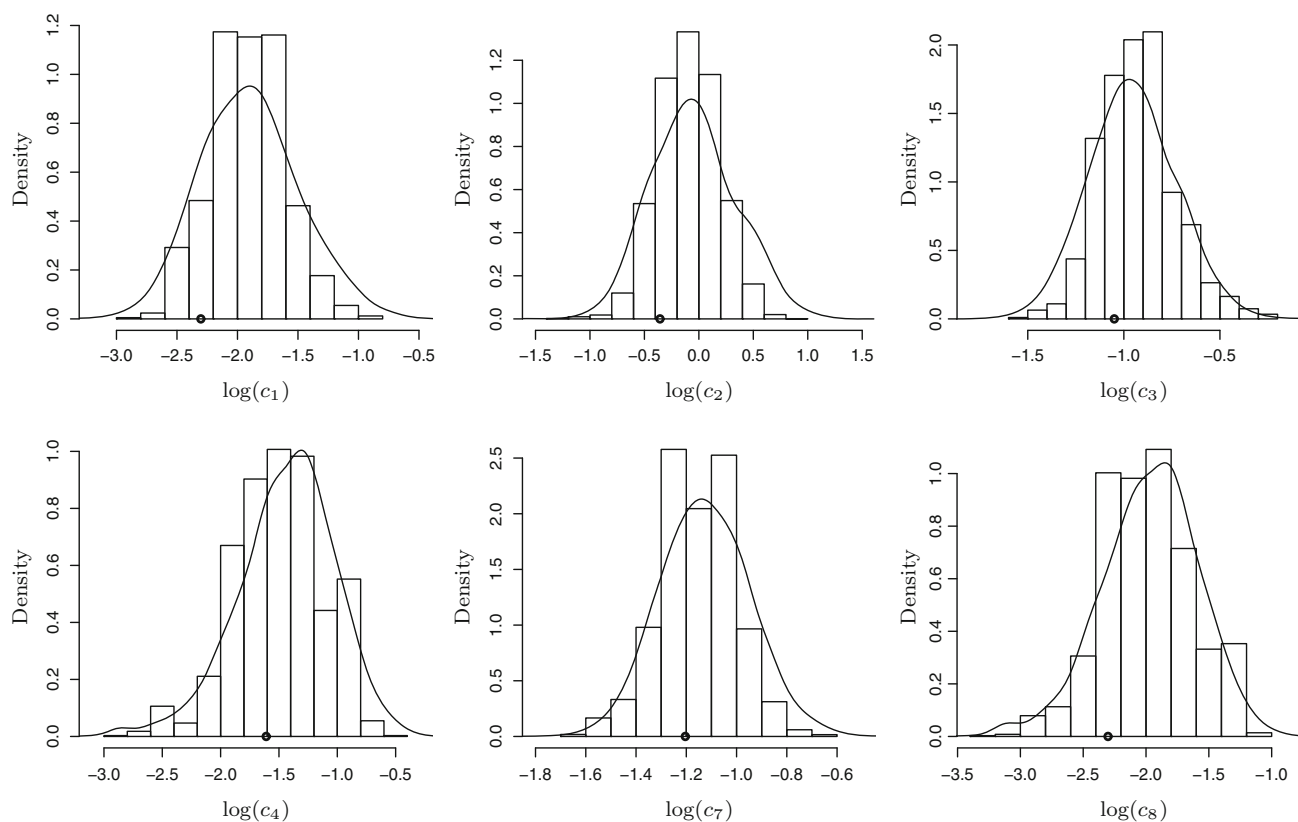
obtained for pMCMC. Using SMC<sup>2</sup> gives output comparable to that of pMCMC, with all biases within an order of magnitude of those for pMCMC, and all RMSE values within a factor of 3. Moreover, it should be noted that we are comparing against a pMCMC scheme with (close to) optimal settings obtained from pilot runs. SMC<sup>2</sup> requires minimal tuning by comparison, yet appears to be an effective offline inference tool in this example.

### 4.2 Prokaryotic autoregulation

Using the model of prokaryotic autoregulation described in Sect. 2.2, we simulated two synthetic data sets (denoted  $\mathcal{D}_1$  and  $\mathcal{D}_2$ ) consisting of 101 observations at integer times on RNA and total protein counts,  $P + 2P_2$ , so that DNA, P and  $P_2$  are not observed exactly. Moreover, we corrupt the observations by adding independent, zero-mean Gaussian innovations to each count. The components making up the observation in (1) are

$$P' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}.$$

To assess the effect of measurement error, we fix  $\sigma_2 = 1$  and take  $\sigma_1 = 1$  for data set  $\mathcal{D}_1$  and  $\sigma_1 = 0.1$  for  $\mathcal{D}_2$ . Following



**Fig. 5** Prokaryotic autoregulation (synthetic data set  $\mathcal{D}_2$ ). Marginal posterior distributions based on the output of the auxiliary SMC<sup>2</sup> scheme (histograms) and pMCMC scheme (kernel density estimates). The true values of the (log) rate constants are indicated

Golightly and Wilkinson (2005), the rate constants used to generate the data were

$$c = (0.1, 0.7, 0.35, 0.2, 0.1, 0.9, 0.3, 0.1).$$

We assume that the initial condition  $x_0 = (8, 8, 8, 5)'$ , the measurement error variances and the rate constants of the reversible dimerisation reactions ( $c_5$  and  $c_6$ ) are known leaving 6 parameters as the object of inference.

We took independent Gamma  $Ga(1, 0.5)$  priors for each rate constant and applied SMC<sup>2</sup> schemes based on the bootstrap and auxiliary (with propagation method 1) particle filters. In each case we took  $N_c = 5000$ , an ESS-threshold of  $\gamma = 50\%$  and an initial number of state particles of  $N_x = 50$ . The value of  $N_x$  was doubled if the acceptance rate calculated in the resample-move step fell below 20%.

Figure 5 shows marginal posteriors based on the output of auxiliary SMC<sup>2</sup> and a long run of pMCMC. We note that even with 6 unknown parameters, the SMC<sup>2</sup> scheme gives accurate inferences despite using relatively few parameter particles. Table 4 and Fig. 6 summarise the output of each SMC<sup>2</sup> scheme. We again compare the accuracy of each scheme via bias and RMSE of the estimators of the marginal posterior means and standard deviations of the (log) rate

constants. Bias and RMSE were computed by comparing estimators based on 50 runs of each SMC<sup>2</sup> scheme with reference values obtained from a long run of pMCMC (with  $5 \times 10^5$  iterations). Table 4 displays these quantities for  $\log(c_1)$  and  $\log(c_2)$  corresponding to the reversible dimer binding and unbinding reactions. Similar results (not shown) are obtained for the remaining unknown rate constants. Both the bootstrap and auxiliary particle filter-driven schemes give comparable bias and RMSE values, and we therefore compare their overall performance using CPU cost. Not surprisingly, as the measurement error is reduced, both schemes require increased numbers of state particles,  $N_x$ , although the relative increase is much smaller when using auxiliary SMC<sup>2</sup>. Consequently, for data set  $\mathcal{D}_1$  ( $\sigma_1 = 1$ ), auxiliary SMC<sup>2</sup> outperforms bootstrap SMC<sup>2</sup> in terms of CPU time by around a factor of 2. This increases to a factor of around 4 for data set  $\mathcal{D}_2$  ( $\sigma_1 = 0.1$ ).

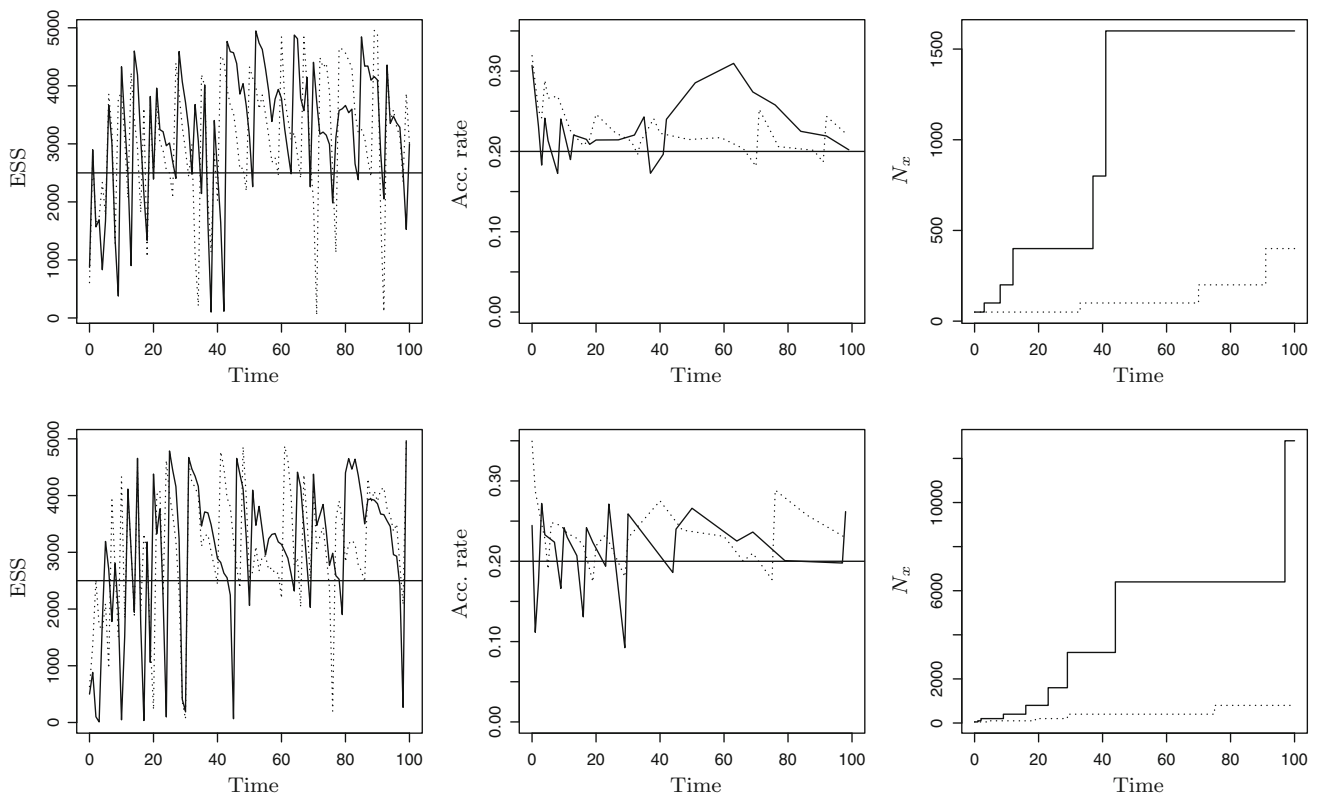
## 5 Discussion

Performing fully Bayesian inference for the rate constants governing complex stochastic kinetic models necessitates the use of computationally intensive Markov chain Monte Carlo

**Table 4** Prokaryotic autoregulation.  $N_x$  at time  $T$ , CPU time (in minutes), bias (and RMSE in parentheses) of estimators of the posterior expectations  $E(\log(c_1)|y_{1:T})$ ,  $E(\log(c_2)|y_{1:T})$  and standard deviations  $SD(\log(c_1)|y_{1:T})$ ,  $SD(\log(c_2)|y_{1:T})$

Filter	$N_x$	CPU (m)	Bias (RMSE)			
			$\hat{E}(\log(c_1) y_{1:T})$	$\hat{E}(\log(c_2) y_{1:T})$	$\widehat{SD}(\log(c_1) y_{1:T})$	$\widehat{SD}(\log(c_2) y_{1:T})$
$\mathcal{D}_1 (\sigma_1 = 1, \sigma_2 = 1)$						
Bootstrap	2688	495	-0.281 (0.051)	-0.066 (0.053)	-0.094 (0.030)	-0.077 (0.036)
Auxiliary	564	242	-0.129 (0.082)	-0.027 (0.067)	-0.098 (0.064)	-0.084 (0.033)
$\mathcal{D}_2 (\sigma_1 = 0.1, \sigma_2 = 1)$						
Bootstrap	8000	1905	-0.011 (0.082)	-0.088 (0.063)	-0.063 (0.056)	-0.063 (0.046)
Auxiliary	1120	474	-0.047 (0.048)	-0.029 (0.047)	-0.079 (0.020)	-0.058 (0.024)

All results are obtained by averaging over 50 runs of each scheme



**Fig. 6** Prokaryotic autoregulation using synthetic data sets  $\mathcal{D}_1$  (top panel) and  $\mathcal{D}_2$  (bottom panel). Left panel: Effective sample size (ESS) against time. Middle panel: Acceptance rate against time. Right panel: Number of state particles  $N_x$  against time. Horizontal lines indicate

the thresholds at which resampling and doubling of  $N_x$  take place. All results are based on a single typical run of an SMC<sup>2</sup> scheme using the bootstrap (solid line) and auxiliary (dotted line) particle filters

(MCMC) methods. The intractability of the observed-data likelihood further complicates matters and is usually dealt with through the use of data augmentation or by replacing the intractable likelihood by an unbiased estimate. Careful implementation of the latter results in a pseudo-marginal Metropolis-Hastings scheme, and, when using a particle filter to obtain likelihood estimates, the algorithm may be referred to as particle MCMC (pMCMC). However, such methods often require careful tuning and initialisation and do not allow

for efficient sequential learning of the parameters (and latent states).

We have therefore focused on a recently proposed SMC<sup>2</sup> scheme, which can be seen as the pseudo-marginal analogue of the iterated batch importance sampling (IBIS) scheme (Chopin 2002), and allows sequential learning of the parameters of interest. The simplest implementation uses a bootstrap particle filter both to compute observed-data likelihood increments and drive a rejuvenation step (so-called

resample move) where all parameter particles are mutated through a pMCMC kernel. This simple implementation is appealing—for example, only the ability to evaluate the density associated with the observation equation, and generate forward realisations from the Markov jump process is required. However, this ‘likelihood-free’ implementation is likely to be extremely inefficient when observations are informative, e.g. when there is relatively little measurement error compared to intrinsic stochasticity. We eschew the simplest implementation in favour of an SMC<sup>2</sup> scheme that is driven by an auxiliary particle filter (APF). That is, the APF is used both to estimate the observed-data likelihood contributions and drive the resample-move step. We compared this approach using two applications: an SIR epidemic model fitted to real data and a simple model of prokaryotic autoregulation fitted to synthetic data.

We find that the proposed approach offers significant gains in computational efficiency relative to the bootstrap filter-driven implementation, whilst still maintaining an accurate particle representation of the full posterior. The computational gains are amplified when intrinsic stochasticity dominates external noise (e.g. measurement error). Use of an appropriate propagation mechanism is crucial in this case, since the probability of generating an (unconditioned) realisation of the latent jump process that is consistent with the next observation, diminishes as either the observation variance decreases or the number of observed components increases.

Using synthetic data and the SIR epidemic model, we also compared the efficiency of SMC<sup>2</sup> with two competing MCMC schemes, namely the APF driven particle MCMC scheme of Golightly and Wilkinson (2015) and a ubiquitously applied data augmentation (DA) scheme (O’Neill and Roberts 1999; Gibson and Renshaw 1998). We find that the DA scheme suffers intolerably poor mixing due to dependence between the latent infection times and the static parameters (see also McKinley et al. (2014)). The pMCMC scheme, which can be seen as the pseudo-marginal analogue of an idealised marginal scheme, offers over an order of magnitude increase in terms of overall efficiency (as measured by autocorrelation time for a fixed computational budget) over DA. The APF driven SMC<sup>2</sup> scheme gives comparable output to that of pMCMC in terms of accuracy (as measured by bias and root-mean-squared error of key posterior summaries). However, we stress again that unlike pMCMC, SMC<sup>2</sup> is simple to initialise, avoids the need for tedious pilot runs, performs sequential learning of the parameters of interest and allows for a computationally efficient estimator of the model evidence. Although not pursued here, model selection is an important problem within the stochastic kinetic framework (see e.g. Drovandi and McCutchan (2016) and the references therein for recent discussions).

## 5.1 Use of other particle filters

The development of an auxiliary particle filter-driven SMC<sup>2</sup> scheme as considered in this paper is possible due to the tractability of the complete data likelihood  $p(x_{(t-1,t]}|x_{t-1}, c)$  for each observation time  $t$ . This tractability may permit the use of other particle filtering strategies. For example, particle Gibbs with ancestor sampling (Lindsten et al. 2014) allows for efficient sampling of state trajectories and could be used in the rejuvenation step in SMC<sup>2</sup>. Recent work by Guarniero et al. (2016) combines ideas underpinning the twisted particle filter of Whiteley and Lee (2014) and the APF to give the iterated APF (iAPF). The algorithm approximates an idealised particle filter where observed-data likelihood estimates have zero variance. Consequently, use of this approach in an SMC<sup>2</sup> requires further attention, although it would appear the iAPF algorithm is at present limited to a class of state space models with conjugate latent processes. Its utility within the SKM framework is therefore less clear.

## 5.2 Further considerations

This work can be directly extended in a number of ways. In our application of the APF, we assumed a constant preweight for each parameter particle. Devising a preweight that is both computationally cheap and accurate remains of interest. In addition, the best performing propagation method is derived using a linear Gaussian approximation to the number of reaction events in an interval of interest, conditional on the next observation. Improvements to this construct that allow for a more accurate approximation of the intractable conditioned process are the subject of ongoing work. Although not considered here, the SMC<sup>2</sup> scheme appears to be particularly amenable to parallelisation over parameter particles, since observed-data likelihood estimates can be computed separately for each parameter value. The use of parallel resampling algorithms (Murray et al. 2016) also merits further attention, to allow full use of modern computational architectures. Finally, we note that the resample-move step may benefit from recent work on correlated pseudo-marginal schemes (Dahlin et al. 2015; Deligiannidis et al. 2016).

**Acknowledgements** The authors would like to thank the associate editor and anonymous referee for their suggestions for improving this paper.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

**Algorithm 3** Alive particle filter

1. Initialisation ( $t = 1$ ). Set  $\hat{p}(y_1|c) = 1$  and  $x_1^i = y_1, i = 1, \dots, N_x$ . For  $t = 2, 3, \dots, T$ :
2. For  $i = 1, 2, \dots$  until  $i = n_t$  is reached such that  $\sum_{j=1}^i I(x_j^i, y_t) = N_x + 1$ :
  - (a) Sample  $a_{t-1}^i$  uniformly from  $\{1, \dots, N_x\}$ .
  - (b) Sample  $x_{(t-1,t)}^i \sim p(\cdot | x_{t-1}^{a_{t-1}^i}, c)$  using Gillespie's direct method.
  - (c) Calculate

$$I(x_t^i, y_t) = \begin{cases} 1, & x_t^i = y_t \\ 0, & \text{otherwise} \end{cases}$$

3. Compute the current estimate of observed-data likelihood  $\hat{p}(y_{1:t}|c) = \hat{p}(y_{1:t-1}|c)\hat{p}(y_t|y_{1:t-1}, c)$  where

$$\hat{p}(y_t|y_{1:t-1}, c) = \frac{N_x}{n_t - 1}.$$

**A Appendix****A.1 Alive SMC<sup>2</sup>**

Consider the case that  $\Sigma = 0$  so that (a subset of) the components of  $X_t$  are observed without error. Running a bootstrap particle filter in this scenario is likely to be problematic, since only trajectories which match the observation at each time  $t$  will be assigned a non-zero weight. To circumvent this problem, Drovandi and McCutchan (2016) use the alive particle filter of Del Moral et al. (2015) inside the SMC<sup>2</sup> scheme. Essentially, at time  $t$ ,  $N_x$  particles from time  $t-1$  are resampled and propagated forward (using Gillespie's direct method) until  $N_x + 1$  matches are obtained (where a match has  $x_t = y_t$ ). This approach can be repeated for each time point and an unbiased estimator of the observed-data likelihood  $p(y_{1:t}|c)$  can then be obtained (Del Moral et al. 2015).

The alive particle filter is described in Algorithm 3. Note that we have assumed for simplicity that  $x_1$  is known, although a more general scenario with uncertain  $x_1$  is easily accommodated by augmenting  $c$  to include the unobserved components of  $x_1$ . The alive SMC<sup>2</sup> algorithm is obtained by running the alive particle filter in steps 2(a) and 3(b) of Algorithm 2.

**References**

- Andersson, Hk, Britton, T.: Stochastic Epidemic Models and Their Statistical Analysis, volume 151 of Lecture Notes in Statistics. Springer, New York (2000)
- Andrieu, C., Doucet, A., Holenstein, R.: Particle Markov chain Monte Carlo methods (with discussion). *J. R. Stat. Soc. B* **72**(3), 1–269 (2010)
- Arkin, A., Ross, J., McAdams, H.H.: Stochastic kinetic analysis of developmental pathway bifurcation in phage  $\lambda$ -infected *Escherichia coli* cells. *Genetics* **149**, 1633–1648 (1998)
- Bailey, N.T.J.: The Mathematical Theory of Infectious Diseases and Its Applications, 2nd edn. Hafner Press [Macmillan Publishing Co., Inc.], New York (1975)
- Boys, R.J., Giles, P.R.: Bayesian inference for stochastic epidemic models with time-inhomogeneous removal rates. *J. Math. Biol.* **55**, 223–247 (2007)
- Boys, R.J., Wilkinson, D.J., Kirkwood, T.B.L.: Bayesian inference for a discretely observed stochastic kinetic model. *Statist. Comput.* **18**, 125–135 (2008)
- Carvalho, C.M., Johannes, M.S., Lopes, H.F., Polson, N.G.: Particle learning and smoothing. *Stat. Sci.* **25**, 88–106 (2010)
- Chopin, N.: A sequential particle filter for static models. *Biometrika* **89**, 539–552 (2002)
- Chopin, N., Iacobucci, A., Marin, J.-M., Mengersen, K., Robert, C.P., Ryder, R., Schäfer, C.: On Particle Learning. Available from [arxiv: 1006.0554](https://arxiv.org/abs/1006.0554) (2010)
- Chopin, N., Jacob, P.E., Papaspiliopoulos, O.: SMC<sup>2</sup>: an efficient algorithm for sequential analysis of state space models. *J. R. Stat. Soc. B* **75**, 397–426 (2013)
- Dahlin, J., Lindsten, F., Kronander, J., Schon, T.B.: Accelerating pseudo-marginal Metropolis-Hastings by correlating auxiliary variables. Available from [arxiv:1511.05483v1](https://arxiv.org/abs/1511.05483v1) (2015)
- Del Moral, P.: Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications. Springer, New York (2004)
- Del Moral, P., Jasra, A., Lee, A., Yau, C., Zhang, X.: The alive particle filter and its use in particle Markov chain Monte Carlo. *Stoch. Anal. Appl.* **33**, 943–974 (2015)
- Deligiannidis, G., Doucet, A., Pitt, M.K.: The correlated pseudo-marginal method. Available from [arxiv: 1511.04992v3](https://arxiv.org/abs/1511.04992v3) (2016)
- Doucet, A., de Freitas, N., Gordon, N.: Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science. Springer, New York (2001)
- Drovandi, C.C., McCutchan, R.A.: Alive SMC<sup>2</sup>: Bayesian model selection for low-count time series models with intractable likelihoods. *Biometrics* **72**, 344–353 (2016)
- Fearnhead, P.: Markov chain Monte Carlo, sufficient statistics, and particle filters. *J. Comput. Graph. Stat.* **11**, 848–862 (2002)
- Fearnhead, P.: Computational methods for complex stochastic systems: a review of some alternatives to MCMC. *Statist. Comput.* **18**, 151–171 (2008)
- Fearnhead, P., Meligkotsidou, L.: Exact filtering for partially observed continuous time models. *J. R. Statist. Soc. B* **66**(3), 771–789 (2004)
- Ferm, L., Lötstedt, P., Hellander, A.: A hierarchy of approximations of the master equation scaled by a size parameter. *J. Sci. Comput.* **34**(2), 127–151 (2008)
- Fulop, A., Li, J.: Efficient learning via simulation: a marginalized resample-move approach. *J. Econom.* **176**, 146–161 (2013)
- Gibson, G.J., Renshaw, E.: Estimating parameters in stochastic compartmental models using Markov chain methods. *IMA J. Math. Appl. Med. Biol.* **15**, 19–40 (1998)
- Gilks, W.R., Berzuini, C.: Following a moving target—Monte Carlo inference for dynamic Bayesian models. *J. R. Statist. Soc. A* **63**, 127–146 (2001)
- Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* **81**, 2340–2361 (1977)
- Golightly, A., Wilkinson, D.J.: Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics* **61**(3), 781–788 (2005)
- 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., Wilkinson, D.J.: Bayesian inference for markov jump processes with informative observations. *SAGMB* **14**(2), 169–188 (2015)
- Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F* **140**, 107–113 (1993)
- Guarniero, P., Johansen, A.M., Lee, A.: The iterated auxiliary particle filter. Available from [arxiv: 1511.06286v2](https://arxiv.org/abs/1511.06286v2) (2016)
- Jacob, P.: Sequential Bayesian inference for implicit hidden Markov models and current limitations. *ESAIM Proc. Surv.* **51**, 24–48 (2015)
- Kypraios, T.: Efficient Bayesian Inference for Partially Observed Stochastic Epidemics and A New Class of Semi-Parametric Time Series Models. Ph.D. thesis, Department of Mathematics, Lancaster University (2007)
- Lin, J., Ludkovski, M.: Sequential Bayesian inference in hidden Markov stochastic kinetic models with application to detection and response to seasonal epidemics. *Statist. Comput.* **24**, 1047–1062 (2013)
- Lindsten, F., Jordan, M.I., Schön, T.B.: Particle Gibbs with ancestor sampling. *J. Mach. Learn. Res.* **15**, 2145–2184 (2014)
- Liu, J., West, M.: Combined parameter and state estimation in simulation-based filtering. In: Doucet, A., de Freitas, N., Gordon, N. (eds.) *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001)
- McKinley, T.J., Ross, J.V., Deardon, R., Cook, A.R.: Simulation-based Bayesian inference for epidemic models. *Comput. Stat. Data Anal.* **71**, 434–447 (2014)
- Murray, L.M., Lee, A., Jacob, P.E.: Parallel resampling in the particle filter. *J. Comput. Graph. Stat.* **25**(3), 789–805 (2016)
- O’Neill, P.D., Roberts, G.O.: Bayesian inference for partially observed stochastic epidemics. *J. R. Statist. Soc. A* **162**, 121–129 (1999)
- Pitt, M., Shephard, N.: Auxiliary variable based particle filters. In: Doucet, A., de Freitas, N., Gordon, N. (eds.) *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001)
- Pitt, M.K., dos Santos Silva, R., Giordani, P., Kohn, R.: On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *J. Econom.* **171**(2), 134–151 (2012)
- Pitt, M.K., Shephard, N.: Filtering via simulation: auxiliary particle filters. *J. Am. Stat. Assoc.* **446**, 590–599 (1999)
- Plummer, M., Best, N., Cowles, K., Vines, K.: CODA: convergence diagnosis and output analysis for MCMC. *R News* **6**(1), 7–11 (2006)
- Sherlock, C., Thiery, A., Roberts, G.O., Rosenthal, J.S.: On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Ann. Stat.* **43**(1), 238–275 (2015)
- Stockdale, J.E., Kypraios, T., O’Neill, P.D.: Modelling and Bayesian analysis of the Abakaliki smallpox data. *Epidemics* (to appear) (2017)
- Storvik, G.: Particle filters in state space models with the presence of unknown static parameters. *IEEE Trans. Signal Process.* **50**, 281–289 (2002)
- Whiteley, N., Lee, A.: Twisted particle filters. *Ann. Stat.* **42**(1), 115–141 (2014)
- Wilkinson, D. J.: *Stochastic Modelling for Systems Biology*, 2nd edn. Chapman & Hall/CRC Press, Boca Raton, FL (2012)