Stat Comput
DOI 10.1007/s11222-011-9271-y

An adaptive sequential Monte Carlo method for approximate Bayesian computation

Pierre Del Moral · Arnaud Doucet · Ajay Jasra

Received: 30 May 2011 / Accepted: 7 July 2011
© Springer Science+Business Media, LLC 2011

Abstract Approximate Bayesian computation (ABC) is a popular approach to address inference problems where the likelihood function is intractable, or expensive to calculate. To improve over Markov chain Monte Carlo (MCMC) implementations of ABC, the use of sequential Monte Carlo (SMC) methods has recently been suggested. Most effective SMC algorithms that are currently available for ABC have a computational complexity that is quadratic in the number of Monte Carlo samples (Beaumont et al., Biometrika 86:983–990, 2009; Peters et al., Technical report, 2008; Toni et al., J. Roy. Soc. Interface 6:187–202, 2009) and require the careful choice of simulation parameters. In this article an adaptive SMC algorithm is proposed which admits a computational complexity that is linear in the number of samples and adaptively determines the simulation parameters. We demonstrate our algorithm on a toy example and on a birth-death-mutation model arising in epidemiology.

Keywords Approximate Bayesian computation · Markov chain Monte Carlo · Sequential Monte Carlo

P. Del Moral
Centre INRIA Bordeaux Sud-Ouest & Institut de Mathématiques, Université Bordeaux I, 33405 Talence cedex, France
e-mail: Pierre.Del-Moral@inria.fr

A. Doucet (✉)
Department of Statistics, University of Oxford, Oxford, OX1 3TG, UK
e-mail: adoucet@stats.ox.ac.uk

A. Jasra
Department of Statistics and Applied Probability, National University of Singapore, Singapore, 117546, Singapore
e-mail: staja@nus.edu.sg

Published online: 03 August 2011

1 Introduction

1.1 Background

Assume we are given a Bayesian model where $\pi(\theta)$ denotes the prior density of the parameter $\theta \in \Theta$ and $f(y|\theta)$ the likelihood of observations $y \in \mathcal{D}$. In this context, inference relies on the resulting posterior density $\pi(\theta|y)$. Unfortunately if the likelihood term $f(y|\theta)$ is expensive or impossible to calculate, it is difficult to use standard computational tools to sample from $\pi(\theta|y)$. ABC is an alternative to such techniques that only requires being able to sample pseudo-observations X from $f(\cdot|\theta)$. In its most common form, ABC draws inference from the following modified posterior density on $\Theta \times \mathcal{D}$

$$\pi_\epsilon(\theta, x|y) = \frac{\pi(\theta) f(x|\theta) \mathbb{I}_{A_{\epsilon,y}}(x)}{\int_{A_{\epsilon,y} \times \Theta} \pi(\theta) f(x|\theta) dx d\theta} \tag{1}$$

with $\epsilon > 0$ a tolerance level, $\mathbb{I}_B(\cdot)$ the indicator function of a given set B and $x \in \mathcal{D}$ corresponds to some pseudo-observations. The set $A_{\epsilon,y}$ corresponds to the set of pseudo-observations which are close in some sense to the true observations y. It is formally defined as follows

$$A_{\epsilon,y} = \left\{ z \in \mathcal{D} : \rho\big(\eta(z), \eta(y)\big) < \epsilon \right\}$$

where $\eta : \mathcal{D} \to \mathcal{S}$ represents some summary statistics and $\rho : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$ a distance function. A few variations over the ABC posterior presented here have been proposed. For example, it is possible to substitute to the indicator function appearing in (1) a smooth kernel function. The ideas developed in this paper still apply in this context but, for ease of presentation, we will restrict ourselves here to an indicator function. We refer the reader to reference Marin et al. (2011)

which provides an excellent up-to-date survey of ABC methods.

Designing efficient MCMC algorithms to sample from $\pi_\epsilon(\theta, x|y)$ can be difficult. This is why SMC methods have recently become prominent in the ABC context (Beaumont et al. 2009; Peters et al. 2008; Sisson et al. 2007; Toni et al. 2009). The key idea is to decompose the problem of sampling from $\pi_\epsilon(\theta, x|y)$ into a series of simpler sub-problems. The algorithm begins at algorithmic time 0 sampling from $\pi_{\epsilon_0}(\theta, x|y)$, with ϵ_0 large, then simulating from an increasingly constrained sequence of target distributions $\pi_{\epsilon_n}(\theta, x|y)$, that is $\epsilon_n < \epsilon_{n-1}$, at subsequent algorithmic time steps $n \in \{1, \ldots, T\}$. In other words, the tolerance level is decreased until it reaches ϵ. These distributions are approximated by a large number of random samples which are propagated over time using a combination of importance sampling (IS) and resampling.

In the ABC context, Sisson et al. (2007) used the SMC samplers methodology developed in Del Moral et al. (2006) coupled with a partial rejection proposal. Some concerns have been raised about this algorithm in Beaumont et al. (2009). This issue is now resolved; see Sisson et al. (2009). In Beaumont et al. (2009), Toni et al. (2009), the authors have developed methods to improve the performance of the algorithm in Sisson et al. (2007) which can be interpreted as approximations to the "optimal" backward kernel in Del Moral et al. (2006, Sect. 2.4). This leads to algorithms of computational complexity that are quadratic in the number of particles and still requires a careful determination of the sequence of tolerance levels. If the tolerance levels decrease too fast then the algorithm can perform poorly whereas if they decrease too slowly then the algorithm will be too computationally intensive.

1.2 Contributions and organization of the article

In this article an original adaptive SMC method for ABC is developed. In comparison to previous work, our algorithm has the following features. It determines, in an automatic fashion, the sequence of tolerance levels to be used and the parameters of proposal distributions. Additionally it has a computational complexity that is linear in the number N of particles compared to quadratic for the algorithms proposed in Beaumont et al. (2009), Sisson et al. (2009), Toni et al. (2009). However, it is worth mentioning that all the previous SMC algorithms proposed for ABC and the one presented here require simulating a number of pseudo-observations proportional to N. Their computational complexity differ only in the calculation of some importance weights: it is quadratic for the algorithms in Beaumont et al. (2009), Sisson et al. (2009), Toni et al. (2009) and linear for the algorithm presented here. For challenging applications, simulating pseudo-observations is very expensive and this simulation step will dominate the computational cost of calculating the importance weights for N small enough. We also note that the authors in Beaumont et al. (2009) adapt the parameters of proposal densities but not the tolerance levels.

Some variations over our methodology have already appeared in Grelaud (2009, Chap. 5) and in Drovandi and Pettit (2011) where the adaptive schedule is determined based on the quantiles of the empirical distribution of the distance function associated to particles.

The rest of this article is organized as follows. In Sect. 2 we outline the SMC sampler approach of Del Moral et al. (2006) and discuss its application in an ABC context. In Sect. 3 an original adaptive SMC scheme for ABC is introduced. In Sect. 4 the performance of this algorithm is investigated on a toy example and a real data example also considered in Tanaka et al. (2006). Finally various extensions are explored in Sect. 5.

2 SMC samplers for ABC

2.1 SMC samplers

The SMC sampler methodology is a generic approach to approximate a sequence of probability distributions $\{\pi_n\}_{0\leq n\leq T}$ defined upon a common measurable space $(E, \mathcal{E})$ (Del Moral et al. 2006). These distributions are approximated by a collection of N random samples $\{Z_n^{(i)}\}_{i=1}^N$, termed particles. At time 0, the distribution π_0 is selected such that it is easy to approximate it using an importance distribution η_0. The particles $\{Z_{n-1}^{(i)}\}_{i=1}^N$ are then moved, from time $n-1$ to time n, by using a Markov kernel $K_n(z_{n-1}, z_n)$ which denotes the probability or probability density of moving from z_{n-1} to z_n. As the resulting marginal distribution at time n is typically not available, IS cannot be used directly, to correct for the discrepancy between this distribution and the target π_n. To bypass this problem, a sequence of extended probability distributions $\{\widetilde{\pi}_n\}_{1\leq n\leq T}$ are introduced on state-spaces of increasing dimension $(E^{n+1}, \mathcal{E}^{\otimes n+1})$ admitting $\{\pi_n\}_{1\leq n\leq T}$ as marginals; see Del Moral et al. (2006) for details. More specifically, the following sequence of auxiliary densities is introduced for $1 \leq n \leq T$

$$\widetilde{\pi}_n(z_{0:n}) = \pi_n(z_n) \prod_{j=0}^{n-1} L_j(z_{j+1}, z_j) \tag{2}$$

where we used the notation $z_{0:n} = (z_0, \ldots, z_n)$. The quantities $\{L_n\}_{0\leq n\leq T-1}$ are Markov kernels that act backward in time; that is $L_n(z_{n+1}, z_n)$ denotes the probability or probability density of moving from z_{n+1} to z_n. These kernels are consequently termed backward Markov kernels. It is thus clear from (2) that $\{\widetilde{\pi}_n\}$ admit $\{\pi_n\}$ as marginals for any $1 \leq n \leq T$.

We denote by $\delta_x(\cdot)$ the delta-Dirac measure located at x. The SMC sampler algorithm proceeds as follows.

- Step 0. Set $n \leftarrow 0$; for $i = 1, \ldots, N$, sample $Z_0^{(i)} \sim \eta_0$ and compute $W_0^{(i)} \propto \pi_0(Z_0^{(i)})/\eta_0(Z_0^{(i)})$, $\sum_{j=1}^N W_0^{(j)} = 1$.
- Step 1. If $\mathrm{ESS}(\{W_n^{(i)}\}) < N_T$ then resample N particles from

$$\widehat{\pi}_n(dz) = \sum_{i=1}^N W_n^{(i)} \delta_{Z_n^{(i)}}(dz) \tag{3}$$

  also denoted abusively $\{Z_n^{(i)}\}$ and set $W_n^{(i)} = \frac{1}{N}$. Set $n \leftarrow n+1$, if $n = T+1$ stop.
- Step 2. For $i = 1, \ldots, N$, sample $Z_n^{(i)} \sim K_n(Z_{n-1}^{(i)}, \cdot)$, compute

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(Z_n^{(i)}) L_{n-1}(Z_n^{(i)}, Z_{n-1}^{(i)})}{\pi_{n-1}(Z_{n-1}^{(i)}) K_n(Z_{n-1}^{(i)}, Z_n^{(i)})} \tag{4}$$

  and return to Step 1.

In this algorithm, the ESS denotes the so-called Effective Sample Size (Liu 2001, pp. 35–36) given at time n by

$$\mathrm{ESS}\left(\left\{W_n^{(i)}\right\}\right) = \left(\sum_{i=1}^N \left(W_n^{(i)}\right)^2\right)^{-1}. \tag{5}$$

The ESS criterion takes values between 1 and N and its interpretation is that inference based on the N weighted samples is approximately equivalent to inference based on $\mathrm{ESS}(\{W_n^{(i)}\})$ perfect samples from π_n. It allows us to assess the accuracy of the estimator and triggers a resampling step, implemented using the systematic resampling scheme (Kitagawa 1996), whenever its value is below a threshold N_T. Although the ESS is not a perfect measure, it does provide an idea of the behaviour of the algorithm; see Chopin (2002) for a discussion of its limitations and Beskos et al. (2011) for a theoretical study of the behaviour of this measure in high-dimensional settings.

2.2 Algorithm settings for ABC

In the context of ABC, we are interested in sampling from the sequence of target distributions $\{\pi_{\epsilon_n}(\theta|y)\}$ such that $\epsilon_0 > \epsilon_1 > \cdots > \epsilon_T = \epsilon$ where $\pi_{\epsilon_n}(\theta|y)$ is given by the marginal in θ of $\pi_{\epsilon_n}(\theta, x|y)$ in (1). As $\pi_{\epsilon_n}(\theta|y)$ is unknown, even up to a normalizing constant, SMC samplers techniques cannot be applied directly. Therefore, it is necessary to sample from the sequence of target distributions $\pi_n(z) = \pi_{\epsilon_n}(\theta, x|y)$. It should be noted that it is also possible to use SMC to sample from the sequence of targets

$$\pi_{\epsilon_n}(\theta, x_{1:M}|y) \propto \left(\frac{1}{M}\sum_{k=1}^M \mathbb{I}_{A_{\epsilon_n,y}}(x_k)\right)\left(\prod_{k=1}^M f(x_k|\theta)\right)\pi(\theta) \tag{6}$$

for any integer $M \in \mathbb{N}$ (Andrieu et al. 2008). This sequence admits the same marginal in θ for any M. Although it is more expensive to sample from $\pi_{\epsilon_n}(\theta, x_{1:M}|y)$ than $\pi_{\epsilon_n}(\theta, x|y)$ when $M > 1$, this has important advantages as discussed in Sect. 3.2 and illustrated in Sect. 4.

The performance of SMC samplers depends heavily upon the selection of an appropriate sequence $\{\epsilon_n\}$, the transition kernels $\{K_n\}$ and the auxiliary backward transition kernels $\{L_n\}$. Assume $\{\epsilon_n\}$ is fixed for the time being. Once K_n has been selected, it is established in Del Moral et al. (2006, Proposition 1) that the optimal choice for L_{n-1} in terms of minimizing the variance of the incremental weight (4) conditional upon $W_{n-1}^{(i)}$ is given by

$$L_{n-1}^{\mathrm{opt}}(z, z') = \frac{\pi_{n-1}(z')K_n(z', z)}{\int \pi_{n-1}(u)K_n(u, z)du} \tag{7}$$

which leads to

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(Z_n^{(i)})}{\int \pi_{n-1}(z_{n-1})K_n(z_{n-1}, Z_n^{(i)})dz_{n-1}}. \tag{8}$$

In the ABC context, it is unfortunately impossible to compute the denominator of L_{n-1}^{opt} and the associated weight (8) analytically. A solution proposed in Del Moral et al. (2006, Sect. 2.4) consists of approximating (8) numerically using the SMC approximation $\widehat{\pi}_{n-1}(dz)$ of $\pi_{n-1}(dz)$ which leads to

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(Z_n^{(i)})}{\sum_{j=1}^N W_{n-1}^{(j)} K_n(Z_{n-1}^{(j)}, Z_n^{(i)})}. \tag{9}$$

This is the approach followed in Beaumont et al. (2009), Toni et al. (2009), Sisson et al. (2009). The cost of computing each incremental weight is $\mathcal{O}(N)$ for each particle so the overall computational complexity is $\mathcal{O}(N^2)$.

We propose an alternative approach where we use an MCMC kernel of invariant distribution π_n for K_n as recommended in Del Moral et al. (2006). This allows us to use for the backward Markov kernel an approximation of the optimal backward kernel (7) given by the reversal kernel

$$L_{n-1}(z, z') = \frac{\pi_n(z')K_n(z', z)}{\pi_n(z)}. \tag{10}$$

Note that this choice of backward kernels was implicitly made in Chopin (2002), Gilks and Berzuini (2001) and explicitly in related algorithms Neal (2001) where no resampling step is used. Selecting K_n as an MCMC kernel of invariant density π_n and L_{n-1} as the reversal backward kernel (4) leads to a weight update (4)

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\pi_n(Z_{n-1}^{(i)})}{\pi_{n-1}(Z_{n-1}^{(i)})} \propto W_{n-1}^{(i)} \frac{\sum_{k=1}^M \mathbb{I}_{A_{\epsilon_n,y}}(X_{k,n-1}^{(i)})}{\sum_{k=1}^M \mathbb{I}_{A_{\epsilon_{n-1},y}}(X_{k,n-1}^{(i)})}. \tag{11}$$

The cost of computing each incremental weight is $\mathcal{O}(1)$ for each particle so the overall computational complexity is $\mathcal{O}(N)$. Although the variance of the weight (11) is necessarily higher than the variance of the weight (8), these variances will be fairly similar in scenarios where $\pi_n(z)$ does not differ significantly from $\pi_{n-1}(z)$; that is if the difference $\epsilon_{n-1} - \epsilon_n$ is not too large.

In this very specific case, it is clear that if $M = 1$, $\eta_0 = \pi_0$ then (11) yields either $W_n^{(i)} \propto 1$ or $W_n^{(i)} = 0$ and thus $\mathrm{ESS}(\{W_n^{(i)}\})$ is directly proportional to the number of "alive" particles at time $n-1$, that is to the number of particles with strictly positive weights $W_n^{(i)}$. It is also worth noticing that in this case $W_n^{(i)}$ is independent of $\{Z_n^{(i)}\}$. This allows us to swap the order of the sampling and resampling steps; see Del Moral et al. (2006, Remark 1, p. 418). We will also exploit this property in the next section to obtain an adaptive method to determine the tolerance levels.

As $\epsilon_n < \epsilon_{n-1}$, it is typically the case that there is a non-null proportion of particles that have zero weights. This emphasizes the importance of selecting an appropriate sequence of tolerance levels. Indeed, if this sequence decreases too slowly then, with high probability, $W_n^{(i)} = W_{n-1}^{(i)}$ and the algorithm will move too slowly towards the target $\pi_\epsilon(\theta, x_{1:M}|y)$. Conversely, if the $\{\epsilon_n\}$ decrease too quickly, then with high probability, all the weights $\{W_n^{(i)}\}$ can equal zero; hence the SMC sampler approximation would have collapsed. To prevent such a collapse, the algorithms in Peters et al. (2008) and Sisson et al. (2007) targeting $\pi_{\epsilon_n}(\theta, x|y)$ generate particles $Z_n^{(i)} = (\theta_n^{(i)}, X_n^{(i)})$ in regions such that $\rho(\eta(X_n^{(i)}), \eta(y)) < \epsilon_n$. The kernel K_n they use is not an MCMC kernel of invariant distribution $\pi_{\epsilon_n}(\theta, x|y)$ and this makes the selection of an associated backward kernel, to ensure that the variance of $W_n^{(i)}$ remains reasonable, more difficult.

3 An adaptive SMC sampler for ABC

In this section, a simple adaptive SMC algorithm is proposed which: relies on MCMC kernels to move the particles around the space; admits a computational complexity that is linear in N; automatically determines the sequence of tolerance levels so as to prevent the collapse of the SMC approximation.

3.1 An adaptive schedule for tolerance levels

Our method for selecting the tolerance levels $\{\epsilon_n\}$ adaptively is based on the key remark that the expression (11) for the weights $\{W_n^{(i)}\}$ does not depend on $\{Z_n^{(i)}\} = \{(\theta_n^{(i)}, X_{1:M,n}^{(i)})\}$; see Del Moral et al. (2006, Sect. 3.5) for a detailed discussion. We aim to control the ESS over iterations by selecting the tolerance level ϵ_n such that

$$\mathrm{ESS}\left(\left\{W_n^{(i)}\right\}, \epsilon_n\right) = \alpha \mathrm{ESS}\left(\left\{W_{n-1}^{(i)}\right\}, \epsilon_{n-1}\right) \tag{12}$$

for $\alpha \in (0, 1)$ where $W_n^{(i)}$ given in (11) depends on ϵ_n. We have emphasized here notationally that the ESS at time n is a function of ϵ_n. As $\mathrm{ESS}(\{W_n^{(i)}\}, \epsilon_n)$ is not a continuous function of ϵ_n for a finite number N of particles, it is typically impossible to find a value of ϵ_n which exactly solves (12) but bisection can be used to find an approximate solution. This construction prevents the collapse of the SMC approximation. The parameter α is a "quality" index for the resulting SMC approximation of the target. If $\alpha \approx 1$ then we will move slowly towards the target but the SMC approximation will be very good. However, if $\alpha \approx 0$ then we can move very quickly towards the target but the resulting SMC approximation will be unreliable.

In this context, it might also appear sensible to determine the tolerance schedule by solving approximately for ϵ_n at time n

$$\mathrm{PA}\left(\left\{W_n^{(i)}\right\}, \epsilon_n\right) = \alpha \mathrm{PA}\left(\left\{W_{n-1}^{(i)}\right\}, \epsilon_{n-1}\right)$$

where PA, the proportion of alive particles, is given by

$$\mathrm{PA}\left(\left\{W_n^{(i)}\right\}, \epsilon_n\right) = \frac{\sum_{i=1}^N \mathbb{I}_{(0,\infty)}(W_n^{(i)})}{N}. \tag{13}$$

This criterion is equal to ESS/N when $M = 1$ and $\eta_0 = \pi_0$ and, for $M > 1$, it has the attractive property that it is necessarily an increasing function of ϵ_n contrary to the ESS. The PA criterion might appear as a sensible measure of "quality" of our SMC approximation. However, for large values of M, this intuition is ill-founded and the PA criterion should not be used for determining the tolerance levels and/or trigger the resampling steps. Indeed, for the PA to be large, we only need to have a large proportion of particles having at least one components of $X_{1:M}$ in $A_{\epsilon_n, y}$. Clearly if M is reasonably large, this will happen with high probability. We can formalize precisely this intuition by looking at the asymptotic form of the PA criterion. As $N \to \infty$, it is easy to check that $\lim_{N\to\infty} \mathrm{PA}(\{W_n^{(i)}\}, \epsilon_n) = \mathrm{PA}^*$ almost surely where, for any $M \geq 1$, we have

$$\mathrm{PA}^* = 1 - \left(1 - \pi_{\epsilon_{n-1}}(A_{\epsilon_n, y} \times \Theta)\right)^M.$$

Hence we obtain

$$\lim_{M\to\infty} \mathrm{PA}^* = 1$$

whenever $\pi_{\epsilon_{n-1}}(A_{\epsilon_n, y} \times \Theta) > 0$. This demonstrates that the PA should not be used as a criterion to assess the quality of the particle approximation as it converges to 1 geometrically fast as M increases for any $\epsilon_n < \epsilon_{n-1}$.

3.2 Adaptive MCMC kernels

At each time our algorithm applies an MCMC kernel $K_n((\theta, x_{1:M}), (\theta', x'_{1:M}))$ of invariant density $\pi_{\epsilon_n}(\theta, x_{1:M}|y)$. A slightly modified version of the ABC-MCMC scheme of Majoram et al. (2003) can be used to achieve this. Given $Z = (\theta, X_{1:M})$, with $\sum_{k=1}^{M} \mathbb{I}_{A_{\epsilon_n,y}}(X_k) \geq 1$, then $(\theta^*, X^*_{1:M})$ are generated according to a proposal

$$q_n(\theta, \theta^*) \prod_{k=1}^{M} f(x_k^*|\theta^*).$$

This candidate is accepted with acceptance probability given by the Metropolis-Hastings (MH) ratio

$$1 \wedge \frac{\pi_{\epsilon_n}(\theta^*, X^*_{1:M}|y)}{\pi_{\epsilon_n}(\theta, X_{1:M}|y)} \frac{q_n(\theta^*, \theta)}{q_n(\theta, \theta^*)} \prod_{k=1}^{M} \frac{f(X_k|\theta)}{f(X_k^*|\theta^*)}$$

$$= 1 \wedge \frac{\sum_{k=1}^{M} \mathbb{I}_{A_{\epsilon_n,y}}(X_k^*)}{\sum_{k=1}^{M} \mathbb{I}_{A_{\epsilon_n,y}}(X_k)} \frac{q_n(\theta^*, \theta)}{q_n(\theta, \theta^*)} \frac{\pi(\theta^*)}{\pi(\theta)}.$$

This expression outlines the benefit of sampling M variables. We reduce the variance of the MH acceptance ratio as M increases. In the limiting case, i.e. $M \to \infty$, we have $\frac{1}{M} \sum_{k=1}^{M} \mathbb{I}_{A_{\epsilon_n,y}}(X_k) \to \int f(x|\theta) \mathbb{I}_{A_{\epsilon_n,y}}(x) dx$ and our algorithm is similar to a "marginal" MCMC algorithm where X has been integrated out; see Andrieu et al. (2008) for further discussion.

In this framework, we can adaptively determine the parameters of the proposal $q_n(\theta, \theta^*)$ based on the previous approximation of the target π_{n-1}. Contrary to adaptive MCMC methods (Andrieu and Johanes 2008), no stringent condition is required to ensure the validity of the algorithm as the MCMC kernel is only used to build a sensible importance distribution. As in Beaumont et al. (2009), the variance of θ under $\pi_{\epsilon_{n-1}}(\theta|y)$ can be approximated at time $n-1$ using the SMC approximation and used to determine the variance of the proposal density of the MCMC algorithm at time n, i.e. through a normal random walk proposal. Many other possible adaptation schemes are also possible. For example, when using random walk MH, in Sect. 4.2, the proposal variances of the MCMC kernels were adaptively computed at every time step of the SMC using a stochastic approximation like scheme based on expected acceptance probabilities; see Andrieu and Johanes (2008).

3.3 An adaptive SMC method

To summarize, our adaptive SMC method for ABC proceeds as follows. We use $\epsilon_0 = \infty$ so that $W_0^{(i)} = \frac{1}{N}$ and $\mathrm{ESS}(\{W_0^{(i)}\}, \epsilon_0) = N$.

- Step 0. Set $n = 0$; for $i = 1, \ldots, N$, sample $\theta_0^{(i)} \sim \pi(\cdot)$ and $X_{k,0}^{(i)} \sim f(\cdot|\theta_0^{(i)})$ where $k = 1, \ldots, M$.
- Step 1. Set $n \leftarrow n+1$, if $\epsilon_{n-1} = \epsilon$ stop, otherwise determine ϵ_n by solving $\mathrm{ESS}(\{W_n^{(i)}\}, \epsilon_n) = \alpha \mathrm{ESS}(\{W_{n-1}^{(i)}\}, \epsilon_{n-1})$ where

$$W_n^{(i)} \propto W_{n-1}^{(i)} \frac{\sum_{k=1}^{M} \mathbb{I}_{A_{\epsilon_n,y}}(X_{k,n-1}^{(i)})}{\sum_{k=1}^{M} \mathbb{I}_{A_{\epsilon_{n-1},y}}(X_{k,n-1}^{(i)})}. \tag{14}$$

  If $\epsilon_n < \epsilon$ then set $\epsilon_n \leftarrow \epsilon$.
- Step 2. If $\mathrm{ESS}(\{W_n^{(i)}\}) < N_T$ then resample N particles from

$$\widehat{\pi}_{\epsilon_n}(d(\theta, x_{1:M})|y) = \sum_{i=1}^{N} W_n^{(i)} \delta_{(\theta_{n-1}^{(i)}, X_{1:M,n-1}^{(i)})}(d(\theta, x_{1:M})) \tag{15}$$

  also denoted abusively $\{\theta_{n-1}^{(i)}, X_{1:M,n-1}^{(i)}\}$ and set $W_n^{(i)} \leftarrow \frac{1}{N}$.
- Step 3. For $i = 1, \ldots, N$, sample $(\theta_n^{(i)}, X_{1:M,n}^{(i)}) \sim K_n((\theta_{n-1}^{(i)}, X_{1:M,n-1}^{(i)}), \cdot)$ if $W_n^{(i)} > 0$ and return to Step 1.

Note that in this context, $\pi_{\epsilon_n}(\theta, x_{1:M}|y)$ can be approximated by both the weighted measures associated to $\{W_n^{(i)}, (\theta_{n-1}^{(i)}, X_{1:M,n-1}^{(i)})\}$ as in (15) or using $\{W_n^{(i)}, (\theta_n^{(i)}, X_{1:M,n}^{(i)})\}$. The approximations from all the target distributions may be combined to obtain lower variance estimates as discussed in Gramacy et al. (2010).

3.4 Cautionary remarks

Whilst our adaptive procedure can help to move particles towards the high posterior probability regions of the parameter space for reasonably large values of ϵ, we do not claim it can solve all the computational problems associated to ABC. It will perform poorly whenever the Monte Carlo estimates of $\int f(x|\theta) \mathbb{I}_{A_{\epsilon,y}}(x) dx$ implicitly computed by the algorithm have a large relative variance; that is when ϵ is small and/or the distribution $f(x|\theta)$ is diffuse. Indeed in this case the MCMC kernel used to explore the space will mix very poorly and this will result in potentially high variance SMC estimates. In our opinion, one of the key benefits of the methodology proposed here is that it allows us to determine on-the-fly the sequence of tolerance levels $\{\epsilon_n\}$ and to identify through the ESS whenever the tolerance level has reached too low a value. When the tolerance levels determined adaptively start decreasing very slowly, it is indicative of the fact that our algorithm cannot move efficiently the particles around the space anymore and in a sense determines the value of ϵ we should adopt in practice. In our application to epidemiology presented in Sect. 4.2, we use the following heuristic: we stop the algorithm when the MCMC drop below 1.5% and this determines the value of ϵ.

4 Application

The code for the examples are available at http://www2.imperial.ac.uk/~aj2/abc_new1.zip.

4.1 A toy example

4.1.1 Target and implementation details

We begin with the toy example introduced in Beaumont et al. (2009), Sisson et al. (2007). The model is of the form

$$\theta \sim \mathcal{U}_{[-10,10]},$$
$$f(x|\theta) = 0.5\phi(x;\theta,1) + 0.5\phi(x;\theta,1/100)$$

where $\mathcal{U}_{[a,b]}$ denotes the uniform distribution on the interval $[a,b]$ and $\phi(u;m,\sigma^2)$ the one-dimensional normal density of mean m and variance σ^2, evaluated at u. The posterior distribution associated to the observation $y=0$ (recall observed data is y, pseudo data is x) is

$$\pi(\theta|y) \propto \big(\phi(\theta;0,1) + \phi(\theta;0,1/100)\big)\mathbb{I}_{[-10,10]}(\theta).$$

An ABC strategy is used to estimate $\pi(\theta|y)$, with $\eta(x)=x$ and $\rho(x,y)=|x-y|=|x|$ (i.e. the $\mathbb{L}_1$ distance). In this case, using (1), we obtain

$$\pi_\epsilon(\theta|y) \propto \big(\Phi(\epsilon-\theta) - \Phi\big(-(\epsilon+\theta)\big) + \Phi\big(10(\epsilon-\theta)\big) - \Phi\big(-10(\epsilon+\theta)\big)\big)\mathbb{I}_{[-10,10]}(\theta)$$

where $\Phi(u)$ is the cumulative distribution function of the standard normal (Beaumont et al. 2009). For $\epsilon = 0.025$, it is shown in Beaumont et al. (2009) that $\pi_\epsilon(\theta|y)$ is visually indistinguishable from $\pi(\theta|y)$.

The adaptive SMC algorithm described is run using a normal random walk MH kernel for θ of variance given, at time $n \geq 1$, by twice the empirical variance of the $\{\theta_{n-1}^{(i)}\}$ as in Beaumont et al. (2009). We set $N_T = N/2$ and $\epsilon = 0.01$.

4.1.2 Varying M and N

Our experiments use $N \in \{1000, 10000, 100000\}$ particles, $M=1$ and the adaptive SMC algorithm is run for $\alpha \in \{0.9, 0.95, 0.99\}$. In Table 1, the CPU times are given for this adaptive SMC algorithm averaged over 50 realisations using a PC Intel 3.33 GHz.

In Fig. 1 the histograms of the samples obtained by the adaptive SMC method are given. As expected, the results improve as both α and N increase. Note that the particles are resampled after the last time step.

For $N=10000$ and $\alpha = 0.95$ we now investigate the influence of $M \in \{1,5,50\}$ on the performance of the algorithm. In Fig. 2 the average, over the alive particles, acceptance rate of the MH step and the sequence of tolerance levels $\{\epsilon_n\}$ as a function of the time index n are displayed. Note that M has an influence on the number of intermediate distributions required to reach the target. The higher M the smaller this number is and, as expected, the higher M the higher the average acceptance rate for a fixed ϵ. In general, the value of M that is necessary to stabilize the results depends on $f(x|\theta)$; the more diffuse $f(x|\theta)$ is (in x), the higher M should be.

Table 1 Averaged CPU times in seconds for various values of N and α

N	α		
	0.90	0.95	0.99
1000	0.08	0.16	0.77
10000	0.87	1.73	8.50
100000	8.58	17.39	88.39

Ideally, we would like all the M, N and α to be large. With a fixed computational budget, a tradeoff has to be found. It appears unfortunately difficult to give useful general guidelines how to select those parameters as it is highly case dependent.

4.1.3 Adaptive versus deterministic schedules

The relative benefits of the adaptive to deterministic tolerance schedule is now investigated. Our approach requires setting α and ϵ whereas a deterministic schedule requires specifying all the tolerance levels $\{\epsilon_n\}$. A conservative approach consists of adopting a schedule $\{\epsilon_n\}$ which is assumed to decrease slowly so as to prevent a collapse of the algorithm. We compare the following two approaches. First we select a deterministic tolerance schedule, $\epsilon_1 = 10$ and then falling linearly by 0.1 until $\epsilon_n < 0.01$—the last ϵ is then taken as 0.01. A linear decreasing schedule was also adopted in Beaumont et al. (2009, Sect. 2). It has been shown recently that such linear schedules enjoy good theoretical properties in high-dimensional scenarios (Beskos et al. 2011) and so one may suspect it can also produce reasonable results in simpler applications. Second, we determine our schedule adaptively using our adaptive procedure with $\alpha = 0.9$ and $\epsilon = 0.01$. In both scenarios $M=1$ and $N=1000$. Both settings are what we might call conservative, especially given the toy nature of the example. We calculated, across 50 repeats, the $\mathbb{L}_1$ distance between the true and estimated second moment. It was (standard deviation in brackets) 0.19 (0.025) for the adaptive schedule and 0.28 (0.240) for the deterministic schedule.

We display the evolution of the ESS for the two schedules in Fig. 3, as well the sequence of tolerance levels $\{\epsilon_n\}$ computed by the adaptive method. The plots show that, whilst intuitively reasonable, the deterministic schedule leads to

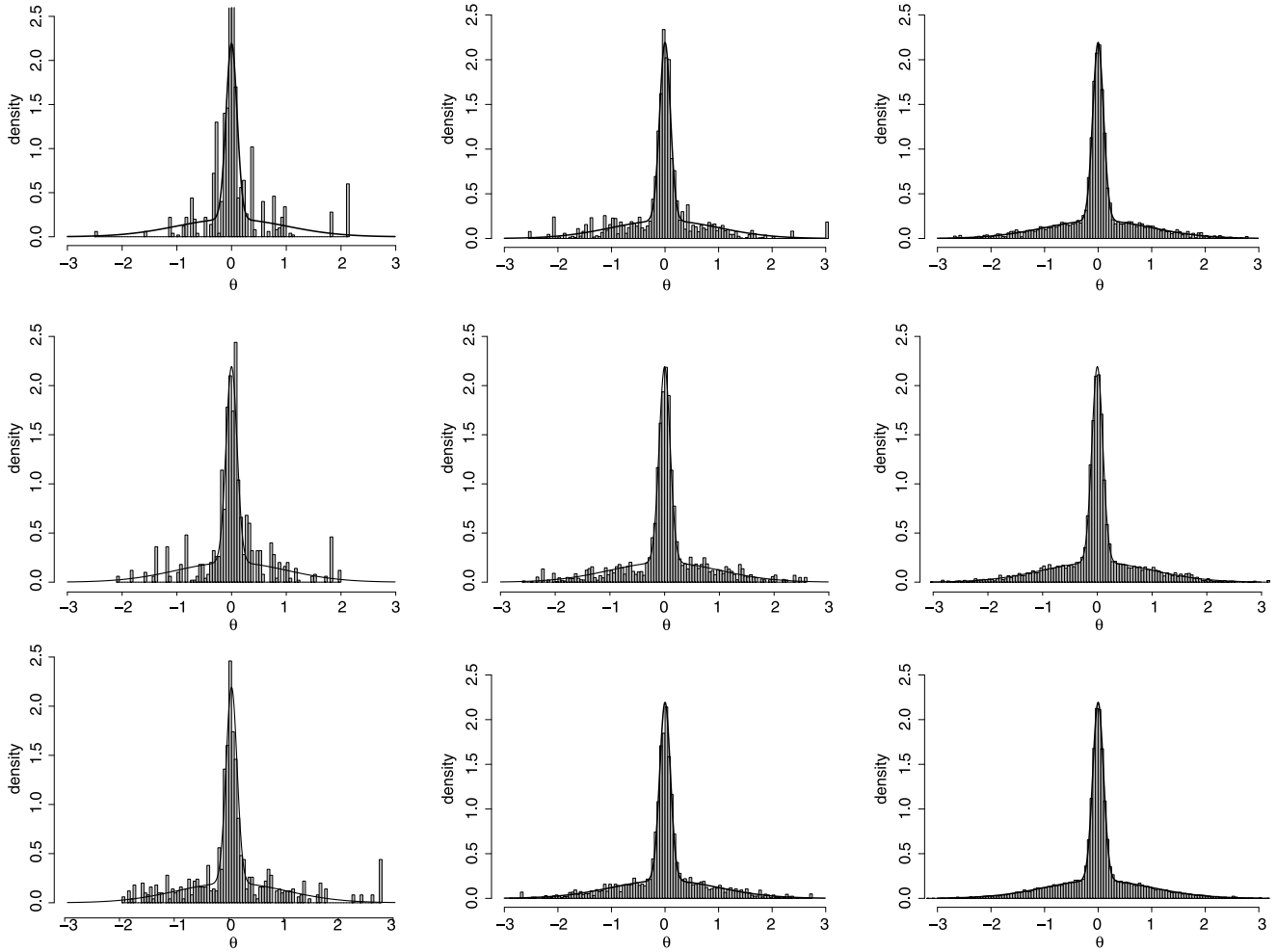
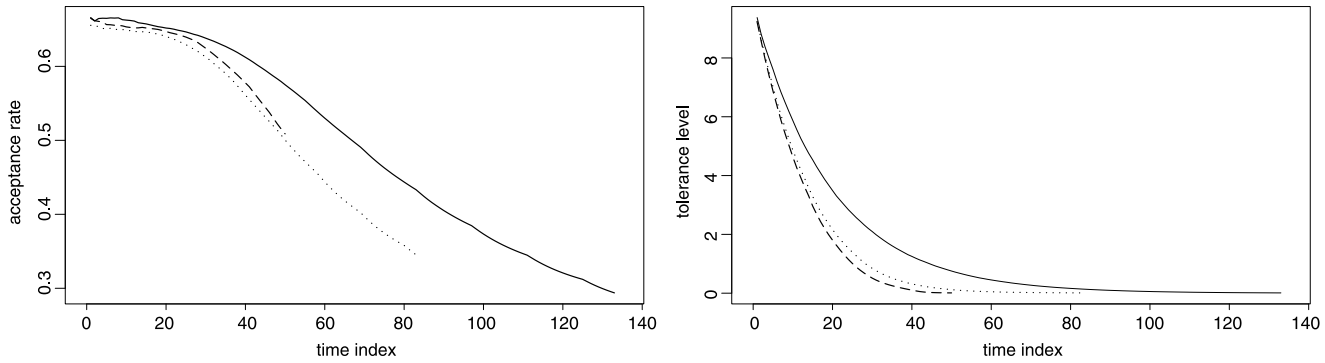


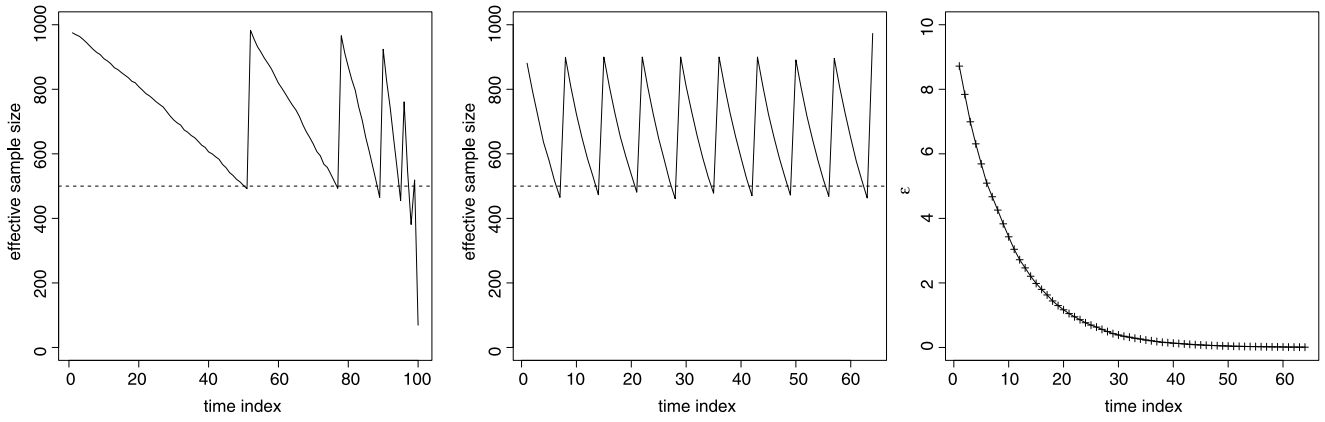
Fig. 1 Histograms of the samples obtained through SMC. Each row corresponds to a different α (from *top* to *bottom* $\alpha = 0.9,\ 0.95$ and 0.99), each column corresponds to a different N (from *left* to *right* $N = 1000,\ 10000$ and 100000). The true target density $\pi_\epsilon(\theta|y)$ is the *full line*



(a) Acceptance rate (b) Tolerance level

Fig. 2 Average acceptance rate of the MH step (*left*) and sequence of tolerance levels $\{\epsilon_n\}$ (*right*) as a function of n for $M = 1$ (*solid*), 5 (*dots*), 10 and 50 (*dashes*)

the ESS decreasing drastically close to ϵ (Fig. 3(a)). Conversely, the adaptive procedure has a fairly consistent resampling rate, and Monte Carlo estimates are likely to be far more reliable on the basis of the given output (Fig. 3(b)). Note that the sequence of tolerance levels computed by our adaptive algorithm exhibits an exponential decrease which is quite intuitive given its construction. It could be argued that a deterministic schedule decreasing exponentially



(a) Deterministic Schedule (b) Adaptive Schedule (c) ϵ_n

Fig. 3 *Left*: ESS obtained using an SMC sampler run with a deterministic schedule, starting at $\epsilon_1 = 10$ and falling by 0.1 at each step. *Middle*: ESS obtained using the adaptive SMC sampler using $\alpha = 0.9$. *Right*: Tolerance levels computed using the adaptive SMC sampler. Both deterministic and adaptive schemes were run for similar CPU times with $M = 1$

could provide better results. However, it would remain difficult to specify the parameters of this exponential function; this problem is circumvented by our algorithm.

4.1.4 Comparison with population Monte Carlo

Finally, we performed a comparison with the $\mathcal{O}(N^2)$ Population Monte Carlo (PMC) method of Beaumont et al. (2009). The PMC algorithm settings are as in their Fig. 1 (i.e. $M = 1$ and $\epsilon_1 = 2$ falling 0.5 and ending at $\epsilon = 0.01$), and is run for $N \in \{1000, 2000, 3000, 4000, 5000\}$. To compare, our adaptive SMC method is run with $\alpha = 0.95$ and for the same CPU times as the corresponding PMC algorithm. This leads to $N \in \{3400, 13000, 28000, 50000, 78000\}$. The results are given in Table 2. This table reveals that, for a fixed computational complexity, the SMC algorithm significantly outperforms the PMC algorithm. However we acknowledge that this is a favourable scenario for our algorithm as the cost of simulating pseudo-observations is low for this toy example. For challenging applications, simulating pseudo-observations is typically very expensive and we cannot expect such gains.

4.2 Intractable birth-death-mutation process

4.2.1 Model and data

We consider a birth-death-mutation model as detailed in Tanaka et al. (2006). The model is designed to capture the dynamics of the growth in the number of infectious diseases over a period of time. The birth (rate φ), corresponds to new infections, the death (rate τ) to death or recovery of a host and the mutation (rate ξ) is assumed to always give rise to genotypes that have not appeared before. We denote by $X_i(t)$ the number of cases of genotype i and $G(t)$ the number of distinct genotypes that have existed in the population up to and including time t. A full mathematical presentation of the transition probabilities of the birth-death-mutation model is detailed in Tanaka et al. (2006) and the references therein but from an ABC viewpoint we only require being able to simulate pseudo-observations from the likelihood. This can be achieved as follows:

- At time $t = 1$, the process begins with one genotype and a single individual of that type: $X_1(1) = 1$, $G(1) = 1$.
- Set $t \leftarrow t + 1$. If $t > 10000$ then stop, otherwise draw an event of type birth with probability $\varphi/(\varphi + \tau + \xi)$, death with probability $\tau/(\varphi + \tau + \xi)$ or mutation otherwise. A genotype i is picked with probability $X_i(t-1)/\sum_{j=1}^{G(t-1)} X_j(t-1)$. If the event is a birth, then $X_i(t) = X_i(t-1) + 1$, $G(t) = G(t-1)$. If the event is a death, $X_i(t) = X_i(t-1) - 1$, $G(t) = G(t-1)$. If the event is a mutation, $G(t) = G(t-1) + 1$, $X_i(t) = X_i(t-1) - 1$ and $X_{G(t)}(t) = 1$; i.e. a new genotype with a single individual is created.
- Sample n individuals without replacement from the population of 10000 individuals. We obtain clusters of size n_i with $\sum_{i=1}^{g} n_i = n$ where g is the number of distinct genotypes in the sample. Note that necessarily we have $g \leq G(10000)$.

The observed data are from Small et al. (1994) (San-Francisco Tuberculosis data) and constitute 326 different genotypes of $n = 473$ isolates; the data are summarized as

$$30^1 \quad 23^1 \quad 15^1 \quad 10^1 \quad 8^1 \quad 5^2 \quad 4^4 \quad 3^{13} \quad 2^{20} \quad 1^{282}$$

where m^k indicates there were k clusters of size m.

In our implementation of ABC the summary statistics for data are as in Tanaka et al. (2006):

Table 2 $\mathbb{L}_1$-Distance between the true and estimated second moment. The algorithms were run 50 times for similar CPU times and the average distance, with standard deviation in brackets, across the runs is reported

Particles PMC	1000	2000	3000	4000	5000
PMC	0.128 (0.107)	0.107 (0.161)	0.105 (0.118)	0.076 (0.059)	0.071 (0.050)
SMC	0.089 (0.054)	0.042 (0.028)	0.034 (0.023)	0.025 (0.016)	0.022 (0.017)

$$\eta = (\eta_1, \eta_2) = \left(g, 1 - \sum_{i=1}^{g} (n_i/n)^2 \right)$$

and we also use the same distance metric and priors as in Tanaka et al. (2006) $\rho(\eta, \overline{\eta}) = \frac{1}{n}|\eta_1 - \overline{\eta}_1| + |\eta_2 - \overline{\eta}_2|$. As noticed by Fearnhead and Prangle (2010), the parameters (φ, τ, ξ) are not likelihood-identified as any rescaled version of the parameters $(\alpha\varphi, \alpha\tau, \alpha\xi)$ for $\alpha > 0$ has the same likelihood. Hence it would be appropriate to set one of the parameters to a given value. However, to compare our results with those of Tanaka et al. (2006), we follow their approach by setting the following priors: $\varphi \sim \mathcal{G}a(1, 0.1)$ ($\mathcal{G}a(a, b)$ is the Gamma distribution of mean b/a), $\xi \sim \mathcal{TN}(0.198, 0.06735^2)$ where $\mathcal{TN}$ is the truncated normal restricted to $\mathbb{R}^+$ and $p(\tau|\varphi) \propto \mathbb{I}_{[0,\varphi)}(\tau)$; i.e. ξ is assigned a very informative prior which deals partially with the identifiability issue.

4.2.2 Simulation results

The objective of this example is to illustrate the ability of our SMC method to be able to sample from the ABC posterior for small values of ϵ. The SMC sampler is run with $M = 15$, $\alpha = 0.9$, $\epsilon = 0.00045$, $N = 1000$ and the resampling threshold is set to $N_T = 500$. The choice of $M = 15$ was made for robustness of results across multiple runs. The MH kernels used within the SMC algorithm were based on normal random walks after reparameterizing onto the real line. We determine the final valueϵ by stopping the algorithm when the MCMC acceptance rate for one parameter drops below 1.5%; this yields $\epsilon = 0.00045$. For comparison, we run the MCMC algorithm of Bortot et al. (2009). The artificial prior on ϵ is selected as $p(\epsilon) \propto \mathbb{I}_{[0.00045,10]}(\epsilon)e^{-2000\epsilon}$ to 'push' ϵ towards small values. The MCMC proposal standard deviations for the normal random walks being (0.05, 0.05, 0.7) for (φ, τ, ξ) respectively reparameterized over the real line. The MCMC algorithm is a Metropolis-within-Gibbs algorithm which sweeps over updating ϵ and then each of the (φ, τ, ξ) in turn.

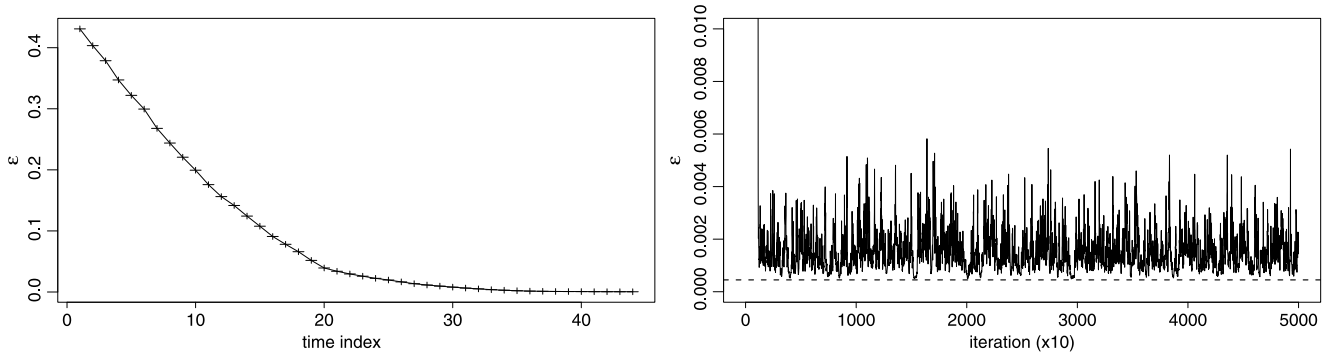
The CPU time for the SMC sampler was 147100 seconds for these simulation settings (run on a PC Intel 3.33 GHz). For approximately the same amount of CPU time, we could run the MCMC algorithm for 50000 iterations, taking every 10th sample. The results are displayed in Figs. 4, 6 and 7.

In Fig. 4 the evolution of $\{\epsilon_n\}$ for both methods can be seen. Figure 4(a) displays the decay of $\{\epsilon_n\}$ for the SMC sampler. As the MCMC kernels begin to mix slowly, the rate of decline of $\{\epsilon_n\}$ falls. However the SMC algorithm operates reasonably well along the path to ϵ (Fig. 6(a)) as the ESS never drops below 477. Figure 4(b) shows the exploration of the MCMC on the ϵ space. The algorithm is able to move slowly around the support, but struggles to consistently visit low values of ϵ (i.e. in the range (0.00045,0.001))—only 1213 samples were obtained from the 5000 taken (i.e. roughly only 20% of the samples were within this range). Similar behaviour was observed on different runs.

The MH acceptance rates for the MCMC algorithm were 0.55 (ϵ), 0.51 (φ), 0.55 (τ), and 0.36 (ξ). Slow mixing on all the parameters was observed (see Fig. 5) with the exception of ξ, which has a posterior very similar to its prior as observed in Tanaka et al. (2006).

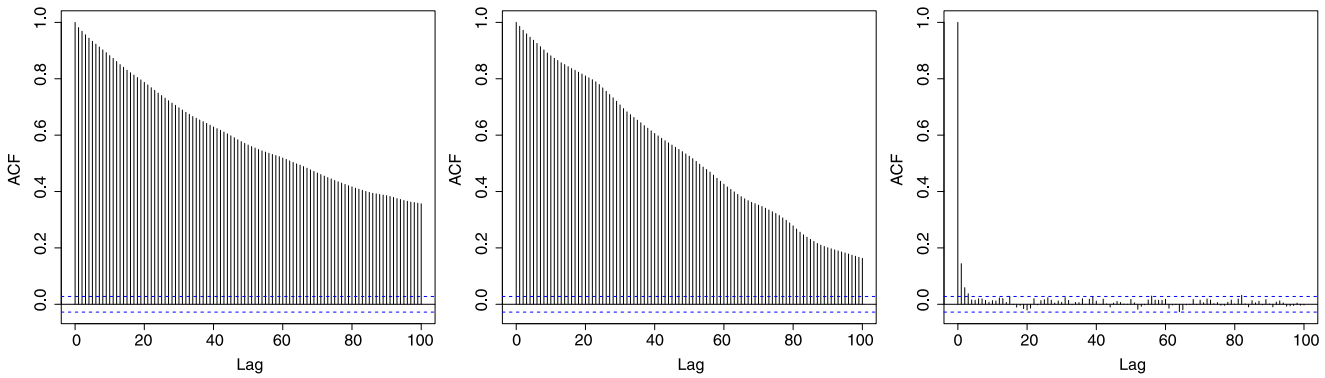
We compared the MCMC and SMC algorithms in terms of posterior means and standard deviations; see Table 3. They target different distributions as ϵ is made random in the MCMC sampler whereas $\epsilon = 0.00045$ for the SMC sampler so a direct comparison is obviously difficult. So as to perform a fairer comparison, we computed the posterior means and standard deviations for the MCMC algorithm conditional upon $\epsilon \in (0.00045, 0.001)$. We observed that the estimates of the death τ and mutation ξ rates do not differ significantly. However in terms of birth rate φ, there is much more discrepancy. To assess the robustness of the SMC algorithm, we additionally run it for $N = 10000$ particles and obtained estimates relatively similar to those obtained using $N = 1000$ particles.

In Fig. 7 we display some of the results we have obtained. These results differ slightly from those in Tanaka et al. (2006); in (a) the net transmission rate $\varphi - \tau$ has a mode at higher values and is far more positively skewed whereas in (b) it appears that the doubling rate $\log(2)/(\varphi - \tau)$ has a mode at lower values. Similarly in (c) and (d) the joint posteriors exhibit shapes different from the results in Tanaka et al. (2006). It is remarked, however, that the strong positive correlation between τ and φ observed in Tanaka et al. (2006) is also observed here. Note, however, that it is not possible to perform a direct comparison between our results and those in Tanaka et al. (2006) as we use a much smaller final value of ϵ than in Tanaka et al. (2006).



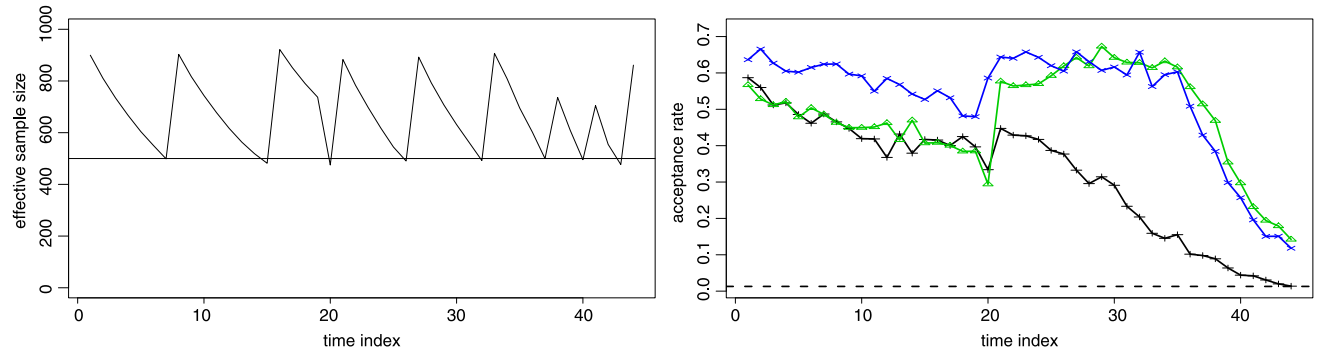
(a) SMC (b) MCMC

Fig. 4 (**a**) Tolerance levels $\{\epsilon_n\}$ computed by an SMC sampler with 1000 particles. (**b**) MCMC samples for ϵ; the dotted line is 0.00045



(a) φ (b) τ (c) ξ

Fig. 5 Autocorrelation plots for the MCMC method of Bortot et al. (2009). 50000 iterations were simulated and every 10th sample taken

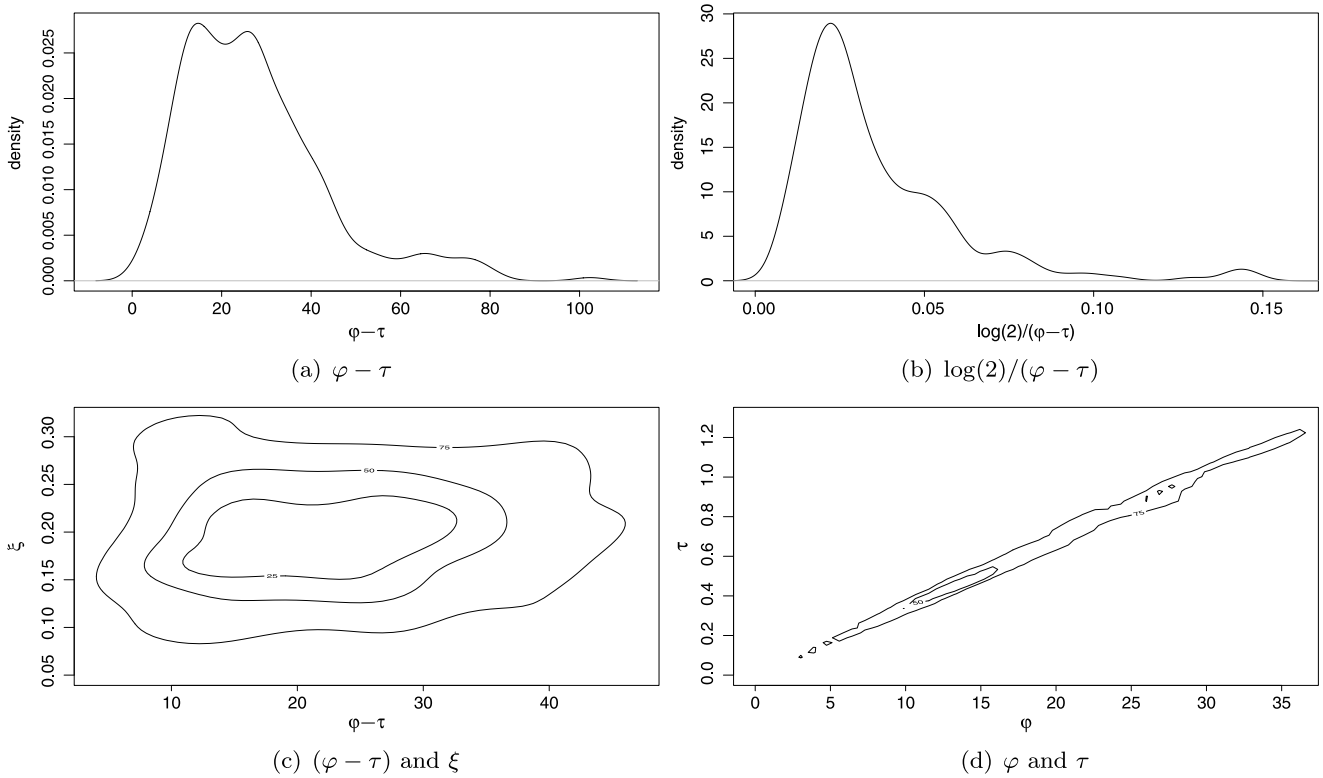


(a) Effective Sample Size (b) Acceptance Rates

Fig. 6 (Color online) Performance of the SMC sampler. (**a**) Displays the ESS against the time index, and (**b**) the acceptance rates of the MCMC moves for the 3 parameters φ (*black*) and τ (*green*). In (**b**) the *dotted line* is 0.013

5 Discussion

In this paper we have presented an adaptive SMC algorithm for ABC. Our approach has a computational cost that is linear in the number of samples and is able to adaptively calculate the tolerance levels in a sensible manner. Although little user input is required, it is difficult to *a priori* decide what the best combination of parameters N, M and α is as it is highly model dependent. We have demonstrated the performance of our algorithm on a toy example and a more complex application from genetics (Tanaka et al. 2006). It would be worth investigating the experimental performance of this algorithm on more challenging problems.



(a) $\varphi - \tau$ (b) $\log(2)/(\varphi - \tau)$

(c) $(\varphi - \tau)$ and ξ (d) φ and τ

Fig. 7 Inference from the SMC sampler

Table 3 Posterior means and standard deviations. The first number is the mean, the second is the standard deviation and the third in brackets is the standard error across 20 runs. The MCMC draws inference conditional upon $\epsilon \in (0.00045, 0.001)$

Parameter	SMC ($N = 1000$)	SMC ($N = 10000$)	MCMC
Birth φ	28.30 (0.41), 16.87 (0.18)	29.70, 17.7	33.99 (0.68), 14.89 (0.21)
Death τ	0.97 (0.08), 0.58 (0.11)	1.02, 0.60	1.16 (0.11), 0.51 (0.13)
Mutation ξ	0.20 (0.03), 0.06 (0.01)	0.20, 0.06	0.20 (0.02), 0.06 (0.04)

We have not provided any convergence results in this article. Precise convergence results for adaptive SMC methods have been recently obtained in Del Moral et al. (2011); these ideas may be modified to study the algorithm presented here. In more details, one can define the 'limiting' ideal algorithm where the target densities are determined inductively by solving the following equation for ϵ_n at time n

$$\mathrm{ESS}^*(\epsilon_n) = \alpha \mathrm{ESS}^*(\epsilon_{n-1})$$

where, for any $\epsilon > 0$, $\lim_{N\to\infty} \mathrm{ESS}(\epsilon) = \mathrm{ESS}^*(\epsilon)$ almost surely; see Del Moral et al. (2011) for precise results. Then one needs to show that the algorithm which uses the empirical versions of these criteria converge towards the 'limiting' ideal algorithm as N increases.

Acknowledgements We thank the two referees for their useful comments which have helped us to improve significantly the paper.

References

Andrieu, C., Johanes, T.: A tutorial on adaptive Markov chain Monte Carlo methods. Stat. Comput. **18**, 343–373 (2008)

Andrieu, C., Berthelesen, K., Doucet, A., Roberts, G.O.: The expected auxiliary variable method for Monte Carlo simulation. Working paper (2008)

Beaumont, M.A., Cornuet, J.M., Marin, J.M., Robert, C.P.: Adaptive approximate Bayesian computation. Biometrika **86**, 983–990 (2009)

Beskos, A., Crisan, D., Jasra, A.: On the stability of sequential Monte Carlo methods in high dimensions. Technical Report, Imperial College London (2011)

Bortot, P., Coles, S., Sisson, S.: Inference for stereological extremes. J. Am. Stat. Assoc. **102**, 84–92 (2009)

Chopin, N.: A sequential particle filter for static models. Biometrika **89**, 539–552 (2002)

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. R. Stat. Soc. B **68**, 411–436 (2006)

Del Moral, P., Doucet, A., Jasra, A.: On adaptive resampling strategies for sequential Monte Carlo methods. Bernoulli (2011, to appear)

Drovandi, C.C., Pettit, A.N.: Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. Biometrics **67**, 225–233 (2011)

Fearnhead, P., Prangle, D.: Semi-automatic approximate Bayesian computation. Technical Report, University of Lancaster (2010)

Gilks, W.R., Berzuini, C.: Following a moving target—Monte Carlo inference for dynamic Bayesian models. J. R. Stat. Soc. B **63**, 127–146 (2001)

Gramacy, R., Samworth, R., King, R.: Importance tempering. Stat. Comput. **20**, 1–7 (2010)

Grelaud, A.: Méthodes sans vraisemblance appliquées à l'étude de la sélection naturelle et la prédiction de structure tridimensionnelle des protéines. Ph.D. Thesis, Université Paris-Dauphine (2009)

Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comput. Graph. Stat. **5**, 1–25 (1996)

Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer, New York (2001)

Majoram, P., Molitor, J., Plagnol, V., Tavaré, S.: Markov chain Monte Carlo without likelihoods. Proc. Natl. Acad. Sci. **100**, 15324–15328 (2003)

Marin, J.-M., Pudlo, P., Robert, C.P., Ryder, R.: Approximate Bayesian computational methods. Stat. Comput. (2011, to appear)

Neal, R.M.: Annealed importance sampling. Stat. Comput. **11**, 125–139 (2001)

Peters, G.W., Fan, Y., Sisson, S.A.: On sequential Monte Carlo, partial rejection control and approximate Bayesian computation. Technical report, University of New South Wales (2008)

Sisson, S., Fan, Y., Tanaka, M.M.: Sequential Monte Carlo without likelihoods. Proc. Natl. Acad. Sci. **104**, 1760–1765 (2007)

Sisson, S., Fan, Y., Tanaka, M.M.: Correction: Sequential Monte Carlo without likelihoods. Proc. Natl. Acad. Sci. **106**, 1760 (2009)

Small, P.M., Hopewell, P., Singh, P., Paz, A., Parsonnet, J., et al.: The epidemiology of tuberculosis in San Francisco: a population-based study using conventional and molecular methods. N. Engl. J. Med. **330**, 1703–1709 (1994)

Tanaka, M.M., Francis, A.R., Luciani, F., Sisson, A.: Using approximate computation to estimate tuberculosis transmission parameters from genotype data. Genetics **173**, 1511–1520 (2006)

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., Stumpf, M.P.H.: Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. J. R. Soc. Interface **6**, 187–202 (2009)