Bayesian Analysis (2020) **TBA**, Number TBA, pp. 1–27

Adaptive Approximate Bayesian Computation Tolerance Selection

Umberto Simola[*], Jessi Cisewski-Kehe[†], Michael U. Gutmann[‡], and Jukka Corander[§]

Abstract. Approximate Bayesian Computation (ABC) methods are increasingly used for inference in situations in which the likelihood function is either computationally costly or intractable to evaluate. Extensions of the basic ABC rejection algorithm have improved the computational efficiency of the procedure and broadened its applicability. The ABC – Population Monte Carlo (ABC-PMC) approach has become a popular choice for approximate sampling from the posterior. ABC-PMC is a sequential sampler with an iteratively decreasing value of the tolerance, which specifies how close the simulated data need to be to the real data for acceptance. We propose a method for adaptively selecting a sequence of tolerances that improves the computational efficiency of the algorithm over other common techniques. In addition we define a stopping rule as a by-product of the adaptation procedure, which assists in automating termination of sampling. The proposed automatic ABC-PMC algorithm can be easily implemented and we present several examples demonstrating its benefits in terms of computational efficiency.

Keywords: complex stochastic modeling, likelihood-free methods, sequential Monte Carlo.

1 Introduction

Approximate Bayesian Computation (ABC) provides a framework for inference in situations where the relationship between the data and the parameters does not lead to a tractable likelihood function, but where forward simulation of the data-generating process is possible. ABC has been used in many areas of science such as biology (Thornton and Andolfatto, 2006), epidemiology (McKinley et al., 2009; Numminen et al., 2013), ecology (Beaumont, 2010), population modeling (Toni et al., 2009), modeling the population effects of a vaccine (Corander et al., 2017), dark matter direct detection (Simola et al., 2019), and astronomy (Cameron and Pettitt, 2012; Cisewski-Kehe et al., 2019; Ishida et al., 2015; Schafer and Freeman, 2012; Weyant et al., 2013). The basic ABC algorithm (Pritchard et al., 1999; Rubin, 1984; Tavaré et al., 1997) can be explained in four steps. Suppose the parameter vector $\theta \in \mathbb{R}^p$ is the target of inference, then (i) draw the model parameters from the prior distribution, $\theta_{\text{prop}} \sim \pi(\theta)$, (ii) produce a synthetic sample of the data by using θ_{prop} in the forward simulation model, $y_{\text{prop}} \sim f(y \mid \theta_{\text{prop}})$,

[*] Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland, umberto.simola@helsinki.fi

[†] Department of Statistics and Data Science, Yale University, New Haven, CT, USA, jessica.cisewski@yale.edu

[‡] School of Informatics, University of Edinburgh, Edinburgh, United Kingdom, michael.gutmann@ed.ac.uk

[§] Department of Biostatistics, University of Oslo, Oslo, Norway, jukka.corander@medesin.uio.fi

https://doi.org/10.1214/20-BA1211

(iii) compare the true data, y_{obs}, with the generated sample, y_{prop}, using a distance function, $\rho(\cdot,\cdot)$, and defining the distance as $d = \rho(s(y_{\text{obs}}), s(y_{\text{prop}}))$ where $s(\cdot)$ is some (possibly multi-dimensional) summary statistic of the data, (iv) if the distance, d, is less than or equal to a fixed tolerance, ϵ, then θ_{prop} is retained, otherwise it is discarded. This is repeated until a desired particle sample size, N, is achieved.

Following the notation of Marin et al. (2012), the resulting ABC posterior can be written as

$$\pi_\epsilon(\theta \mid y_{\text{obs}}) = \int \left[\frac{f(y_{\text{prop}} \mid \theta)\pi(\theta)\mathbb{I}_{A_{\epsilon,y_{\text{obs}}}}(y_{\text{prop}})}{\int_{A_{\epsilon,y_{\text{obs}}}\times\Theta} f(y_{\text{prop}} \mid \theta)\pi(\theta) dy_{\text{prop}} d\theta} \right] dy_{\text{prop}},$$

where $\mathbb{I}_{A_{\epsilon,y_{\text{obs}}}}(\cdot)$ is the indicator function for the set $A_{\epsilon,y_{\text{obs}}} = \{y_{\text{prop}} \mid \rho(s(y_{\text{obs}}), s(y_{\text{prop}})) \leq \epsilon\}$. There are many extensions to the basic ABC algorithm (e.g., Blum 2010; Blum et al. 2013; Ratmann et al. 2013; Csilléry et al. 2010; Del Moral et al. 2012; Drovandi and Pettitt 2011; Fearnhead and Prangle 2012; Joyce and Marjoram 2008; Marin et al. 2012), but here we focus on the ABC – Population Monte Carlo (ABC-PMC) approach introduced by Beaumont et al. (2009). However, the proposed methodology could be used in other sequential versions of ABC that require selecting a sequence of tolerances. The proposed adaptive approximate Bayesian computation tolerance selection algorithm (aABC-PMC) targets the same kind of approximate posterior sampling problems as the original ABC-PMC algorithm, and may be subject to the same limitations in the case of high-dimensional parameter spaces. ABC has been successfully used in numerous situations where the likelihood function is intractable and the number of parameters varies from 2 to 5 (e.g. Beaumont et al. 2009; Cisewski-Kehe et al. 2019; Csilléry et al. 2010; Cornuet et al. 2008; Del Moral et al. 2012; Gutmann and Corander 2016; Järvenpää et al. 2016; Jennings and Madigan 2016; Jennings et al. 2016; Numminen et al. 2013; Silk et al. 2013; Simola et al. 2019; Sisson et al. 2007; Toni et al. 2009). Our algorithm is designed to significantly improve upon the original ABC-PMC method under similar circumstances.

The ABC-PMC algorithm by Beaumont et al. (2009) is based on an adaptive importance sampling approach, where, given a series of decreasing tolerances $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$ (T being the final iteration), the proposal distribution is sequentially updated in order to improve the efficiency of the algorithm. This is done by constructing a series of intermediate proposal distributions, with the details of the steps presented in Algorithm 1. The first iteration of the ABC-PMC algorithm uses tolerance ϵ_1 and draws proposals from the specified prior distribution(s); the corresponding ABC posterior is denoted by π_{ϵ_1}. Rather than starting the rejection sampling over using a smaller ϵ, the algorithm proceeds sequentially by drawing proposals from the ABC posterior approximated in the previous iteration. After a parameter value, typically referred to as a *particle*, is selected from the set of available particles from the previous iteration, it is also translocated according to some kernel function (e.g. a Gaussian kernel) to avoid degeneracy of the sampler. Since the proposals are not drawn directly from the prior π, importance weights are used. The importance weight for a particle $J = 1,\ldots,N$ at iteration t is:

$$W_t^{(J)} \propto \pi(\theta_t^{(J)}) / \sum_{K=1}^{N} W_{t-1}^{(K)} \phi \left[\tau_{t-1}^{-1} \left(\theta_t^{(J)} - \theta_{t-1}^{(K)} \right) \right], \tag{1.1}$$

Algorithm 1 ABC-PMC algorithm for θ.

```
Given a series of decreasing tolerances $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$
if $t = 1$ then
    for $J = 1, \ldots, N$ do
        Set $d_1^{(J)} = \epsilon_1 + 1$
        while $d_1^{(J)} > \epsilon_1$ do
            Propose $\theta^{(J)}$ by drawing $\theta_{\text{prop}} \sim \pi(\theta)$,
            Generate $y_{\text{prop}} \sim f\left(y \mid \theta^{(J)}\right)$
            Calculate distance $d_1^{(J)} = \rho(s(y_{\text{obs}}), s(y_{\text{prop}}))$
        end while
        Set weight $W_1^{(J)} = N^{-1}$
    end for
else if $2 \leq t \leq T$ then
    Set $\tau_t^2 = 2 \cdot \text{var}\left(\{\theta_{t-1}^{(J)}, W_{t-1}^{(J)}\}_{J=1}^{N}\right)$
    for $J = 1, \ldots, N$ do
        Set $d_t^{(J)} = \epsilon_t + 1$
        while $d_t^{(J)} > \epsilon_t$ do
            Select $\theta_t^*$ from $\theta_{t-1}^{(J)}$ with probabilities $\left\{W_{t-1}^{(J)} / \sum_{K=1}^{N} W_{t-1}^{(K)}\right\}_{J=1}^{N}$
            Propose $\theta_t^{(J)} \sim \mathcal{N}(\theta_t^*, \tau_t^2)$
            Generate $y_{\text{prop}} \sim f\left(y \mid \theta_t^{(J)}\right)$
            Calculate distance $d_t^{(J)} = \rho(s(y_{\text{obs}}), s(y_{\text{prop}}))$
        end while
        Set weight $W_t^{(J)} \propto \pi(\theta_t^{(J)}) / \sum_{K=1}^{N} W_{t-1}^{(K)} \phi\left[\tau_{t-1}^{-1}\left(\theta_t^{(J)} - \theta_{t-1}^{(K)}\right)\right]$
    end for
end if
```

where $\phi(\cdot)$ is the density function of a standard normal distribution,[1] τ_{t-1}^2 is the variance (twice the weighted sample variance of the particles from iteration $t-1$ is used, as recommended in Beaumont et al. 2009), and $\pi(\cdot)$ is the prior distribution. We note that the definition for the importance weight provided in (1.1) is up to a normalization constant. In fact each importance weight is normalized such that $\sum_{J=1}^{N} W_t^{(J)} = 1$. While the particles are drawn from a sequentially improving proposal distribution, the tolerances also decrease such that $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$, to increase the fidelity of the resulting approximation to the underlying posterior. The common strategies for selecting this sequence adaptively, highlighted in Section 1.1, can lead to inefficient sampling as well as avoiding relevant regions of the parameter space (Silk et al., 2013). The key contributions of this article are (i) a method for selecting the $\epsilon_{1:T} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_T)$ in a

[1] The probability density function of a Q-dimensional standard normal distribution is $\phi(X) = (2\pi)^{-\frac{Q}{2}} \exp\left(-\frac{1}{2}X^T X\right)$ with the expected value of the random vector X is $E[X] = \vec{0}$ (where $\vec{0}$ is a Q-dimensional vector of zeros) and its covariance matrix is $\text{Var}[X] = I_Q$, where I_Q is the $Q \times Q$ identity matrix.

manner that results in improved computational efficiency, and (ii) a rule for determining when the algorithm terminates (i.e. determining T).

1.1 Selecting the Tolerance Sequence and Stopping Rules

There are three common approaches for selecting the tolerance sequence, $\epsilon_{1:T}$: (i) fixing the values in advance (Beaumont et al., 2009; McKinley et al., 2009; Sisson et al., 2007; Toni et al., 2009), (ii) adaptively selecting ϵ_t based on some quantile of $\{d_{t-1}^{(J)}\}_{J=1}^{N}$, the distances of the accepted particles from iteration $t-1$ (Cisewski-Kehe et al., 2019; Ishida et al., 2015; Lenormand et al., 2013; Simola et al., 2019; Weyant et al., 2013), or (iii) adaptively selecting ϵ_t based on some quantile of the effective sample size (ESS) values (Del Moral et al., 2012; Numminen et al., 2013). These approaches can lead to inefficient sampling as discussed below and demonstrated in the simulation study in Section 3. It turns out that selecting tolerances using a predetermined quantile can, if not selected wisely, lead to the particle system getting stuck in local modes (Silk et al., 2013). Hence the exact sequence of tolerances has an impact not only on the computational efficiency of the algorithm but also on convergence towards the true posterior. We emphasize, however, that obtaining a high-fidelity approximation to the true posterior using ABC is not guaranteed, as this depends on a number of conditions to be met, including a careful selection of summary statistics. Silk et al. (2013) propose an adaptive approach for selecting the tolerance sequence at each iteration by estimating the threshold-acceptance rate curve (TAR curve), which is used to balance the amount of shrinkage of the tolerance with the acceptance rate. This approach requires the estimation of the TAR curve at each iteration of the algorithm. The naive, but computationally impractical approach to estimating the TAR curve (noted as such in Silk et al. 2013), is to simulate a Monte Carlo estimate of the acceptance rate at a range of different tolerances using the ABC forward model, which would have to be repeated at each iteration of the ABC algorithm. Instead, they suggest a more practical method for estimating the TAR curve by building an approximation to the forward model (in their example, using a mixture of Gaussians and the unscented transform of Julier et al. 2000). The TAR curve approach is able to avoid local optima values, but requires the extra step of building a fast approximation of the ABC data-generating model. Our proposed algorithm is similarly able to avoid local modes, but uses quantities that are directly available in the algorithm. More details are presented in Section 3.

After determining the sequence of tolerances, it is also necessary to determine when to stop a sequential ABC sampling algorithm. An ABC algorithm is often stopped when either a desired (low) tolerance is achieved (Sisson et al., 2007) or after a fixed number of iterations T (Beaumont et al., 2009). Ishida et al. (2015) showed that once the ABC posterior stabilizes, further reduction of the tolerance leads to low acceptance rates without meaningful improvement in the ABC approximation to the posterior. They stop the algorithm once the acceptance rate drops below a threshold set by the user.

The first main contribution of this paper is to extend the ABC-PMC algorithm so that the quantile used to update the tolerance in each iteration, q_t, is automatically and efficiently selected, rather than being fixed in advance to a quantile that is used for each iteration. It is worth noticing that efficiency is not only a matter of having a high acceptance rate, as this can be easily accomplished by using larger quantiles, but rather a balance between the acceptance rate and a suitable amount of shrinkage of the tolerance. Moreover the series of tolerances needs to be selected in such a way that the algorithm avoids getting stuck in local modes. As the second contribution, we develop an automatic stopping rule directly based on the behavior of the sequential ABC posterior.

The rest of the paper is organized as follows. In Section 2 the adaptive selection of q_t for determining the tolerance sequence is presented along with the proposed stopping rule. Section 3 is dedicated to a simulation study to compare quantile-based selection of tolerances using ABC-PMC with the proposed procedure. The final example considered uses real data on colonizations of the bacterium *Streptococcus pneumoniae* (Numminen et al., 2013). Concluding remarks are given in Section 4.

2 Methodology

Using the same quantile to update the tolerance at each iteration can be computationally inefficient and results in the particle system getting stuck in local modes (see the example in Section 3.2). In this section we introduce a method for adaptively selecting the quantile such that each iteration has its own quantile, q_t, set based on the online performance of the algorithm.

2.1 Initial Sampling and Automatic Tolerance Selection Rule

In order to initialize the tolerance sequence we use the following approach. Let N be the desired number of particles to approximate the posterior. The initial tolerance ϵ_1 can be adaptively selected by sampling $N_{\text{init}} = kN$ draws from the prior, for some $k \in \mathbb{Z}^+$ (Cisewski-Kehe et al., 2019). Then the N particles of the N_{init} total particles with the smallest distances are retained, and $\epsilon_1 = \max(d_1^{(1*)}, \ldots, d_1^{(N*)})$, where $d_1^{(1*)}, \ldots, d_1^{(N*)}$ are the N smallest distances of the N_{init} particles sampled. This initialization procedure effectively selects a distance quantile for the first step by the selection of an appropriate k, but making this first step adaptive is easier than trying to guess a good ϵ_1. Trying to specify a reasonable ϵ_1 can be especially challenging when testing different summary statistics or distance functions because the scale of the distances can be different. It is important to note that k must be large enough to result in a satisfactory initial exploration of the parameter space, otherwise the algorithm might get stuck in local regions of the parameter space. This challenge also holds true in general for other ABC algorithms, including when ϵ_1 is predefined (i.e. not set adaptively). Providing a general and suitable value for k regardless of the problem that is considered is challenging, since this choice depends on a number of factors such as the definition of the prior distribution(s), the forward model and where relevant regions of the parameter space are (the latter being unknown). Therefore the parameter k has to be suitably tuned by the user once the forward model and the prior distribution(s) have been defined. The problem of selecting k is further discussed in Section 3.

For the subsequent tolerances, $\epsilon_{2:T}$, the general idea is to gauge the amount of shrinkage for iteration $t+1$ by determining the value of ϵ_{t+1} based on the amount of improvement between $\hat{\pi}_{\epsilon_{t-1}}$ and $\hat{\pi}_{\epsilon_t}$. In particular, we can use the estimated ABC posteriors to select a quantile to update the tolerance for the next iteration, and adjust the next tolerance based on how slowly or rapidly the sequential ABC posteriors are changing. More specifically, after each iteration $t > 1$, the following ratio can be estimated using the weighted particles:

$$\hat{c}_t = \sup_{\theta} \frac{\hat{\pi}_{\epsilon_t}(\theta)}{\hat{\pi}_{\epsilon_{t-1}}(\theta)}. \tag{2.1}$$

Since $\hat{\pi}_{\epsilon_{t-1}}(\theta)$ and $\hat{\pi}_{\epsilon_t}(\theta)$ from (2.1) are both proper densities, they will be either exactly the same, making $\hat{c}_t = 1$, or there must be a place where $\hat{\pi}_{\epsilon_t}(\theta) > \hat{\pi}_{\epsilon_{t-1}}(\theta)$, making $\hat{c}_t > 1$. Then the proposed quantile for iteration t (in order to determine ϵ_{t+1}) is

$$q_t = \frac{1}{\hat{c}_t}, \tag{2.2}$$

which varies between 0 and 1. Small values of q_t imply q_{t-1} lead to a large improvement between $\hat{\pi}_{\epsilon_{t-1}}$ and $\hat{\pi}_{\epsilon_t}$, which then results in a larger percentage reduction of the tolerance for the coming iteration, $t+1$. On the other hand, once the ABC posterior stabilizes, q_t tends to 1 as $\hat{\pi}_{\epsilon_{t-1}}$ and $\hat{\pi}_{\epsilon_t}$ become more similar.

The form of (2.2) was motivated by the Accept-Reject (A/R) algorithm (Andrieu et al., 2003; Robert and Casella, 2013). The A/R algorithm has a target distribution, a proposal distribution, and a rule to decide whether or not an element coming from the proposal distribution should be accepted as an element coming from the target distribution. If the form of the ABC posterior distribution was known, A/R sampling would work as follows. A candidate, θ^*, would be proposed from $\hat{\pi}_{\epsilon_{t-1}}(\theta|y_{\text{obs}})$, and would be accepted with probability $\frac{\hat{\pi}_{\epsilon_t}(\theta^*|y_{\text{obs}})}{c\cdot\hat{\pi}_{\epsilon^*_{t-1}}(\theta|y_{\text{obs}})}$, where $c \in (1, \infty)$ is a positive real constant number selected such that $\hat{\pi}_{\epsilon_t}(\theta|y_{\text{obs}}) \leq c \cdot \hat{\pi}_{\epsilon_{t-1}}(\theta|y_{\text{obs}})$ (Robert and Casella, 2013). In A/R sampling, the unconditional acceptance probability is $\frac{1}{c}$ (Hesterberg, 1988). The constant c acts as a proxy for the difference between the proposal and the target distributions (e.g., if they are the same distribution, then $c = 1$ and all proposals would be accepted).

The ABC algorithm does not follow the A/R sampling scheme, but the notion of $1/c$ relating to the sampling efficiency in the A/R algorithm inspired the proposed adaptive tolerance selection idea. For some future iteration, say iteration $t+1$, the ABC posterior distribution is unknown so the previous two ABC posteriors at iterations $t-1$ and t are used as the proposal and target distributions, respectively, so that $\hat{c}_t$ can be computed in (2.1). If there was a substantial change between $\hat{\pi}_{\epsilon_{t-1}}$ and $\hat{\pi}_{\epsilon_t}$, then $\hat{c}_t$ would be larger resulting in a smaller quantile, q_t, for specifying ϵ_{t+1}. As $\hat{\pi}_{\epsilon_{t-1}}$ and $\hat{\pi}_{\epsilon_t}$ become more similar, larger quantiles q_t are assigned. The proposed form of c_t allows the tolerance selection to be based on changes in the ABC posterior from the previous iteration where substantial changes between iterations $t-1$ and t result in a substantial decrease in the proposed tolerance for ϵ_{t+1}. This continues until a substantial decrease in the proposed tolerance does not result in a substantial change in the ABC posterior, at which point the amount of shrinkage in the tolerance becomes smaller.

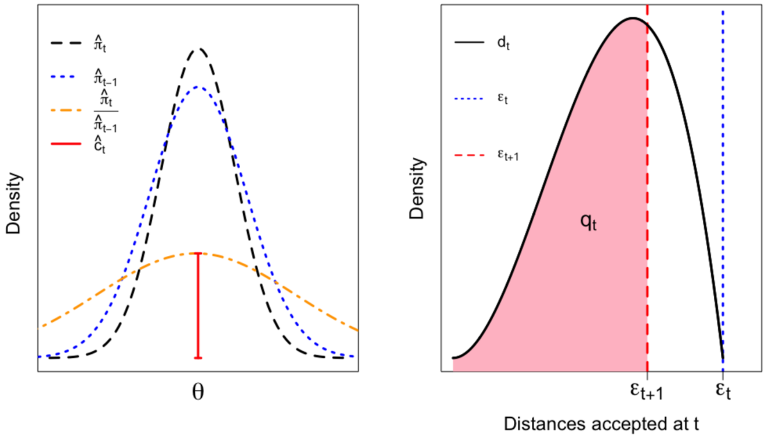


Figure 1: Illustration of the selection of q_t. (left) The proposal distribution ABC posterior $\hat{\pi}_{t-1}$, the resulting ABC posterior $\hat{\pi}_t$ and their ratio $\frac{\hat{\pi}_t}{\hat{\pi}_{t-1}}$, with $\hat{c}_t$ defined according to (2.1) and used for setting q_t, as defined in (2.2). (right) The (arbitrary) distribution of distances is from the accepted distances at iteration t, $\{d_t^{(J)}\}_{J=1}^N$, with ϵ_t being the largest possible value. The next iteration's tolerance, ϵ_{t+1}, is set as the q_t quantile of $\{d_t^{(J)}\}_{J=1}^N$.

We found the rule based on (2.2) to work well empirically. One challenge with a theoretical evaluation of the proposed algorithm, and other algorithms designed to optimize the tolerance shrinkage and acceptance rate, is that the acceptance rate depends on the forward simulation model. In general ABC settings, the forward simulation model does not have a closed-form expression.

An illustration of the proposed quantile selection procedure is provided in Figure 1. If $\hat{\pi}_{t-1}$ was used as the proposal for iteration $t+1$ (instead of $\hat{\pi}_t$), then q_t would be the percentage decrease in the acceptance rate from iteration t, *i.e.* if acc_t is the acceptance rate for iteration t, then acc_{t+1} would be approximately $q_t \times \text{acc}_t$. However, we are not proposing from $\hat{\pi}_{t-1}$, but rather $\hat{\pi}_t$ so the *decrease* in the acceptance rate is mitigated by the *improvement* in the proposed particles from iteration t. When there is a large improvement in the ABC posterior from $\hat{\pi}_{t-1}$ to $\hat{\pi}_t$, then q_t is smaller, allowing for a larger drop in the tolerance. This larger percentage drop in tolerance does not result in an equal percentage drop in acceptance rate because the new proposal distribution, $\hat{\pi}_t$, is better than $\hat{\pi}_{t-1}$. Conversely, if $\hat{\pi}_{t-1}$ is close to $\hat{\pi}_t$, then the improvement in the ABC posterior is not enough to allow for a large decrease in the acceptance rate and consequently q_t is closer to 1.

The evaluation of (2.1) relies on the calculation of the ratio between the (possibly multidimensional) density functions, defined here as r. A naive solution would be to separately calculate the density for $\hat{\pi}_t$ and $\hat{\pi}_{t-1}$ using some Kernel Density Estimate (KDE) method (see Silverman 2018 for a review), and then estimate the ratio from those estimates. Then, the supremum of the previously calculated ratio can be obtained, for example, through an optimization procedure that computes the density over a grid of values. However, this is not a reliable solution, in particular for high-dimensional cases for which division by an estimated quantity can magnify the estimation error (Sugiyama et al., 2008). In order to address the problem of properly estimating r with $\hat{r}$, and therefore solving (2.1), alternatives to the KDE solution are available, such as ratio estimation methods (REM) (Sugiyama et al., 2012). The main advantage of using REM is that the calculation of the desired ratio does not include density estimation, which would involve dividing by an estimated KDE. Additionally, when using a KDE, kernel and bandwidth need to be selected, which can affect the result. Poorly estimating the density of the denominator of r, in particular, can potentially increase the error of the estimated ratio (Sugiyama et al., 2010). There are several different REM frameworks (e.g. Bickel et al. 2007; Gretton et al. 2009; Sugiyama et al. 2008, 2010), but we use the ratio matching approach of Sugiyama et al. (2008) discussed in more detail next.

In order to introduce the REM framework, consider $\theta \in \mathbb{R}^p$ and two generic samples $\{\theta_i^L\}_{i=1}^L$ and $\{\theta_j^M\}_{j=1}^M$, where L and M are the sample sizes for the first and the second sample, respectively. The sample $\{\theta_i^L\}_{i=1}^L$ has as corresponding density $p_L(\theta)$, while the sample $\{\theta_i^M\}_{i=1}^M$ has as corresponding density $p_M(\theta)$. The density ratio $r(\theta)$ can be defined as $r(\theta) = \frac{p_L(\theta)}{p_M(\theta)}$. The basic idea of the ratio matching approach is to match a density ratio model $\hat{r}(\theta)$ with the true density ratio $r(\theta)$ under some divergence (Sugiyama et al., 2010). Several divergences can be used to compare $\hat{r}(\theta)$ with $r(\theta)$. A common divergence is the Bregman divergence (Bregman, 1967), along with some of its related divergences such has the unnormalized Kullback-Leibler divergence and the squared distance. In particular, the unnormalized Kullback-Leibler divergence minimizes the divergence between $p_L(\theta)$ and $\hat{p}_L(\theta) = \hat{r}(\theta)p_M(\theta)$ by means of the following criterion:

$$\min_{\hat{r}} \int p_L(\theta) \log \frac{p_L(\theta)}{\hat{r}(\theta)p_M(\theta)} d\theta. \tag{2.3}$$

By decomposing the Kullback-Leibler divergence defined in (2.3), $\hat{r}(\theta)$ can be estimated by solving the objective function $\max_{\hat{r}} \int p_L(\theta) \log \hat{r}(\theta) d\theta$ (Hido et al., 2011; Sugiyama et al., 2010). Further details on the unnormalized Kullback-Leibler divergence and on other REM approaches are found in Sugiyama et al. (2012). As pointed out by Sugiyama et al. (2010), a further non-negligible advantage of using REM, and in particular the ratio matching approach, is the applicability of gradient-based algorithms and quasi-Newton methods for optimization over $\hat{r}(x)$.

In the analyses of the present work we use the ratio matching approach and the Kullback–Leibler importance estimation procedure (KLIEP) (Hido et al., 2011; Sugiyama et al., 2010, 2008) in order to estimate, at the end of each iteration t, the ratio of densities defined in (2.1). Recall that the densities involved in (2.1) are $\hat{\pi}_{\epsilon_t}(\theta)$ and $\hat{\pi}_{\epsilon_{t-1}}(\theta)$. Once the ratio between $\hat{\pi}_{\epsilon_t}(\theta)$ and $\hat{\pi}_{\epsilon_{t-1}}(\theta)$ has been estimated, the supremum of (2.1) is calculated by using an optimizer over the parameter space, such as the one proposed by Brent (2013). The quantile used to reduce the tolerance for the coming iteration is finally retrieved by using (2.2). The steps discussed above are performed at the end of each iteration as long as the stopping rule, defined in (2.5) and discussed below, is not satisfied. Estimation of $\hat{r}$ is carried out by using the densratio package,[2] which is freely available in the R software (R Core Team, 2019).

The acceptance rate is also useful for evaluating the computational burden of the ABC-PMC algorithm, defined as:

$$\text{acc}_t = \frac{N}{\text{D}_t}, \tag{2.4}$$

where D_t is the number of draws done at iteration t in order to produce N accepted values. Equation (2.4) generally decreases with each iteration because as the tolerance decreases, the number of elements D_t required to get N accepted particles generally increases (Lintusaari et al., 2017).

2.2 Stopping Rule

There are several published ideas in the literature on how to determine the number of iterations in an ABC-PMC algorithm. Often one picks some T based on the computational resources available, but this can be needlessly inefficient. Ishida et al. (2015) proposed to stop the algorithm once the acceptance rate is smaller than some specified, fixed tolerance. The proposed stopping rule is directly based on the estimated sequential ABC posterior distributions, which avoids unnecessary additional iterations of the algorithm.

The ABC-PMC algorithm produces a sequence of T posterior distributions, $\hat{\pi}_{\epsilon_t}$, where ϵ_t identifies the tolerance used in iteration t, with $t = 1, \ldots, T$ and $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$. When defining a stopping rule, it turns out that (2.2) can be used not only to adaptively selecting the quantile used to reduce the tolerance across the iterations, but also to indicate when to stop the procedure once the sequential ABC posterior stops changing significantly.

The series of quantiles defined through (2.2) generally increases as the tolerance decreases. In particular, since the quantile used to reduce the tolerance is based on the online performance of the ABC posterior distribution, once the ABC posterior has stabilized, $q_t \approx 1$. This follows directly from (2.1) because once the ABC posterior has stabilized $\hat{c}_t \approx 1$, and further reductions of the tolerance (i.e. additional iterations) do not necessarily lead to an improvement by the ABC posterior distribution. In other words, once the ABC posterior stabilizes, the series of the quantiles defined through (2.2) stops increasing and the upper bound of 1 implies that no further reduction will improve the ABC posterior distribution. This leads to an automatic and simple stopping rule, which is employed starting from the third iteration, *i.e.* once the transformation kernel has been used twice to avoid premature stopping. Our algorithm is stopped at time t when

$$q_t > 0.99 \text{ for } t \geq 3. \tag{2.5}$$

Hence, the algorithm is stopped once the quantile used to reduce the tolerance suggests that further reduction is not necessary since the ABC posterior has stabilized.

[2] `https://github.com/hoxo-m/densratio`.

Using (2.2) as an automatic rule to shrink the tolerance and (2.5) as the stopping rule, the ABC-PMC algorithm is stopped once additional iterations with smaller tolerances do not lead to significant changes in the ABC posterior.[3]

3 Illustrative Examples

Next we provide a comparison between the original ABC-PMC algorithm and our extension proposed in Section 2, the aABC-PMC, by using three examples. In the first example the Gaussian mixture model by Sisson et al. (2007) is used in order to demonstrate the computational efficiency of the proposed aABC-PMC procedure. Then the aABC-PMC algorithm is used for a model from Silk et al. (2013), which has local modes, in order to illustrate how the proposed automatic tolerance selector is able to avoid getting stuck in local regions of the parameter space. The final example, originally presented in Numminen et al. (2013), uses data on colonizations of the bacterium *Streptococcus pneumoniae* and represents a computationally expensive forward model. Expensive forward models are a challenge for ABC methods because the computational cost can be prohibitive for practical applications, and in these cases selecting an appropriate sequence of tolerances is crucial. A fourth example, the Lotka–Volterra model by Toni et al. (2009), is presented (see Appendix A in Supplementary Material, Simola et al., 2020).

In order to compare the proposed procedure with the original ABC-PMC algorithm, both the computational time and the total number of draws until the stopping criterion is satisfied are considered. The Hellinger distance is used for evaluating the similarity between the 1-dimensional marginal ABC posterior distributions at the final iteration, $\hat{\pi}_{\epsilon_T}$, and a benchmark, π_{true}, which is defined as:

$$H(\hat{\pi}_{\epsilon_T}, \pi_{\text{true}}) = \left(\int \left(\sqrt{\hat{\pi}_{\epsilon_T}(y)} - \sqrt{\pi_{\text{true}}(y)} \right)^2 dy \right)^{\frac{1}{2}}. \tag{3.1}$$

The benchmark, π_{true}, is the true posterior distribution if it is available in closed form, which is the case in the first two presented examples (see Sections 3.1 and 3.2). In the final example, since the true posterior distribution is not available, the ABC posteriors from Numminen et al. (2013), are used as benchmarks (see Section 3.3).

In order to estimate the 1-dimensional marginal ABC posterior distributions from the samples and their corresponding importance weights, a KDE (Silverman, 2018) is used with a Gaussian kernel and a smoothing bandwidth parameter h. The bandwidth is selected using Silverman's rule-of-thumb (Silverman, 1986).

Finally, unless otherwise noted, the number of particles in the ABC procedures is set to $N = 1,000$.

[3] The desired sample size N has an impact on the evaluation of (2.5). This problem arises also in the classical Markov Chain Monte–Carlo (MCMC) analysis when determining the length of the MCMC chain (Gelman et al., 2014). An N that is too small leads to more variability of the estimated posterior in (2.5), which could lead to the algorithm stopping prematurely.

Sisson et al. (2007)				aABC-PMC				
t	ϵ_t	D_t	H_{dist}	t	ϵ_t	q_t	D_t	H_{dist}
1	1.000	2,595	0.34	1	1.96		5,000	0.39
2	0.5013	8,284	0.29	2	0.45	0.20	7,095	0.29
3	0.2519	8,341	0.26	3	0.072	0.15	24,216	0.22
4	0.1272	7,432	0.24	4	0.035	0.45	44,919	0.20
5	0.0648	10,031	0.23					
6	0.0337	17,056	0.20					
7	0.0181	34,178	0.21					
8	0.0102	72,704	0.20					
9	0.0064	171,656	0.19					
10	0.0025	1,089,006	0.20					
Total		1,421,283					81,230	

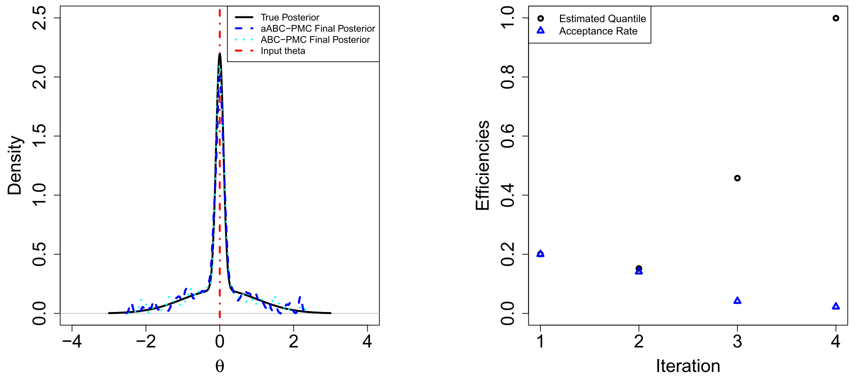
Table 1: Gaussian mixture model. The number of draws needed in each iteration to reach $N = 1,000$ accepted values for the ABC-PMC and the aABC-PMC algorithm. (The displayed results were obtained by running the procedure 21 times and using the run that produced the median number of total draws.) For the aABC-PMC algorithm, the quantile automatically selected through the iterations is displayed under q_t. The procedure stopped once the quantile $q_5 = 0.999$ was proposed. For the ABC-PMC algorithm a total of $1,421,283$ (243 sec.) draws were required, while our aABC-PMC takes $81,230$ (88 sec.) draws overall.

3.1 Gaussian Mixture Model

The first application of the aABC-PMC is an example from Sisson et al. (2007), which is also analyzed by Beaumont et al. (2009). It is a Gaussian mixture model with two Gaussian components with known variances and mixture weights, but an unknown common mean, $f(y \mid \theta) = 0.5\mathcal{N}(\theta, 1) + 0.5\mathcal{N}(\theta, 0.01)$ and prior $\pi(\theta) \sim \mathrm{Unif}(-10, 10)$. With a single observation $y_{\mathrm{obs}} = 0$, the true posterior distribution is

$$\pi(\theta \mid y_{\mathrm{obs}}) \sim 0.5\mathcal{N}(0, 1) + 0.5\mathcal{N}(0, 0.01). \tag{3.2}$$

For consistency with the results of Sisson et al. (2007) and Beaumont et al. (2009), the distance function used is $\rho(y_{\mathrm{obs}}, y_{\mathrm{prop}}) = |y_{\mathrm{obs}} - y_{\mathrm{prop}}|$, $N = 1,000$, and a Gaussian kernel for resampling the particles is used. Both Sisson et al. (2007) and Beaumont et al. (2009) manually define the series of tolerances. In particular, Sisson et al. (2007) carry out $T = 10$ iterations with a fixed series of tolerances $\epsilon_{1:10}$ displayed in Table 1. To evaluate the reliability of the aABC-PMC, a comparison with the ABC-PMC is done both in terms of computational time and total number of draws. The results of the analysis are shown in Table 1 and are based on 21 independent runs with the same dataset, $y_{\mathrm{obs}} = 0$. The table includes the values for the run that produced the median number of total draws. The aABC-PMC outperforms ABC-PMC in the terms of total draws (81,230 vs 1,421,283) and a faster computational time (88 seconds vs



(a) Final ABC posteriors.

(b) aABC-PMC quantities.

Figure 2: Gaussian mixture model example. (a) ABC-PMC and aABC-PMC final posterior distributions and (b) sequential quantities computed for the aABC-PMC method. The q_t's (black circles) generally increase through the iterations until the ABC posterior has stabilized. The acceptance rate (blue triangles) decreases throughout the iterations, which is why it is desirable to stop the algorithm once the ABC posterior has stabilized.

243 seconds). The final ABC posteriors for each method are displayed in Figure 2a. Though the aABC-PMC method is computationally more efficient than the ABC-PMC approach, the final ABC posteriors are very similar. This suggests that after a suitable tolerance is achieved, decreasing the tolerance further does not necessarily lead to a better approximation of the posterior distribution.

From Table 1, we note that the final tolerance for Sisson et al. (2007) is $\epsilon_{10} = 0.0025$ ($H_{\text{dist}} = 0.20$) while the automatic stopping rule of aABC-PMC leads to 4 iterations with a final tolerance of $\epsilon_4 = 0.035$ ($H_{\text{dist}} = 0.20$). In Figure 2b, the q_t's retrieved by using (2.2) are displayed (black circles), which increase until the final iteration, while the acceptance rate (blue triangles) decreases. Neglecting to stop the algorithm once the ABC posterior has stabilized can be inefficient since the number of draws needed in order to complete further iterations can drastically increase, as evidenced by the increasing D_t for later iterations displayed in Table 1.

Next, we show the behavior of the aABC-PMC algorithm for different choices of the number of proposed values from the prior distribution at the first iteration of the procedure. Initial particle sample sizes, N_{init}, of N, 2N, 5N, and 10N are considered (with $N = 1,000$), and the results are displayed in Table 2. The initial particle sample size that seems to best balance the total number of draws and the time required to satisfy the stopping rule in this example is $5N$, with similar final ABC posterior distributions based on H_{dist} (see Table 2); the posteriors are displayed in Figure 3.

	T	D_t	ϵ_1	ϵ_T	time (sec)	H_{dist}
N	14	276,885	11.54	0.035	208	0.23
2N	10	109,720	4.97	0.077	150	0.29
5N	4	81,230	1.96	0.035	88	0.20
10N	4	90,194	1.00	0.059	105	0.17

Table 2: aABC-PMC algorithm with different choices for N_{init} $(N, 2N, 5N, 10N)$ for the Gaussian mixture model example.

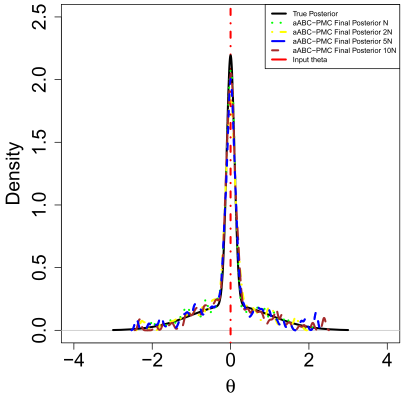


Figure 3: aABC-PMC posteriors with different choices for N_{init} $(N, 2N, 5N, 10N)$ for the Gaussian mixture model example.

Using the EasyABC `R` package[4] we carried out the same analysis for the ABC-SMC algorithm by Del Moral et al. (2012). The ABC-SMC algorithm by Del Moral et al. (2012) is discussed in Section 3.3. For each initial particle sample size, N_{init}, of N, 2N, 5N, and 10N, 21 independent runs with the same dataset are performed and the runs that produced the median number of total draws are compared to the corresponding run obtained by our adaptive approach. Our choices for setting the parameters required by the ABC-SMC algorithm (see Section 3.4) are: $N = 1,000$, $\epsilon = 0.035$, $\alpha = 0.5$, $M = 1$ and $\text{nb}_{\text{threshold}} = N/2$. We note that for the last three parameters, the default values are used, according to the suggestions by Del Moral et al. (2012). The results of the analysis are summarized in Table 3 and the corresponding posterior distributions are displayed in Figure 4. For all four N_{init} values considered, the final tolerances returned by the ABC-SMC algorithm are comparable with the one obtained by our approach with $N_{\text{init}} = 5N$ $(\epsilon_4 = 0.035)$. However the corresponding ABC-SMC posterior distributions do not match the true posterior distribution as well as our proposed approach. In particular, the ABC-SMC algorithm does not seem to capture the (low) variance coming from the second component of the Gaussian mixture model. Similar results were also obtained by the ABC-SMC sampler proposed in Bonassi and West (2015). A further comparison when using the ABC-SMC algorithm by Del Moral et al. (2012) is available in the Appendix B of the Supplementary Material.

[4] `https://cran.r-project.org/web/packages/EasyABC`.

	D_t	time (sec)	H_{dist}	ϵ_t
N	25,023	17	0.35	0.037
2N	47,192	54	0.37	0.031
5N	124,890	322	0.33	0.031
10N	249,696	1254	0.34	0.03

Table 3: ABC-SMC algorithm with different choices for N_{init} $(N, 2N, 5N, 10N)$ for the Gaussian mixture model example.

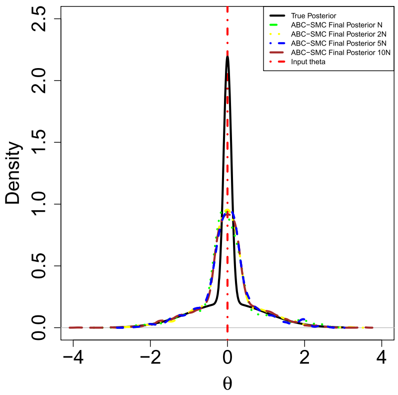


Figure 4: ABC-SMC final posterior distributions with different choices for N_{init} $(N, 2N, 5N, 10N)$ for the Gaussian mixture model example.

3.2 Presence of a Local Mode

The sequence of tolerances has an impact not only on the computational efficiency of the algorithm, but also on its ability to find the true posterior (Silk et al., 2013), noting again that convergence to the true posterior using ABC is not guaranteed. To demonstrate the performance of aABC-PMC in the presence of local modes, we consider an example proposed in Silk et al. (2013). The (deterministic) forward model is $g(\theta) = (\theta - 10)^2 - 100\exp(-100(\theta - 3)^2)$. The input value is set to $\theta = 3$ leading to a single observation $y_{\text{obs}} = -51$. The true posterior distribution is a Dirac function at 3. The specifications for the distance function (L^1 norm), the prior distribution (a normal distribution with mean of 10 and variance of 10), and the desired number of particles $(N = 1,000)$ are taken from Silk et al. (2013).

Figure 5 displays the locations of the accepted particles (orange x's) against the distances for a range of θ's, which highlights the challenge for ABC with this model. There is a local minimum distance around $\theta = 10$, but the global minimum distance occurs at the true value of $\theta = 3$. Initial steps of the ABC algorithm will find the local minimum, but the algorithm can easily get stuck around $\theta = 10$ if the sequential tolerances are not selected carefully. The series of plots in Figure 5 shows the behavior of the aABC-PMC algorithm by focusing on the values for θ that were accepted (orange x's). After 6 iterations, the aABC-PMC algorithm has found the global minimum distance around

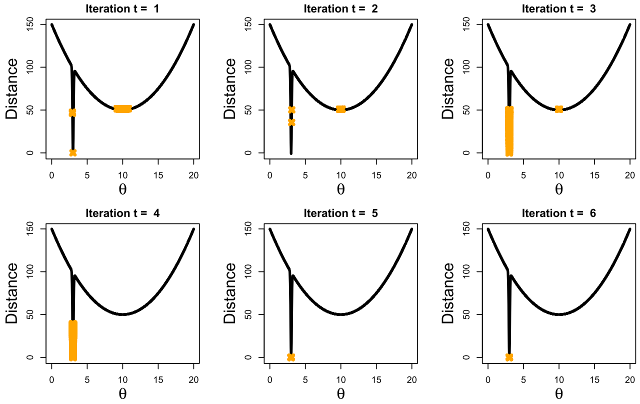


Figure 5: Example from Silk et al. (2013) to investigate the performance of the proposed aABC-PMC in the presence of a local optimal value. The accepted θ are plotted as orange x's against the corresponding distance by iteration.

the true θ. The results of the analysis, based on 21 independent runs, are summarized in Table 4, where $384,347$ total particles were used by the proposed aABC-PMC algorithm. The table includes the values for the run that produced the median number of total draws.

It is apparent from Figure 5 that the third iteration was an important step in which the large reduction of the tolerance allowed the algorithm to consider those few particles coming from the global optimal value at $\theta = 3$. Although the raw tolerance hardly decreases between the first and the second iteration ($\epsilon_1 = 51.59$ and $\epsilon_2 = 51.02$), there is a substantial change between the ABC posteriors, from $\hat{\pi}_{\epsilon_2}$ to $\hat{\pi}_{\epsilon_3}$. The majority of the accepted values from $t = 2$ are sampled near the local mode at $\theta = 10$, but the reduction resulting from the slightly smaller ϵ_3 leads to the majority of values proposed near $\theta = 3$ to be accepted.

In order to compare the proposed aABC-PMC algorithm with the ABC-PMC approach of Silk et al. (2013) (see Section 1.1), we estimated the TAR curve and the corresponding thresholds (Silk et al., 2013). The TAR curve is obtained by plotting on the x-axis several thresholds ϵ that might be picked for the next iteration of ABC simulations and on the y-axis their corresponding acceptance rates. The threshold ϵ recommended for the next ABC-PMC iteration is then selected by locating the "elbow" of the estimated TAR curve (Silk et al., 2013). Since the forward model is computationally cheap, an approximation to the forward model was not needed. Instead, the TAR curve was estimated at each iteration by setting arbitrary grid points of tolerances having range in $(0, \epsilon_{t-1})$, running the ABC-PMC algorithm (for $t > 1$ the previous iteration's particle system and the Gaussian perturbation kernel are used), and then calculating the acceptance rate according to (2.4). This procedure was repeated 100 times and the resulting average TAR curve was used to retrieve the tolerance for the coming iteration, as was done in Figure 2(left) of Silk et al. (2013). As result, a plot of acceptance rate vs.

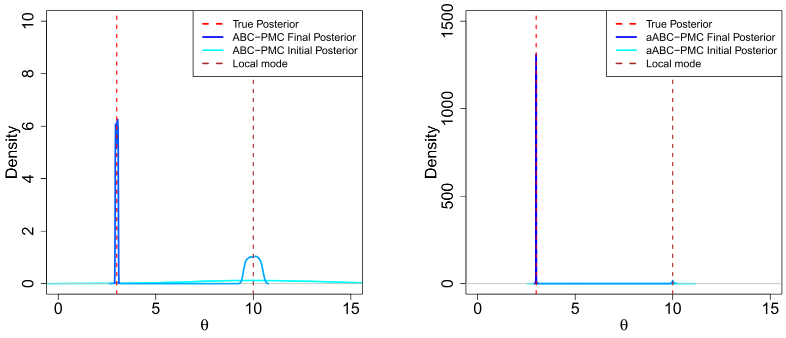
TAR curve (Silk et al., 2013)				aABC-PMC				
t	ϵ_t	D_t	H_{dist}	t	ϵ_t	q_t	D_t	H_{dist}
1	150	1,000	1.37	1	51.59		5,000	1.38
2	51.26	11,560	1.26	2	51.02	0.19	8,130	1.36
3	50.84	1,403,040	0.174	3	51.00	0.16	99,596	0.68
				4	39.33	0.17	138,972	0.43
				5	0.07	0.06	32,045	0.067
				6	0.00025	0.90	100,604	0.064
Total		1,415,600					384,347	

Table 4: The number of draws needed in each iteration to reach $N = 1,000$ accepted values for the ABC-PMC with the TAR curve-selected tolerances and the aABC-PMC algorithm. (The displayed results were obtained by running the procedure 21 times and using the run that produced the median number of total draws.) For the aABC-PMC algorithm, the quantile automatically selected through the iterations is displayed under q_t. The procedure stopped once the quantile $q_7 = 0.9991$ was calculated. For the ABC-PMC algorithm a total of $1,415,600$ (310 sec.) draws are required, while our aABC-PMC takes $384,347$ (258 sec.) draws overall. The number of draws listed for Silk et al. (2013) does *not* include the draws required to build the TAR curve; however, we did include the TAR curve construction in the computational time.

tolerances was obtained; the tolerance is set at the value corresponding to the elbow of the TAR curve. The series of tolerances, displayed together with the number of draws in Table 4, is $\epsilon_{1:3} = (150, 51.26, 50.84)$ and the corresponding ABC posterior distributions are displayed in Figure 6a. The number of draws listed for Silk et al. (2013) does *not* include the draws required to build the TAR curve; however, we did include the TAR curve construction in displayed computational time. We note that the true posterior distribution, which is a Dirac function centered in $\theta = 3$, is not suitably approximated by Silk et al. (2013) ($H_{\text{dist}} = 0.17$).

In order to calculate the Hellinger distance in this example, we approximate the true posterior (i.e., a Dirac function at $\theta = 3$) with an N-dimensional vector with all elements equal to 3.

For $t = 4$, the estimated TAR curve did not have an elbow and, consequently, there was no additional shrinkage of the tolerance resulting in an ABC posterior that was not a suitable approximation to the true posterior distribution; the final tolerance ϵ_3 was too high. We tried making adjustments to the TAR curve grid to see if this could be improved. When using fewer grid points (e.g. 10) for the TAR curve, we were able to improve the performance. However, this improved performance was due to poorer approximation to the TAR curve. In general, it would be preferable if a better estimate of the TAR curve lead to better performance. A higher resolution TAR curve grid with 1000 grid points also was not able to find the global optimal solution. In contrast, as shown in Figure 6b, the proposed aABC-PMC approach provides a better approximation of the true posterior distribution although the number of draws required by the simulator



(a) ABC posterior distributions (Silk et al., 2013).

(b) aABC-PMC posterior distributions.

Figure 6: ABC posterior distributions by iteration using (a) the TAR curve, and (b) the proposed aABC-PMC algorithm. The true posterior distribution, which is a Dirac function centered at $\theta = 3$ is better captured by the aABC-PMC algorithm ($H_{\text{dist}} = 0.064$), compared to the ABC-PMC method based on the TAR curve ($H_{\text{dist}} = 0.17$). Note that the vertical axes are on different scales.

is only of $384,347$ (compared to 1,415,600 draws required by ABC-PMC with the 100 point TAR curve grid).

Silk et al. (2013) note that if the particles are sampled from a large region of the parameter space that has a negligible mass in the posterior distribution, there is a risk of getting stuck in this parameter region if the tolerance is not selected carefully. In other words, the parameter space needs to be sufficiently explored in order to get enough particles in regions near the global optimal value. In the first iteration of the aABC-PMC algorithm the number of particles sampled directly from the prior was kN with $k = 5$, which seems to work well in the examples considered. We emphasize that moving toward relevant regions of the parameter space needs to happen in the first few iterations of the ABC-PMC procedure, since uniformly small reductions in the tolerance sequence (e.g. using a fixed $q_t \geq 0.25$) could end up removing those few important particles near the global optimal value, even if the number of particles sampled directly from the prior is $5N$.

The initial exploration of the parameter space and the definition of small enough quantiles in the first iterations appears to be why in the procedure based on the TAR curve, the total number of draws needed by the ABC-PMC algorithm is large, making it very expensive computationally. In fact, at the end of the second iteration, the majority of the previous iteration's accepted particles are drawn near the local minimum. Moreover, since their $N_{\text{init}} = N$, only few candidates close to the global optimum are available. This means that when a particle is resampled, it will likely come from regions near to the local minimum and therefore it may be easily rejected during the third iteration of the ABC-PMC algorithm, for which the selected tolerance is $\epsilon_3 = 50.84$.

	D_t	time (sec)	H_{dist}	ϵ_t
N	56,535	31	0.374	0.00027
2N	111,478	106	0.368	0.00050
5N	278,412	606	0.388	0.00017
10N	521,945	2305	0.41	0.00022

Table 5: ABC-SMC algorithm with different choices for N_{init} $(N, 2N, 5N, 10N)$ for the Silk et al. (2013)'s model.

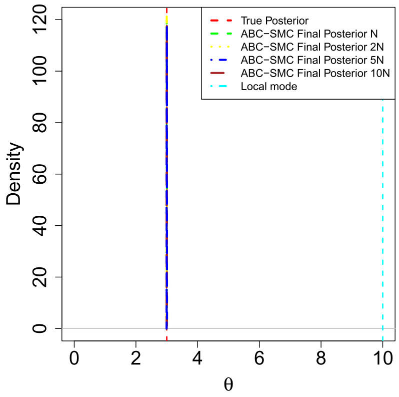


Figure 7: ABC-SMC final posterior distributions with different choices for N_{init} $(N, 2N, 5N, 10N)$ for the Silk et al. (2013)'s model.

The proposed aABC-PMC algorithm allows for small q_t's early on, when larger improvements occur between the sequential ABC posteriors. By doing so, larger reductions in the tolerance sequence can be taken in the first iterations of the ABC-PMC, which results in moving away from local optimal values into better regions of the parameter space. If a sufficient reduction of the tolerance is not made early on, achieving a good approximation of the true posterior distribution is unlikely because the distances associated with the local optimal values will overwhelm the particle system so that it gets stuck in the local region.

As previously done for the Gaussian Mixture Model example presented in Section 3.1, we conclude the analysis of this model by performing a comparison between our adaptive aABC-PMC approach and the ABC-SMC algorithm by Del Moral et al. (2012). Again, four initial particle sample sizes of N_{init} are considered (N, 2N, 5N, and 10N) and 21 independent runs with the same dataset are performed. The results include the runs that produced the median number of total draws and are compared to the corresponding results obtained by our adaptive approach. The 5 parameters required by the ABC-SMC algorithm have been fixed as follows: $N = 1,000$, $\epsilon = 0.00025$, $\alpha = 0.5$, $M = 1$ and $\text{nb}_{\text{threshold}} = N/2$. We note again that default values are used for the last three parameters, following the suggestions by Del Moral et al. (2012). The results of the analysis are summarized in Table 5 and the corresponding posterior distributions are displayed in Figure 7. From Table 5 with $k = 5$, although the number of total draws of the ABC-SMC algorithm is smaller than the corresponding total number of draws obtained by the adaptive aABC-PMC, our procedure is faster in terms of computational time. Moreover, the final ABC posterior distribution obtained by the aABC-PMC algorithm ($H_{\text{dist}} = 0.064$) matches the true posterior distribution better than the one obtained by the ABC-SMC sampler ($H_{\text{dist}} = 0.388$). On the other hand, the ABC-SMC sampler successfully explores relevant regions of the parameter space for $k = 1$ and $k = 2$, while our aABC-PMC failed to reach the global mode for $k = 1, 2$ because too few particles from the global mode were drawn in the first iteration of the procedure. However, the final ABC posterior distribution obtained by the aABC-PMC algorithm with the recommended $k = 5$, and for which $H_{\text{dist}} = 0.064$, better matches the true posterior compared to any ABC posterior distribution obtained by the ABC-SMC algorithm (Figure 7).

3.3 Bacterial Infection in Day Care Centers Example

The final model we consider, discussed by Numminen et al. (2013), uses data on colonizations of the bacterium *Streptococcus pneumoniae*. Discussion about mathematical models for such scenarios, known as household models, can be found in Hoti et al. (2009) or Brooks-Pollock et al. (2011). According to the specifications provided in Numminen et al. (2013), the transmission process is modeled with four parameters. Two parameters, β and Λ, account for the hazards of infection from the day care center and from the community, respectively. Another parameter, θ, scales the probability of co-infection. Finally, the parameter γ corresponds to the rate of clearance of an infection. In the following analyses we considered $\gamma = 1$ fixed and known, to be consistent with the analysis in Numminen et al. (2013).

The observed data consists of the identified pneumococcal strains in a total of 611 children from 29 day care centers, with varying numbers of sampled attendees per day (Vestrheim et al., 2008, 2010). For each of the 29 day care centers, a binary matrix with varying number of sampled attendees is available. For each sampled attendee, the state of carrying one of the 33 different pneumococcal strains or not is indicated by a 1 or 0, respectively, in the binary matrix. As pointed out in Gutmann and Corander (2016) statistical inference is challenging in this setting since the data represent a snapshot of the state of the sampled attendees at a single time point only. Moreover, the modeled system involves infinitely many correlated unobserved variables, since the modeled process evolves in continuous time. Using the observed colonizations with bacterial strains, the following four summary statistics are obtained for each of the 29 day care centers: the Shannon index of diversity of the distribution of the observed strains, the number of different strains, the prevalence of carriage among the observed individuals, and the prevalence of multiple infections among the observed individuals. By doing so, the dimensionality of the problems reduces from a $611 \cdot 33 \cdot 29 = 584,727$ dimensional space to a $4 \cdot 29 = 116$ dimensional space.

Numminen et al. (2013) use the four summary statistics and four tolerances, $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4)$, for each iteration of their procedure. Instead, we use the approach of Gutmann and Corander (2016). Each of the four summary statistics is rescaled so that the maximum value for each of the four the summary statistics is one. Then the summary statistics are vectorized in order to obtain a single vector of dimension 116. Finally the L^1 distance between the vector corresponding to y_{prop} and the vector corresponding to y_{obs} is calculated, with the result divided by 116. By doing so, only one tolerance is used in the ABC procedure.

The series of tolerances used in Numminen et al. (2013) was based on the ABC-Sequential Monte Carlo (ABC-SMC) method proposed by Del Moral et al. (2012). The ABC-SMC method of Del Moral et al. (2012) adaptively proposes a series of tolerances by estimating, at the end of each iteration, the effective sample size (ESS). For a generic iteration t the ESS is defined as:

$$\text{ESS}(\{W_t^{(J)}\}_{J=1}^N) = \left(\sum_{J=1}^{N}\left(W_t^{(J)}\right)^2\right)^{-1}, \tag{3.3}$$

where $W_t^{(J)}$ is the importance weight for particle $J = 1, \ldots, N$ at iteration t as defined in (1.1). Once the ESS is estimated by using (3.3), the new tolerance ϵ_{t+1} is obtained by solving the following for ϵ_{t+1}:

$$\text{ESS}(\{W_t^{(J)}\}_{J=1}^N, \epsilon_{t+1}) = q_t \text{ESS}(\{W_{t-1}^{(J)}\}_{J=1}^N, \epsilon_t), \tag{3.4}$$

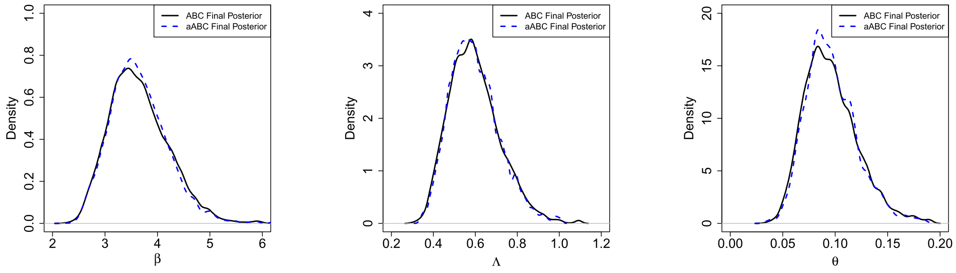
where q_t is some pre-selected quantile which varies between 0 and 1. Numminen et al. (2013) had to adjust this to work for their setting with four tolerances. We note that our aABC-PMC approach does not require the specification of a quantile q_t, nor other parameters such as the number M of simulations performed for each particle, the minimal effective sample size threshold below which a resampling of particles is performed, $\text{nb}_{\text{threshold}}$, and the final tolerance level, ϵ_{final}. Further details on the ABC-SMC algorithm and discussions on how to properly select its required parameters can be found in Del Moral et al. (2012).

The prior distributions for the three parameters of interest are $\beta \sim \text{Unif}(0, 11)$, $\Lambda \sim \text{Unif}(0, 2)$, and $\theta \sim \text{Unif}(0, 1)$. Starting from the second iteration of the ABC-PMC algorithm, proposals are perturbed with Gaussian kernels, using the specifications of Beaumont et al. (2009). The desired particle sample size was set at $N = 10,000$. For the aABC-PMC algorithm, the initial number of draws sampled from the prior distributions is set to $N_{\text{init}} = 5 \times 10,000$, in order to appropriately explore the parameter space.

The results of the analysis are summarized in Table 6, where the proposed adaptive rule for selecting the quantile performs better than the ABC-SMC algorithm both in terms of the computational time (3 days and 5 hours vs. 4 days and 12 hours using a cluster computer) and the total number of draws ($1,085,696$ draws vs. $2,199,760$ draws). Because the proposed sampling procedure stops after $t = 4$ iterations, the expensive forward model is used fewer times, achieving final posterior distributions in a shorter amount of time. We note that the number of particles sampled in the first iteration has an important role in the performance of the algorithm. In fact, having sampled from the priors $\text{D}_1 = 50,000$ particles allowed the aABC-PMC algorithm to initiate with a smaller tolerance $\epsilon_1 = 1.26$ compared to the ABC-SMC algorithm ($\epsilon_1 = 3.91$ by fixing $\text{D}_1 = 10,000$ particles).

	Numminen et al. (2013)			aABC-PMC		
t	ϵ_t	D_t	t	ϵ_t	q_t	D_t
1	3.91	10,000	1	1.26		50,000
2	1.94	121,374	2	1.04	0.19	154,142
3	1.28	277,997	3	0.97	0.31	489,239
4	0.99	572,007	4	0.93	0.74	792,315
5	0.84	1,218,760				
Total		2,199,760				1,085,696

Table 6: Bacterial infection in day care centers results. The number of draws needed in each iteration to reach $N = 10,000$ accepted values for the ABC-SMC as presented in Gutmann and Corander (2016) and the proposed aABC-PMC algorithm. In the aABC-PMC algorithm also the quantile automatically selected through the iterations is available. The procedure stopped once the quantile $q_5 = 0.993$ was calculated. For the ABC-SMC algorithm a total of $2,199,760$ (4 days and 12 hours on a cluster with 200 cores) draws are required, while our aABC-PMC takes $1,085,696$ draws (3 days and 5 hours on a cluster with 200 cores).



(a) Final ABC posteriors for β. (b) Final ABC posteriors for λ. (c) Final ABC posteriors for θ.

Figure 8: Bacterial infection in day care centers ABC posteriors. Comparison between the final posterior distributions for β, λ and θ obtained by using Del Moral et al. (2012)'s adaptive selection of the tolerances (solid black) and by using the aABC-PMC algorithm (dashed blue).

The ABC posteriors for the three parameters β, Λ and θ for the tolerances of Numminen et al. (2013) selected by using ABC-SMC and the proposed aABC-PMC approach are displayed in Figures 8. We note that the final tolerance from Numminen et al. (2013), $\epsilon_5 = 0.83$, is slightly smaller than the final tolerance of aABC-PMC, $\epsilon_4 = 0.93$, but the posteriors for β, Λ and θ are comparable, with the Hellinger distances respectively equals to $H_{\text{dist}} = 0.079, 0.097, 0.093$.[5]

[5] The Hellinger distances are calculated between the ABC posterior distributions found by Numminen et al. (2013) and the corresponding ABC posterior distributions retrieved with our aABC-PMC approach.

4 Concluding Remarks

The ABC-PMC algorithm of Beaumont et al. (2009) has lead to great improvements over the basic ABC rejection algorithm in terms of sampling efficiency. However, to use ABC-PMC it is necessary to define a sequence of tolerances along with the total number of iterations. We propose an approach leveraging ratio estimating methods for shrinking the tolerances by adaptively selecting a suitable quantile based on the progression of the estimated ABC posteriors. The proposed adjustment to the existing algorithm is shown to be able to deal with the possible presence of local modes and shrinks the tolerance in such a way that fewer draws are needed from the forward model compared to commonly used techniques for selecting the tolerances. A simple criterion for stopping the algorithm based on the behavior of the sequential ABC posterior distribution is also presented. The empirical performance in the examples considered suggests the proposed aABC-PMC algorithm is superior to the other options considered in terms of computational time and the number of draws from the forward model. Based on the computational experiments we envisage that the proposed aABC-PMC algorithm performs generally well when dealing with small to moderate dimensional problems for which the original ABC-PMC algorithm was developed. It remains as a challenge for the future research to generalize these samplers to higher dimensional models.

Supplementary Material

Supplementary Material of “Adaptive Approximate Bayesian Computation Tolerance Selection” (DOI: 10.1214/20-BA1211SUPP; .pdf).

References

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. I. (2003). “An introduction to MCMC for machine learning.” *Machine learning*, 50(1–2): 5–43. 6

Beaumont, M. A. (2010). “Approximate Bayesian computation in evolution and ecology.” *Annual review of ecology, evolution, and systematics 41*, 96: 379–406. MR3939526. doi: https://doi.org/10.1146/annurev-statistics-030718-105212. 1

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009). “Adaptive approximate Bayesian computation.” *Biometrika*, 96(4): 983–990. MR2767283. doi: https://doi.org/10.1093/biomet/asp052. 2, 3, 4, 11, 20, 22

Bickel, S., Brückner, M., and Scheffer, T. (2007). “Discriminative learning for differing training and test distributions.” In *Proceedings of the 24th international conference on Machine learning*, 81–88. ACM. 8

Blum, M., Nunes, M., Prangle, D., and Sisson, S. (2013). “A comparative review of dimension reduction methods in approximate Bayesian computation.” *Statistical Science*, 28(2): 189–208. MR3112405. doi: https://doi.org/10.1214/12-sts406. 2

Blum, M. G. (2010). "Approximate Bayesian Computation: A nonparametric perspective." *Journal of American Statistical Association*, 105(491): 1178–1187. MR2752613. doi: https://doi.org/10.1198/jasa.2010.tm09448. 2

Bonassi, F. and West, M. (2015). "Sequential Monte Carlo with Adaptive Weights for Approximate Bayesian Computation." *Bayesian Analysis*, (10): 171–187. MR3420901. doi: https://doi.org/10.1214/14-BA891. 13

Bregman, L. M. (1967). "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming." *USSR computational mathematics and mathematical physics*, 7(3): 200–217. MR0215617. 8

Brent, R. P. (2013). *Algorithms for minimization without derivatives*. Courier Corporation. MR0339493. 8

Brooks-Pollock, E., Becerra, M. C., Goldstein, E., Cohen, T., and Murray, M. B. (2011). "Epidemiologic inference from the distribution of tuberculosis cases in households in Lima, Peru." *Journal of Infectious Diseases*, 203(11): 1582–1589. 19

Cameron, E. and Pettitt, A. N. (2012). "Approximate Bayesian Computation for Astronomical Model Analysis: A Case Study in Galaxy Demographics and Morphological Transformation at High Redshift." *Monthly Notices of the Royal Astronomical Society*, 425: 44–65. 1

Cisewski-Kehe, J., Weller, G., Schafer, C., et al. (2019). "A preferential attachment model for the stellar initial mass function." *Electronic Journal of Statistics*, 13(1): 1580–1607. MR3939305. doi: https://doi.org/10.1214/19-ejs1556. 1, 2, 4, 5

Corander, J., Fraser, C., Gutmann, M. U., Arnold, B., Hanage, W. P., Bentley, S. D., Lipsitch, M., and Croucher, N. J. (2017). "Frequency-dependent selection in vaccine-associated pneumococcal population dynamics." *Nature ecology & evolution*, 1(12): 1950. 1

Cornuet, J., Santos, F., Beaumont, M., Robert, C., Marin, J., Balding, D., Guillemaud, T., and Estoup, A. (2008). "Inferring population history with DIY ABC: a user-friendly approach to Approximate Bayesian Computation." *Bioinformatics*. MR2767283. doi: https://doi.org/10.1093/biomet/asp052. 2

Csilléry, K., Blum, M. G., Gaggiotti, O. E., and François, O. (2010). "Approximate Bayesian Computation (ABC) in practice." *Trends in ecology & evolution*, 25(7): 410–418. 2

Del Moral, P., Doucet, A., and Jasra, A. (2012). "An adaptive sequential Monte Carlo method for approximate Bayesian computation." *Statistics and Computing*, 22(5): 1009–1020. MR2950081. doi: https://doi.org/10.1007/s11222-011-9271-y. 2, 4, 13, 18, 20, 21

Drovandi, C. C. and Pettitt, A. N. (2011). "Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. Biometrics." *Statistics and Computing*, 67(1): 225–233. MR2898834. doi: https://doi.org/10.1111/j.1541-0420.2010.01410.x. 2

Fearnhead, P. and Prangle, D. (2012). "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation." *Journal of the Royal Statistical Society Series B*, 74(3): 419–474. MR2925370. doi: https://doi.org/10.1111/j.1467-9868.2011.01010.x. 2

Gelman, A., Carln, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2014). *Bayesian Data Analysis*. Chapman & Hall. MR3235677. 10

Gretton, A., Smola, A. J., Huang, J., Schmittfull, M., Borgwardt, K. M., and Schölkopf, B. (2009). "Covariate shift by kernel mean matching." 8

Gutmann, M. U. and Corander, J. (2016). "Bayesian optimization for likelihood-free inference of simulator-based statistical models." *The Journal of Machine Learning Research*, 17(1): 4256–4302. MR3555016. 2, 19, 21

Hesterberg, T. C. (1988). "Advances in importance sampling." Ph.D. thesis, Stanford University. MR2637036. 6

Hido, S., Tsuboi, Y., Kashima, H., Sugiyama, M., and Kanamori, T. (2011). "Statistical outlier detection using direct density ratio estimation." *Knowledge and information systems*, 26(2): 309–336. 8

Hoti, F., Erästö, P., Leino, T., and Auranen, K. (2009). "Outbreaks of Streptococcus pneumoniae carriage in day care cohorts in Finland–implications for elimination of transmission." *BMC infectious diseases*, 9(1): 102. 19

Ishida, E., Vitenti, S., Penna-Lima, M., Cisewski, J., de Souza, R., Trindade, A., Cameron, E., et al. (2015). "cosmoabc: Likelihood-free inference via Population Monte Carlo Approximate Bayesian Computation." *Astronomy & Computing*, 13: 1–11. 1, 4, 9

Järvenpää, M., Gutmann, M., Vehtari, A., and Marttinen, P. (2016). "Gaussian process modeling in approximate Bayesian computation to estimate horizontal gene transfer in bacteria." *arXiv preprint* arXiv:1610.06462. MR3875699. doi: https://doi.org/10.1214/18-AOAS1150. 2

Jennings, E. and Madigan, M. (2016). "astroABC: An Approximate Bayesian Computation Sequential Monte Carlo sampler for cosmological parameter estimation." *Astronomy and Computing*. 2

Jennings, E., Wolf, R., and Sako, M. (2016). "A new approach for obtaining cosmological constraints from type IA supernovae using approximate Bayesian computation." *Astronomy and Computing*. 2

Joyce, P. and Marjoram, P. (2008). "Approximately sufficient statistics and Bayesian computation." *Statistical Applications in Genetics and Molecular Biology*, 7(1): 1–16. MR2438407. doi: https://doi.org/10.2202/1544-6115.1389. 2

Julier, S., Uhlmann, J., and Durrant-Whyte, H. F. (2000). "A new method for the nonlinear transformation of means and covariances in filters and estimators." *IEEE Transactions on automatic control*, 45(3): 477–482. MR1762859. doi: https://doi.org/10.1109/9.847726. 4

Lenormand, M., Jabot, F., and Deuant, G. (2013). "Adaptive approximate Bayesian computation for complex models." *Computational Statistics*, 6(28): 2777–2796. MR3141363. doi: https://doi.org/10.1007/s00180-013-0428-3. 4

Lintusaari, J., Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. (2017). "Fundamentals and recent developments in approximate Bayesian computation." *Systematic biology*, 66(1): e66–e82. 9

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). "Approximate Bayesian computational methods." *Statistics and Computing*, 22(6): 1167–1180. MR2992292. doi: https://doi.org/10.1007/s11222-011-9288-2. 2

McKinley, T., Cook, A., and Deardon, R. (2009). "Inference in epidemic models without likelihoods." *The International Journal of Biostatistics*, 171(5). MR2533810. doi: https://doi.org/10.2202/1557-4679.1171. 1, 4

Numminen, E., Cheng, L., Gyllenberg, M., and Corander, J. (2013). "Estimating the transmission dynamics of Streptococcus pneumoniae from strain prevalence data." *Biometrics*, 69(3): 748–757. MR3106603. doi: https://doi.org/10.1111/biom.12040. 1, 2, 4, 5, 10, 19, 20, 21

Pritchard, J. K., Seielstad, M. T., and Perez-Lezaun, A. (1999). "Population Growth of Human Y Chromosomes: A study of Y Chromosome Microsatellites." *Molecular Biology and Evolution*, 16(12): 1791–1798. 1

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/. 9

Ratmann, O., Camacho, A., Meijer, A., and Donker, G. (2013). "Statistical modelling of summary values leads to accurate Approximate Bayesian computations." Unpublished. 2

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media. MR1707311. doi: https://doi.org/10.1007/978-1-4757-3071-5. 6

Rubin, D. B. (1984). "Bayesianly justifiable and relevant frequency calculations for the applied statistician." *The Annals of Statistics*, 12(4): 1151–1172. MR0760681. doi: https://doi.org/10.1214/aos/1176346785. 1

Schafer, C. M. and Freeman, P. E. (2012). *Statistical Challenges in Modern Astronomy V*, chapter 1, 3–19. Lecture Notes in Statistics. Springer. 1

Silk, D., Filippi, S., and Stumpf, M. (2013). "Optimizing threshold-schedules for sequential approximate Bayesian computation: applications to molecular systems." *Statistical Applications in Genetics and Molecular Biology*, 5(12): 603–618. MR3108049. doi: https://doi.org/10.1515/sagmb-2012-0043. 2, 3, 4, 10, 14, 15, 16, 17, 18

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. CRC press. MR0848134. doi: https://doi.org/10.1007/978-1-4899-3324-9. 10

Silverman, B. W. (2018). *Density estimation for statistics and data analysis*. Routledge. 7, 10

Simola, U., Cisewski-Kehe, J., Gutmann, M. U., and Corander, J. (2020). "Supplementary Material of "Adaptive Approximate Bayesian Computation Tolerance Selection"." *Bayesian Analysis*. doi: `https://doi.org/10.1214/20-BA1211SUPP`. 10

Simola, U., Pelssers, B., Barge, D., Conrad, J., and Corander, J. (2019). "Machine learning accelerated likelihood-free event reconstruction in dark matter direct detection." *Journal of Instrumentation*, 14(03): P03004. 1, 2, 4

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). "Sequential Monte Carlo without likelihoods." *Proceedings of the National Academy of Science*, 104(6): 1760–1765. MR2301870. doi: `https://doi.org/10.1073/pnas.0607208104`. 2, 4, 10, 11, 12

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. (2008). "Direct importance estimation with model selection and its application to covariate shift adaptation." In *Advances in neural information processing systems*, 1433–1440. 8

Sugiyama, M., Suzuki, T., and Kanamori, T. (2010). "Density Ratio Estimation: A Comprehensive Review (Statistical Experiment and Its Related Topics)." MR2895762. doi: `https://doi.org/10.1017/CBO9781139035613`. 8

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density ratio estimation in machine learning*. Cambridge University Press. 8

Tavaré, S., Balding, D. J., Griffiths, R., and Donnelly, P. (1997). "Inferring coalescence times from DNA sequence data." *Genetics*, 145: 505–518. 1

Thornton, K. and Andolfatto, P. (2006). "Inference in epidemic models without likelihoods." *Genetics*, 172: 1607–1619. 1

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. H. (2009). "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems." *Journal of the Royal Society, Interface / the Royal Society*, 6(31): 187–202. 1, 2, 4, 10

Vestrheim, D. F., Høiby, E. A., Aaberge, I. S., and Caugant, D. A. (2010). "Impact of a pneumococcal conjugate vaccination program on carriage among children in Norway." *Clinical and Vaccine Immunology*, 17(3): 325–334. 19

Vestrheim, D. F., Løvoll, Ø., Aaberge, I. S., Caugant, D. A., Høiby, E. A., Bakke, H., and Bergsaker, M. R. (2008). "Effectiveness of a 2+ 1 dose schedule pneumococcal conjugate vaccination programme on invasive pneumococcal disease among children in Norway." *Vaccine*, 26(26): 3277–3281. 19

Weyant, A., Schafer, C., and Wood-Vasey, W. M. (2013). "Likelihood-free cosmological inference with type Ia supernovae: approximate Bayesian computation for a complete treatment of uncertainty." *The Astrophysical Journal*, 764: 116. 1, 4

Acknowledgments

The authors thank IT-University of Helsinki and Yale's Center for Research Computing for the computational resources provided to execute the analyses of the present work. U. Simola was supported by the Academy of Finland grant no. 1316602. J. Corander was supported by the ERC grant no. 742158. The authors are grateful for the comments and feedback from the anonymous associate editor and referees, which significantly helped to improve this work.

Bayesian Analysis (0000) **00**, Number 0, pp. 1

Supplementary Material of "Adaptive Approximate Bayesian Computation Tolerance Selection"

Umberto Simola[*], Jessi Cisewski-Kehe[†], Michael U. Gutmann[‡], Jukka Corander[*,§]

Appendix A: Lotka–Volterra Model

The Lotka-Volterra model (Lotka, 1925; Volterra, 1927) describes two interacting populations and in their original ecological setting representing predators and prey. Since the true posterior distribution is not available, the ABC posteriors from Toni et al. (2009), are used as benchmarks. The interaction between the predators (y) and the prey (x) is defined by the following two differential equations:

$$\frac{dx}{dt} = ax - xy \tag{A.1}$$

$$\frac{dy}{dt} = bxy - y, \tag{A.2}$$

where the parameters of interests are a and b.

Inference on this model using ABC was considered originally in Toni et al. (2009), and we use their same model, dataset, summary statistic and distance function in order to test the performance of our proposed aABC-PMC algorithm. The same configuration was also used by Järvenpää et al. (2016) in order to test their ABC extensions. The dataset for the analysis $(x_{\text{obs}}, y_{\text{obs}})$ was obtained by using Eq. (A.1) and Eq. (A.2) with input values $a = 1$ and $b = 1$. The sample size is $n = 8$ for the two species, and the distance function for comparing real data, (x, y), with the simulated data, $(x_{\text{sim}}, y_{\text{sim}})$, is defined as:

$$d[(x_{\text{obs}}, y_{\text{obs}}), (x_{\text{sim}}, y_{\text{sim}})] = \sum_{i=1}^{n} \left[(x_{i,\text{obs}} - x_{i,\text{sim}})^2 + (y_{i,\text{obs}} - y_{i,\text{sim}})^2 \right]. \tag{A.3}$$

The forward model solves the deterministic set of differential equations defined above for x and y, then Gaussian noise is added from $N(0, 0.5^2)$ to obtain a simulated dataset, $(x_{\text{sim}}, y_{\text{sim}})$. The prior for both a and b is a Uniform distribution with the range $[-10, 10]$.

[*] Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland umberto.simola@helsinki.fi

[†] Department of Statistics and Data Science, Yale University, New Haven, CT, USA jessica.cisewski@yale.edu

[‡] School of Informatics, University of Edinburgh, Edinburgh, United Kingdom michael.gutmann@ed.ac.uk

[§] Department of Biostatistics, University of Oslo, Oslo, Norway jukka.corander@medesin.uio.fi

DOI: 0000

For $t > 1$ of the ABC-PMC algorithm, rather than using the Gaussian perturbation kernel as displayed in Algorithm 1, the selected particles are additively perturbed by draws from an $U(-0.1, 0.1)$, in order to avoid degeneracy of the samplers. The importance weights are calculated by taking into account of the fact that a Uniform kernel is used. In Toni et al. (2009), the series of tolerances is manually selected as listed in Table 1.

For the proposed aABC-PMC procedure, the initial number of draws sampled from the prior distributions is set at $N_{\text{init}} = 5 \times 1,000$ in order to sufficiently explore the parameter space. A comparison between the two procedures is done as before, in terms of the computational time and the total number of draws, with the results shown in Table 1. The results are based on 21 independent runs and the table includes the values for the run that produced the median number of total draws. Although aABC-PMC requires more iterations, the proposed procedure outperforms Toni et al. (2009)'s implementation of ABC-PMC in terms of total number of draws and computational time.

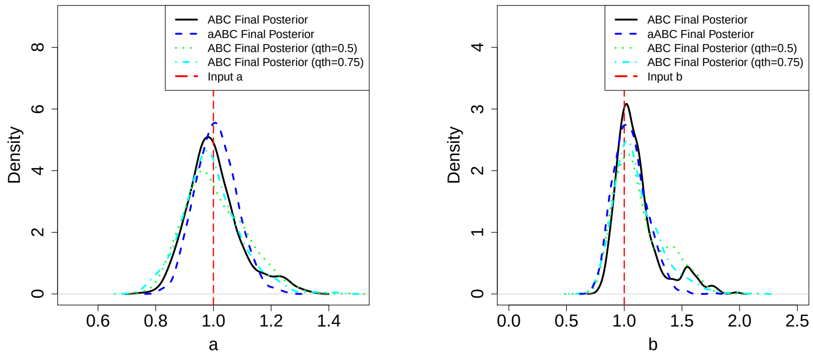
The ABC posteriors for parameters a and b for the manually-selected tolerances of Toni et al. (2009) and the proposed aABC-PMC approach are displayed in Figure 1. Additionally, ABC posteriors are displayed for two quantile-selected tolerances (0.5 and 0.75) for comparison. For each iteration $t > 1$ of the ABC–PMC procedure, in the first analysis the quantile of level 0.5 (i.e. the median) of the distances of the accepted particles from iteration $t - 1$ is used in order to provide the tolerance for the next iteration, while in the second analysis the quantile of level 0.75 of the distances of the accepted particles from iteration $t-1$ is used. The quantile-selected tolerance algorithms are stopped once the final number of draws needed by the aABC–PMC is reached. The ABC–PMC algorithm that uses a quantile-selected tolerance of 0.5 is stopped after $T = 11$ iterations, with a final tolerance equal to $\epsilon_{11} = 5.57$. When a quantile of 0.75 is used, $T = 21$ iterations are performed by the ABC–PMC algorithm and the final tolerance is $\epsilon_{21} = 5.43$. Both final tolerances are higher than the one adaptively selected by the proposed aABC-PMC approach, which is $\epsilon_9 = 4.76$. As shown in Fig. 1, when using the ABC–PMC algorithm with quantile-selected tolerances of 0.5 and 0.75, less informative posterior distributions are obtained than the posterior distributions obtained with the aABC–PMC approach.

The series of tolerances for the proposed aABC-PMC algorithm is adaptively selected in such a way that the forward model is drawn from fewer times than the manually-selected tolerances from Toni et al. (2009) and the common quantile-selected approaches. Though the final tolerance from Toni et al. (2009), $\epsilon_5 = 4.23$, is smaller than the final tolerance of aABC-PMC, $\epsilon_9 = 4.76$, the posteriors for a ($H_{\text{dist}} = 0.25$) and b ($H_{\text{dist}} = 0.34$) are comparable (Figure 1)[1].

[1] The Hellinger distances are calculated between the ABC posterior distributions found by Toni et al. (2009) and those retrieved with the proposed aABC-PMC approach.

Toni et al. (2009)			aABC-PMC			
t	ϵ_t	D_t	t	ϵ_t	q_t	D_t
1	30	3,541	1	27.82		5,000
2	16	48,402	2	25.56	0.65	2,734
3	6	52,471	3	22.59	0.50	3,645
4	5	25,097	4	18.52	0.33	5,853
5	4.3	47,521	5	10.19	0.09	12,429
			6	6,70	0.17	12,146
			7	5,73	0,49	12,295
			8	4,99	0.51	20,890
			9	4.76	0.78	22,870
Total		177,032				97,862

Table 1: Lotka–Volterra model results. The number of draws needed in each iteration to reach $N = 1,000$ accepted values for the ABC-PMC algorithm of Toni et al. (2009) and the aABC-PMC algorithm. (The displayed results were obtained by running the procedure 21 times and using the run that produced the median number of total draws.) The aABC-PMC algorithm quantiles automatically selected through the iterations are listed under q_t. The procedure stopped once the quantile $q_{10} = 0.999$ was calculated. For the ABC–PMC algorithm a total of $177,032$ ($1,074$ sec.) draws were required, while the aABC-PMC took $97,862$ (548 sec.) draws overall.



(a) Final ABC posteriors for a.

(b) Final ABC posteriors for b.

Figure 1: Lotka–Volterra posterior distributions. Comparison between the final posterior distributions for (a) a and (b) b obtained using Toni et al. (2009)'s manually selecting the tolerances (black), by fixing the quantile equal to .50 (green) and 0.75 (cyan), and by using the aABC-PMC (blue).

Appendix B: Gaussian Mixture Model: Sedki et al. (2012)'s Results

Thanks to Sedki et al. (2012), a comparison for this example with the method proposed by Del Moral et al. (2012) was possible. (We also discuss the Del Moral et al. (2012) method in Sec. 3.3 of the main text.) The desired particle sample size was fixed at N=10,000 and their final threshold was $\epsilon_{final} = 0.09$. Using the same ϵ_{final}, the results of the analyses for the aABC-PMC approach are based on 21 independent runs with the same dataset $y_{\text{obs}} = 0$ and $k = 5$ (i.e. $N_{\text{init}} = 50000$). The run with the median number of total draws used the simulator $399,577$ times. For the ABC–SMC algorithm, we set the parameters required by the algorithm by using the specifications privided in Sedki et al. (2012): $N = 10000$, $\alpha = 0.95$, $M = 50$, $\text{nb}_{\text{threshold}} = N/2$ and $\epsilon_{\text{final}} = 0.09$ (these parameters are defined in Sec. 3.3 of the main text, while further details can be found in Del Moral et al. (2012)). We performed the analyses on 21 independent runs with the same dataset, $y_{\text{obs}} = 0$, and considered that run the produced the median number of total draws. For that run the total number of draws was equal to $3,088,550$. We note that the number of draws reported by Sedki et al. (2012) in their paper is $4.6 \cdot 10^6$ (Table 1, Sedki et al. 2012).

References

Del Moral, P., Doucet, A., and Jasra, A. (2012). "An adaptive sequential Monte Carlo method for approximate Bayesian computation." *Statistics and Computing*, 22(5): 1009–1020. 4

Järvenpää, M., Gutmann, M., Vehtari, A., and Marttinen, P. (2016). "Gaussian process modeling in approximate Bayesian computation to estimate horizontal gene transfer in bacteria." *arXiv preprint arXiv:1610.06462*. 1

Lotka, A. (1925). *Elements of physical biology*, volume 1. Baltimore, MD: Williams & Wilkins Co. 1

Sedki, M., Pudlo, P., Marin, J.-M., Robert, C. P., and Cornuet, J.-M. (2012). "Efficient learning in ABC algorithms." *arXiv preprint arXiv:1210.1388*. 4

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. H. (2009). "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems." *Journal of the Royal Society, Interface / the Royal Society*, 6(31): 187–202. 1, 2, 3

Volterra, V. (1927). *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi*. C. Ferrari. 1