Rare event ABC-SMC

Ivis Kerama, Thomas Thorne and Richard G. Everitt

7th November 2022

Abstract

Approximate Bayesian computation (ABC) is a well-established family of Monte Carlo methods for performing approximate Bayesian inference in the case where an “implicit” model is used for the data: when the data model can be simulated, but the likelihood cannot easily be pointwise evaluated. A fundamental property of standard ABC approaches is that the number of Monte Carlo points required to achieve a given accuracy scales exponentially with the dimension of the data. Prangle et al. (2018) proposes a Markov chain Monte Carlo (MCMC) method that uses a rare event sequential Monte Carlo (SMC) approach to estimating the ABC likelihood that avoids this exponential scaling, and thus allows ABC to be used on higher dimensional data. This paper builds on the work of Prangle et al. (2018) by using the rare event SMC approach within an SMC algorithm, instead of within an MCMC algorithm. The new method has a similar structure to SMC$^2$ (Chopin et al., 2013), and requires less tuning than the MCMC approach. We demonstrate the new approach, compared to existing ABC-SMC methods, on a toy example and on a duplication-divergence random graph model used for modelling protein interaction networks.

1 Introduction

1.1 Approximate Bayesian computation

This paper concerns Bayesian inference of the parameters $\theta$ of “implicit” models: those where the likelihood cannot be evaluated pointwise at $\theta$. This problem is encountered in a number of different fields, including epidemiology, ecology, economics, particle physics, cosmology and genetics. Suppose we wish to use a model $f_{\theta}(y)$ for observed data $y$ that depends on parameters $\theta$, and wish to estimate the parameters of the model (sometimes known as "calibrating" the model) using Bayesian inference.

Our aim in performing this calibration depends on the problem we are tackling. Three possible aims are:

- **inference** of the parameters $\theta$ or functions of $\theta$, e.g. the reproduction number $R_0$ in an epidemic;
- **prediction** of possible future data using the calibrated model, e.g. number of new infections in an epidemic;
- **model criticism or comparison** to help us understand if $f$ is a realistic model, either in terms of prediction, or in terms of describing the observed data.

In each situation the posterior distribution $\pi(\theta | y)$ of parameters $\theta$ plays a central role. For many commonly used choices of $f$, we can evaluate $f_{\theta}(y)$ pointwise at $\theta$. Inference then usually proceeds using a Monte Carlo method for simulating from the posterior distribution $\pi(\theta | y) \propto p(\theta)f_{\theta}(y)$, for example importance sampling (IS), Markov chain Monte Carlo (MCMC) or sequential Monte Carlo (SMC). The standard versions of these algorithms all rely on being able to evaluate the prior and likelihood pointwise at $\theta$, thus cannot be directly implemented for implicit models. This led to the development of approximate Bayesian computation (ABC), which replaces the likelihood with an assessment as to whether $\theta$ is plausible under the posterior by simulating $x \sim f_{\theta}({\cdot})$ and checking if statistics of $x$ are close to statistics of the observed data.
out in Del Moral. This integral is intractable in general, hence the use of a Monte Carlo estimator. As pointed
simulations from some circumstances it may be more efficient to take the sample average of an estimate of the true ABC likelihood since it uses only a single Monte Carlo point, and in
is normalising a kernel with tolerance (or bandwidth) $\epsilon$ for
$\sum_{i=1}^{N} \frac{p(\theta^{i}) q(\theta^{i-1} \mid \theta^{i}) P_{\epsilon}(y \mid x^{i})}{p(\theta^{i-1}) q(\theta^{i-1} \mid \theta^{i-1}) P_{\epsilon}(y \mid x^{i-1})};$.

$\text{for } i = 1 : N \text{ do}$
4. Simulate $\theta^{*} \sim q(\cdot \mid \theta^{i-1});$
5. Simulate $x^{*} \sim f_{\theta^{*}}(\cdot);$
6. Let $(\theta^{i}, x^{i}) = (\theta^{*}, x^{*})$ with probability

$$1 + \frac{p(\theta^{*}) q(\theta^{i-1} \mid \theta^{*}) P_{\epsilon}(y \mid x^{*})}{p(\theta^{i-1}) q(\theta^{i} \mid \theta^{i-1}) P_{\epsilon}(y \mid x^{i-1})};$$

7. Else let $(\theta^{i}, x^{i}) = (\theta^{i-1}, x^{i-1});$
end

One way of understanding ABC-MCMC is as a “pseudo-marginal” method (Beaumont, 2003; Andrieu and Roberts, 2009). This class of methods uses an unbiased approximation to the likelihood in place of an exact likelihood in a Metropolis-Hastings algorithm; it is shown in Andrieu and Roberts (2009) that the limiting distribution of the $\theta$-points generated by such an algorithm is the same as if the exact likelihood were used. In the case of ABC–MCMC, the likelihood estimate at a point $\theta$ is simply $P_{\epsilon}(y \mid x)$, where $x \sim f_{\theta}(\cdot)$. This is a Monte Carlo estimate of what we call the “true” ABC likelihood

$$l(y \mid \theta) = \int_{x} P_{\epsilon}(y \mid x) f_{\theta}(x) \, dx.$$

This integral is intractable in general, hence the use of a Monte Carlo estimator. As pointed out in Del Moral et al. (2012), the estimated ABC likelihood $P_{\epsilon}(y \mid x)$ is a very high variance estimate of the true ABC likelihood since it uses only a single Monte Carlo point, and in some circumstances it may be more efficient to take the sample average of $P_{\epsilon}(y \mid \cdot)$ for several simulations from $f_{\theta}(\cdot)$. In this case, for $N_{x}$ points simulated from $f_{\theta}(\cdot)$, the estimated likelihood is

$$\hat{l}(y \mid \theta) = \frac{1}{N_{x}} \sum_{n=1}^{N_{x}} P_{\epsilon}(y \mid x^{n}). \quad (1)$$

For expository purposes, it is useful to view our Monte Carlo estimator as an IS estimator of the normalising constant $\int_{x} P_{\epsilon}(y \mid x) f_{\theta}(x) \, dx$ of the unnormalised target distribution $P_{\epsilon}(y \mid x) f_{\theta}(x)$ when using proposal $f_{\theta}(x)$. The importance sampling estimator is unbiased, and

\begin{algorithm}[h]
\caption{ABC-MCMC}
\begin{algorithmic}[1]
\Input{$N \geq 0$}
\Output{$\{\theta^{i}, x^{i}\}_{i=0}^{N}$}
1 Initialise $\theta^{0}$;  
2 Simulate $x^{*} \sim f_{\theta^{0}}(\cdot)$;  
\For{$i = 1 : N$}
4 Simulate $\theta^{*} \sim q(\cdot \mid \theta^{i-1});$
5 Simulate $x^{*} \sim f_{\theta^{*}}(\cdot);$
6 Let $(\theta^{i}, x^{i}) = (\theta^{*}, x^{*})$ with probability

$$1 + \frac{p(\theta^{*}) q(\theta^{i-1} \mid \theta^{*}) P_{\epsilon}(y \mid x^{*})}{p(\theta^{i-1}) q(\theta^{i} \mid \theta^{i-1}) P_{\epsilon}(y \mid x^{i-1})};$$

7 Else let $(\theta^{i}, x^{i}) = (\theta^{i-1}, x^{i-1});$
\EndFor
\end{algorithmic}
\end{algorithm}
its variance of an depends on the distance between the proposal and the target (Agapion et al., 2017). Andrieu and Roberts (2009) tells us that the unbiasedness of the estimated likelihood will result in ABC-MCMC having the same invariant distribution as if we had used the true ABC likelihood.

A likelihood estimator with a higher variance usually results in a less efficient MCMC algorithm. For the estimated ABC likelihood, we have that the distance between target $P_l(y \mid x) f_\theta(x)$ and proposal $f_\theta(x)$ (and hence the variance of the estimator), will tend to be larger when the dimension of $y$ is higher and when $\epsilon$ is smaller. The dimension of $x$ has a particularly large impact: the variance of the estimator increases exponentially with the dimension of $y$. This is the reason that the raw data is rarely used in ABC: common practice is to reduce the dimension of the data by using summary statistics, and thus reduce the variance of the likelihood estimator. If the statistics are not sufficient, this results in a different posterior to using the full data: the variance has been reduced at the cost of introducing bias. A similar tradeoff is made when choosing an appropriate $\epsilon$. The ABC likelihood will only result in a posterior equal to the true posterior as $\epsilon \to 0$, however this is the case where the variance of the likelihood estimator is at its highest. In practice some $\epsilon > 0$ is used, chosen such that the variance of estimates from ABC-MCMC are not too high, although at the cost of introducing a bias.

The methodology used in this paper is focussed on trying to avoid, as far as possible, the need to reduce the dimension of the data by choosing summary statistics, whilst avoiding a high variance likelihood estimator. The approach we use is described in the following section.

### 1.3 Unpacking the black box simulator and using rare event SMC

Many methods for estimating the likelihood can be seen as different ways of approaching this bias-variance tradeoff: for example some approaches assume some parametric or non-parametric model for joint or conditional distributions of $\theta, y$, with the aim of reducing variance whilst attempting to introduce little additional bias (Cranmer et al., 2020). Our method is one in an alternative class of approaches that make use of a decomposition of the simulator that is available in many situations. We rewrite the simulator as a deterministic transformation $H$ of the parameter $\theta$ and a random vector $u$ that is drawn from a tractable distribution $\phi(\cdot \mid \theta)$. We believe this “reparameterisation trick” was first introduced in the ABC context in Andrieu et al. (2012); it has been used in a number of papers since (e.g. Meeds and Welling, 2015; Forneron and Ng, 2016; Moreno et al., 2016; Graham and Storkey, 2017). The important property of this decomposition is that it allows us to replace the target

$$P_e(y \mid x) f_\theta(x)$$

with

$$P_e(y \mid H(u, \theta)) \phi(u \mid \theta).$$

Although we are still using an ABC-style likelihood, the intractable $f_\theta$ is no longer present. This presents the possibility of exploring other ways of moving around the $\theta, u$-space, for example the Hamiltonian Monte Carlo (HMC) in Graham and Storkey (2017)). To understand the possible benefits of using this approach, it is useful to think again about the reasons for the high variance of the standard ABC likelihood: in this case the $u$ variable is drawn independently at each iteration, from a distribution that does not depend on $y$. We might hope that by tailoring $u$ specific to $\theta$ and $y$ - choosing the random vector $u$ such that our likelihood simulations conditional on $\theta$ are close to $y$ - we make an efficiency gain.

In this paper we follow Prangle et al. (2018) in using a "rare event" SMC algorithm for simulating from the conditional distribution of $u \mid \theta, y$ in order to tailor $u$ to $\theta$ and $y$. In fact, this simulation of $u$ conditional on $y$ and $\theta$ is used to estimate the ABC likelihood $l(y \mid \theta) = \int_u P_e(y \mid H(u, \theta)) \phi(u \mid \theta) du$, with a lower variance than the standard approach, which we presented above as importance sampling. This is exactly the marginal particle MCMC algorithm (Andrieu et al., 2010) in the case of ABC. Theoretical results about SMC tell us that we expect the variance of this likelihood estimator to scale more favourably with the dimension of $y$ (which for simplicity we assume is the same as the dimension of $u$) than the standard IS based estimator. Prangle et al. (2018) studies theoretically the computation required to accept a proposed point.
using the IS estimator compared to the SMC as $d \to \infty$, and finds a cost of $O\left(\epsilon^{-d}\right)$ for the IS approach, compared to $O\left(d^2 \log \left(\epsilon^{-1}\right)\right)$ for SMC (also see Beskos et al. (2014)).

This approach should not be confused with ABC-SMC algorithms, which are described later in this paper. These algorithms are focused on using SMC to explore $\theta$-space (sometimes based on a justification of exploring the joint $(\theta,u)$-space). The SMC method employed by Prangle et al. (2018) is instead exploring the conditional distribution of $u \mid \theta, y$. When the ABC kernel is a uniform distribution, such an SMC algorithm has been explored thoroughly in the work of Cérou et al. (2012), under the guise of a method for rare event estimation. For the situation of ABC, we use a sequence of $T$ targets with the final one being $P_{\epsilon_{T}}(y \mid H(u,\theta)) \phi(u \mid \theta)$. The "0th target" (the proposal) is given by $\phi(u \mid \theta)$, and the $t$th target (for $1 \leq t \leq T$) is $P_{\epsilon_{t}}(y \mid H(u,\theta)) \phi(u \mid \theta)$, where $\infty > \epsilon_{1} > \ldots > \epsilon_{T} = \epsilon$. The algorithm then proceeds as in algorithm 2; our notation is that the values taken by particles in the SMC sampler have a superscript; so for example $u_{m}^{t}$ is the $u$-value taken by the $m$th particle at the $t$th target.

\[ \left(\sum_{n=1}^{N_{u}} (w^{n})^2\right)^{-1} \]

falls below some proportion $\alpha \in (0,1)$ of $N_{u}$. All SMC algorithms in this paper include a step that normalises the weights. For algorithm 2, this step uses

\[ w_{n}^{t} = \frac{\tilde{w}_{n}^{t}}{\sum_{i=1}^{N_{u}} \tilde{w}_{i}^{t}}; \]

for each particle; an analogous form is used in the other algorithms.

An illustration of the rare event approach can be found in figure 1.

For this method to be efficient, the MCMC needs to be well-designed, accounting for the fact that $\epsilon_{t}$ is decreasing at each iteration. Prangle et al. (2018) find a slice sampler to be an efficient choice here.

After running this algorithm, the ABC likelihood can be estimated using

\[ \mathcal{L}(y \mid \theta) = \prod_{t=1}^{T} \sum_{n=1}^{N_{u}} \tilde{w}_{n}^{t}. \] (2)
Algorithm 2: Rare event SMC

1. Simulate $N_u$ points, $\{u^n_0\}_{n=1}^{N_u} \sim \phi(u | \theta)$ and set each weight $w^n_0 = 1/N_u$;

2. for $t = 1 : T$ do

   3. for $n = 1 : N_u$ do // reweight

   4. if $t = 1$ then

   5. \[ \tilde{w}^n_t = w^n_{t-1} P_{t-1}(y | H(u^n_{t-1}, \theta)) ; \]

   6. else

   7. \[ \tilde{w}^n_t = \frac{w^n_{t-1}}{P_{t-1}(y | H(u^n_{t-1}, \theta))} P_{t-1}(y | H(u^n_{t-1}, \theta)) ; \]

   8. end

   9. end

10. $\{w^n_t\}_{n=1}^{N_u} \leftarrow \text{normalise}(\{\tilde{w}^n_t\}_{n=1}^{N_u})$;

11. if some degeneracy condition is met then // resample

12. for $n = 1 : N_u$ do

13. \[ \text{Simulate the index } a^n_{t-1} \sim \mathcal{M}\left(\left(w^1_t, ..., w_{N_u}^t\right)\right) \text{ of the ancestor of particle } n ; \]

14. end

15. $w^n_t = 1/N_u$ for $n = 1 : N_u$;

16. end

17. for $n = 1 : N_u$ do // move

18. Simulate $u^n_t \sim K_t(\cdot | a^n_{t-1})$, where $K_t$ is an MCMC move with invariant distribution $P_{t-1}(y | H(u, \theta)) \phi(u | \theta)$ on $u$.

19. end

20. end
1.4 Overview of the rest of the paper

This paper introduces a new approach to using ABC when \( y \) is high-dimensional, through combining ABC-SMC for exploring \( \theta \)-space with rare event SMC for estimating the ABC likelihood. The aim is to gain the strengths of both approaches.

- **Rare event ABC-MCMC** scales better than standard ABC-MCMC as the dimension of the data grows. However, both methods are highly dependent on tuning - primarily, how a user should choose \( \epsilon \) to tradeoff bias and variance expense in a sensible way. In addition, neither method can be used to estimate the model evidence.

- **ABC-SMC** allows \( \epsilon \) to be chosen adaptively, and provides an estimate of the model evidence. Also, it uses a population to explore \( \theta \)-space, and can be more suited than MCMC for exploring multi-modal targets. However, it uses the standard ABC likelihood estimate (equation (1)), which leads to poor performance for high-dimensional \( y \).

In section 2 we introduce the new method, followed by presenting empirical results in section 3 and conclusions in section 4.

2 Rare-event ABC-SMC

We begin this section by describing ABC-SMC, before moving on to the new approach.

2.1 ABC-SMC

The use of SMC samplers in the ABC setting was pioneered by Sisson et al. \(2007\), with the key idea being to use as a sequence of distributions a sequence of ABC posteriors with decreasing tolerances \( \epsilon_t \) for \( t = 1 : T \), where \( \epsilon_1 > \ldots > \epsilon_T \). The first tolerance \( \epsilon_1 \) is typically chosen to give an ABC posterior close to the prior \( p(\theta) \), which is typically used as the initial distribution \( \pi_0 \). The final tolerance \( \epsilon_T \) is chosen to be the desired tolerance, such as we would use in an ABC-MCMC algorithm.

In this paper we use the variation on the idea introduced by Del Moral et al. \(2012\). In this method the sequence of (unnormalised) targets is \( P_{\epsilon_t}(y|x) f_\theta(x)p(\theta) \) for \( t = 1 : T \), with the initial distribution being \( f_\theta(x)p(\theta) \). In algorithm 3 we describe the form of ABC-SMC that uses a likelihood estimate based on \( N \) points drawn from the likelihood for each \( \theta \) : for the \( n \)th one of these points drawn for the \( m \)th particle in \( \theta \)-space, we use the notation \( x_{n,m}^{t,\theta} \). Each step of the algorithm uses the likelihood estimate from equation (1), i.e. for any \( s,t \) we use

\[
\hat{l}_s(y \mid \theta^n_s) = \frac{1}{N^x_T} \sum_{n=1}^{N^x_T} P_{\epsilon_s}(y \mid x_{n,m}^{s,m}).
\]  

(3)

This version of the algorithm differs from the one in Del Moral et al. \(2012\), since we only use an MCMC move when a degeneracy condition is met. The sampling from the mixture distribution in the algorithm is one way of writing a resampling step, followed by an MCMC move. The reason for writing the algorithm in this way is to make a direct comparison with the SMC\(^2\) algorithm of Chopin et al. \(2013\), which we build on in the next section. A figure illustrating the algorithm is shown in figure 2 and a video illustration the steps of the algorithm can be found [here](#).

2.2 Rare-event ABC-SMC\(^2\)

In this section, we introduce our new approach: the use of the structure of the SMC\(^2\) algorithm of Chopin et al. \(2013\) in the ABC setting, through using rare event ABC to estimate likelihood ratios when required. SMC\(^2\) is designed for a state space model setting: \( y_{t:t} \) are noisy observations of a latent time series \( x_{1:t} \). The generative model for this situation is specified in two parts: \( f_\theta(x_{1:t}) \), which models the dynamics of the latent time series, and \( g_\theta(y_{t:t} | x_{1:t}) \), which models the distribution of the observations. SMC\(^2\) may be used to estimate the posterior distribution on both \( \theta \) and \( x_{1:t} \). It is set up using an “external” SMC on \( \theta \)-space, and an “internal” SMC on
Algorithm 3: ABC-SMC

1. Simulate $N_\theta$ points, $\{\theta_0^m\}_{m=1}^{N_\theta} \sim p$ and set each weight $\omega_0^m = 1/N_\theta$;
2. for $m = 1 : N_\theta$ do
3.   for $n = 1 : N_x$ do
4.     $x_{n,m}^0 \sim f(\cdot \mid \theta_0^m)$;
5.   end
6. end
7. for $t = 1 : T$ do
8.   for $m = 1 : N_\theta$ do // reweight
9.     if $t = 1$ then
10.         $\tilde{\omega}_t^m = \omega_{t-1}^m \hat{I}_1 (y \mid \theta_{t-1}^m)$;
11.     else
12.         $\tilde{\omega}_t^m = \omega_{t-1}^m \hat{I}_t (y \mid \theta_{t-1}^m)$;
13.     end
14. next
15. $\{\omega_t^m\}_{m=1}^{N_\theta} \leftarrow$ normalise $(\{\tilde{\omega}_t^m\}_{m=1}^{N_\theta})$;
16. if some degeneracy condition is met then // resample and move
17.   for $m = 1 : N_\theta$ do
18.     Simulate $(\theta_t^m, x_{1:N_x}^1 \mid \cdot)$ from the mixture distribution
19.     $\sum_{i=1}^{N_\theta} \omega_i^t K_t \left\{ \cdot \mid \left( \theta_{i-1}^t, x_{1:N_x}^{1:i} \right) \right\}$,
20.       where $K_t$ is an ABC-MCMC kernel with respect to target $t$ in the SMC (lines 4-7 of algorithm 1 using tolerance $\epsilon_t$).
21.   end
22. $\omega_t^m = 1/N_\theta$ for $m = 1 : N_\theta$;
23. end
Figure 2: An illustration of an ABC-SMC algorithm with two $\theta$-particles for three iterations, with one particle being updated using an ABC-MCMC move. Each particle has its own draws from the model, which are represented by black dots. A uniform ABC kernel is used, with the region of non-zero density for this kernel being represented by the circle. Note that once the draws from the model (the black dots) are made for each particle, they remain the same throughout the algorithm, unless a $\theta$-particle is replaced using an ABC-MCMC move (the proposed point for one particle being represented on the left of the figure), in which case it brings with it its own draws from the model, which are made during the MCMC proposal.
Figure 3: An illustration of the ABC-SMC$^2$ algorithm with two particles for three iterations, with one particle being updated using a rare event ABC-MCMC move, as described in Prangle et al. [2018]. Each particle has its own draws from the model, which are represented by black dots. A uniform ABC kernel is used, with the region of non-zero density for this kernel being represented by the red circle. Note that, in contrast to ABC-SMC the draws from the model (the black dots) are updated using MCMC moves at each iteration of the algorithm. When a particle is replaced using an ABC-MCMC move, it must use this same procedure of updates for its own draws from the model (as shown in the proposed particle on the left).

For our description of the algorithm, we follow as closely as possible, the notation in the SMC$^2$ paper. The “internal” SMC algorithm is the rare event SMC method introduced in the previous section. The “external” SMC algorithm is given in algorithm 4. A figure illustrating the algorithm is shown in figure 3 and a video illustration the steps of the algorithm can be found here.

2.3 Implementation details

Algorithm 2 may be modified in several ways in order to, at each iteration, make use of the current set of weighted particles to inform subsequent steps of the algorithm. These modifications allow many of the tuning parameters of the algorithm to be chosen adaptively. This section describes the adaptive approaches used in this paper.

2.3.1 Adapting the sequence of tolerances

One of the appealing properties of ABC-SMC (algorithm 3) in comparison with ABC-MCMC (algorithm 1) is that the choice of the tolerance $\epsilon$ may be automated in ABC-SMC. The most commonly-used approach is that of Del Moral et al. [2012]. This method adds an additional routine before line 8 of algorithm 3 to determine the choice of $\epsilon_t$ at to be used in the current iteration of the SMC. We make use of the fact that $l_t(y \mid \theta_{t-1}^{\text{true}})$ can be computed very cheaply...
Algorithm 4: Rare event ABC-SMC\(^2\) (RE-ABC-SMC\(^2\))

1. Simulate \(N_0\) points \(\{\theta_0^m\}_{m=1}^{N_0} \sim p\) and set \(\omega_0^m = 1/N_0\);
2. for \(m = 1 : N_\theta\) do
   3. Simulate \(N_x\) points \(\{u_0^{n,m}\}_{n=1}^{N_x} \sim \phi(\cdot \mid \theta_0^m)\);
4. end
5. for \(t = 1 : T\) do
   6. for \(m = 1 : N_\theta\) do // reweight
      7. if \(t = 1\) then
         8. Simulate \((u_1^{1:N_u,m}, a_0^{1:N_u,m})\) using lines 3-16 of algorithm 2 when \(t = 1\), compute
            \[
            l_1(y \mid \theta_0^m) = \sum_{n=1}^{N_u} \omega_n^{m};
            \]
            \[
            \omega_1^m = \omega_0^m l_1(y \mid \theta_0^m);
            \]
      9. else
         10. Simulate \((u_t^{1:N_u,m}, a_{t-1}^{1:N_u,m})\) using lines 3-16 of algorithm 2 compute:
            \[
            l_t(y \mid \theta_{t-1}^m) = \sum_{n=1}^{N_u} \omega_n^{m};
            \]
            \[
            \omega_t^m = \omega_{t-1}^m l_t(y \mid \theta_{t-1}^m);
            \]
      end
   11. end
12. \(\{\omega_t^m\}_{m=1}^{N_\theta} \leftarrow \text{normalise} \left(\{\omega_t^m\}_{m=1}^{N_\theta}\right)\);
13. if some degeneracy condition is met then // resample and move
14. for \(m = 1 : N_0\) do
15. Simulate \((\theta_t^m, u_{1:t}^{1:N_u,m}, a_{1:t-1}^{1:N_u,m})\) from the mixture distribution
   \[
   \sum_{i=1}^{N_u} \omega_t^i K_t \left\{ \cdot \mid (\theta_{i-1}^m, u_{t-1}^{1:N_u,i}, a_{t-1}^{1:N_u,i}) \right\},
   \]
   where \(K_t\) is the MCMC move from Prangle et al. (2018), i.e.:
   \[
   i^* \sim \mathcal{M} \left(\{\omega_t^i\}_{i=1}^{N_u}\right), \text{ then } \theta^* \sim q_t(\cdot \mid \theta_{i^*-1}^m), \text{ then run algorithm 2 up to } \epsilon_t
   \]
   conditional on \(\theta^*\).
   Set \(\theta_t^m = \theta^*\) and \(u_{1:t}^{n,m}, a_{1:t}^{n,m}\) and \(\tilde{\omega}_{n,m}\) to be the variables and unnormalised weights generated when running algorithm 2 with probability
   \[
   1 \land \frac{p(\theta^*) q(\theta_{i^*-1}^m \mid \theta^*)}{p(\theta_{i^*-1}^m) q(\theta^* \mid \theta_{i^*-1}^m)} \frac{\bar{l}(y \mid \theta^*)}{\prod_{t=1}^{N_u} \sum_{n=1}^{N_u} \tilde{w}_t^{n,m}},
   \]
   where \(\bar{l}\) is defined in equation 2
   Else set \(\theta_t^m = \theta_{i^*-1}^m\), \(u_{1:t}^{n,m} = u_{1:t-1}^{n,i^*}\), \(u_{1:t}^{n,m} = u_{1:t}^{n,i^*}\) and \(a_{1:t-1}^{n,m} = a_{1:t-1}^{n,i^*}\).
19. end
20. \(\omega_t^m = 1/N_\theta\) for \(m = 1 : N_\theta\);
21. end
22. end
for different values of $\epsilon$ given that $\{x^{n,m}_t\}_{n=1}^{\theta_u}$ have already been simulated. This means that the weights $\{\omega^{m}_t\}_{m=1}^{N_\theta}$ may be calculated for many different choices of $\epsilon$, with the most appropriate value of $\epsilon$ chosen. Usually a bisection algorithm is used to search for the most appropriate $\epsilon$ within the range $[0, \epsilon_{t-1}]$. The most appropriate $\epsilon$ is usually chosen by finding the $\epsilon$ that results in the estimated conditional effective sample size (CESS) \cite{zhou2015} being closest to some proportion $\beta \in (0, 1)$ of $N_\theta$. The CESS, defined by

$$
CESS_c = \frac{N_\theta \left( \sum_{m=1}^{N_\theta} w^{m}_{t-1} \frac{l_t(\hat{y}_{\theta^{m}_{t-1}})}{l_{t-1}(\hat{y}_{\theta^{m}_{t-1}})} \right)^2}{\sum_{m=1}^{N_\theta} w^{m}_{t-1} \left( \frac{l_t(\hat{y}_{\theta^{m}_{t-1}})}{l_{t-1}(\hat{y}_{\theta^{m}_{t-1}})} \right)^2},
$$

was introduced in as more appropriate measure of degeneracy between two successive steps of the algorithm than the ESS in the case when resampling is not performed at every step. This approach ensures that the sequence of distributions is not chosen to change so quickly that the SMC becomes degenerate.

A similar approach may be used in algorithm\ref{alg:2}. At the same stage of the algorithm (before line 6 in algorithm\ref{alg:2}), a bisection routine may be added to find $\epsilon_t$ for the next step of the algorithm. Again the CESS may be used to chose the most appropriate $\epsilon_t$, in this case given by

$$
CESS_c = \frac{N_\theta \left( \sum_{m=1}^{N_\theta} w^{m}_{t-1} \frac{l_t(\hat{y}_{\theta^{m}_{t-1}})}{l_{t-1}(\hat{y}_{\theta^{m}_{t-1}})} \right)^2}{\sum_{m=1}^{N_\theta} w^{m}_{t-1} \left( \frac{l_t(\hat{y}_{\theta^{m}_{t-1}})}{l_{t-1}(\hat{y}_{\theta^{m}_{t-1}})} \right)^2},
$$

Once more the calculation of this criterion is computationally cheap given the variables $\{u^{n,m}_{t-1}\}_{n=1}^{\theta_u}$ generated at the previous step of the algorithm. In this case, to calculate the ratio $\frac{l_t(\hat{y}_{\theta^{m}_{t-1}})}{l_{t-1}(\hat{y}_{\theta^{m}_{t-1}})}$ for each candidate $\epsilon$, we need only to run lines 3-9 of algorithm\ref{alg:2}.

### 2.3.2 Adapting the MCMC proposals

Algorithm\ref{alg:1} makes use of two MCMC steps: one in the external SMC, which we may think of as a move on $\theta$, and the other in the internal SMC, which is a move on $u$. \cite{prangle2018} noted the importance of the internal MCMC move adapting to match the changing scale of its target as $\epsilon_t$ changes, and the same is true of the external MCMC move. \cite{prangle2018} used a slice sampler to achieve this effect, but other approaches are also possible, such as using the current population of particles to estimate the scale of a proposal for a Metropolis-Hastings algorithm, as is used for example in \cite{everitt2021}.

### 2.3.3 Adapting the number of MCMC moves

As ABC-SMC, algorithm\ref{alg:2} runs, $\epsilon_t$ decreases, and it becomes more unlikely that the simulation $x$ is close to $y$. The result is that the acceptance rate of the ABC-MCMC moves used by the algorithm decreases as $t$ increases. Eventually the acceptance rate drops to zero, and whilst the tolerance $\epsilon_t$ continues to decrease and thus resampling is performed, the population ends up consisting of many duplicated particles. We observed a similar effect when running RE-ABC-SMC\ref{alg:2}, but for a much lower tolerance than for ABC-SMC.

One approach to avoiding this issue is to adapt the number of MCMC iterations used as the algorithm progresses. \cite{south2019} suggest the following approach to choosing the number of iterations for which to run a Metropolis-Hastings algorithm: at each SMC iteration they examine the acceptance rate (denoted $\hat{p}_{\text{acc}}$) across all particles of the first iteration of the MCMC move, and choose the number of iterations to be

$$
\left[ \frac{\log(c)}{\log (1 - \hat{p}_{\text{acc}})} \right],
$$
with \([\cdot]\) denoting the ceiling function, such that there is an estimated probability of \(1 - c\) that each particle is moved at least once.

In the following section we use this adaptive approach in each implementation of ABC-SMC or RE-ABC-SMC\(^2\), taking \(c = 0.2\).

### 2.3.4 Splitting different sources of randomness

Section [1.3] describes how we make use of the reparameterisation of a simulator model as a deterministic transformation \(H(u, \theta)\) of the parameter \(\theta\) and the random vector \(u\), where \(u\) encompasses all of the stochasticity in the model. We then use MCMC moves on \(u\) within the algorithm. However, for some models, such as that in section [3.2], the space of \(u\) can be extremely complicated: e.g. it may be of variable dimension. Whilst in principle it is possible to design MCMC moves for such spaces, in practice it may not be possible to do so effectively. Therefore it may be desirable to split \(u\) in two parts: \(u_s\), which will be updated by the MCMC, and \(u_g\) which will not be moved by the MCMC. This requires no change to algorithm [4]. Although the MCMC move on \(u\) is not irreversible, the algorithm is still a valid SMC sampler, as can be seen through noting that the ABC-SMC in algorithm [3] is a special case of algorithm [4] where \(u = u_s\), so that no MCMC moves are used on \(u\).

### 3 Empirical results

#### 3.1 Gaussian model

This section considers a slightly amended version of a Gaussian model studied in [Prangle et al.] (2018). We simulated \(d\) points \(y = \{y_i\}_{i=1}^d\) from a truncated univariate Gaussian distribution with \(\mu = 0\), \(\sigma = 3\), lower bound \(\alpha = 0\) and no upper bound. The aim of the inference is to estimate the posterior distribution of \(\theta = \sigma\) given \(y\). We used the uniform prior \(\theta \sim U(0, 10)\)

\[P_i(y|x)\] was chosen to be the uniform distribution \(U((y-x)^2 - \epsilon, (y-x)^2 + \epsilon)\). We use \(H(u, \theta) = ||\Phi^{-1}(u)||\) for \(u \in [0, 1]\), where \(\Phi^{-1}\) is the inverse Gaussian cdf. The MCMC move used to update \(u\) is the slice sampler from [Prangle et al.] (2018). For the move step on \(\theta\), we used a Metropolis-Hastings algorithm with a truncated (at zero) Gaussian proposal with variance chosen to be equal to the variance of the weighted particles after the reweighting step. The number of iterations of the MCMC was determined adaptively as described in section [2.3.3].

In all SMC algorithms we resample when the ESS drops below the proportion \(\alpha = 0.5\) of the number of particles, and we chose the sequence of \(\epsilon_i\) adaptively such that \(\epsilon_i\) is as close as possible to giving a CESS of \(\beta N_\theta\), where \(\beta = 0.9\).

We ran ABC-SMC and RE-ABC-SMC\(^2\) on three different scenarios: with \(d = 25\), \(50\) and \(100\). As described in section [1.3], we expect the performance of ABC-SMC to significantly deteriorate as \(d\) increases, whilst the performance of RE-ABC-SMC\(^2\) should not deteriorate as quickly. For each scenario, we initially ran a pilot run of RE-ABC-SMC\(^2\), and terminated it when the acceptance rate of the Metropolis-Hastings moves became close to zero: this corresponded to a tolerance of \(\epsilon = 3\) for \(d = 25\) (using \(N_\theta = 250, N_u = 100\)), \(\epsilon = 5\) for \(d = 50\) (using \(N_\theta = 1000, N_u = 500\)) and \(\epsilon = 10\) for \(d = 100\) (using \(N_\theta = 1000, N_u = 2000\)). We recorded the runtime for RE-ABC-SMC\(^2\) in each scenario, then ran ABC-SMC as was used by RE-ABC-SMC\(^2\). ABC-SMC used \(N_s = 1\) in every case, and for \(N_\theta\), a number of particles that gave it approximately the same runtime per iteration as RE-ABC-SMC\(^2\): \(N_\theta = 3 \times 10^3, 7 \times 10^3, 2.5 \times 10^5\) for \(d = 25, 50, 100\) respectively.

Figure [4] shows the evolution of \(\epsilon_i\), as found using the adaptive algorithm from section [2.3.1] over SMC iterations. We observe that the RE-ABC-SMC\(^2\) algorithm is able to achieve a much lower tolerance than ABC-SMC, and that this tolerance decreases more quickly.

A lower tolerance does not necessarily mean that an algorithm provides better estimates of the posterior. We also examined the estimates of the posterior mean from each algorithm at the point at which it was terminated. We ran each algorithm 50 times: box plots of the posterior mean estimates are shown in figure [4]. The left plot in each scenario corresponds to ABC-SMC, and the middle corresponds to RE-ABC-SMC\(^2\) with the parameters described in the previous paragraph. Recall that the true value of \(\theta = 3\). We observe that in all cases, the estimated posterior mean from RE-ABC-SMC\(^2\) is centred around the true value with comparatively lower
variance than that from ABC-SMC. We also observe that the estimate from ABC-SMC is biased: evidence that the tolerance achieved by ABC-SMC is not low enough to produce accurate results. Further, we ran RE-ABC-SMC\(^2\) with an increased number of \(N_u\) particles (the box plots on the right of each scenario) and observed an improved performance, as expected.

### 3.2 Parameter inference in duplication-divergence random graph models

Various random graph models have been proposed to model the formation of complex networks in biology. In particular in the study of protein interaction networks, growth models where networks are built through the addition of nodes over many steps, have been considered, such as linear preferential attachment models (Barabási and Albert, 1999), producing scale-free networks, and biologically inspired node duplication models (Vázquez et al., 2003; Pastor-Satorras et al., 2003). It has been shown that the final structure of some of these models can depend heavily on the unobserved initial state of the network used in the growth model (Hormozdiari et al., 2007).

In this example we consider a duplication-divergence model of network growth (Vázquez et al., 2003; Pastor-Satorras et al., 2003), where starting from an initial seed network, the network is grown in discrete steps, selecting a node at random to duplicate at each step. The duplicated node retains the edges of the original with a probability \(p\), and forms a link to the original node with probability \(r\). This mimics the biological process of gene duplication, where genes are copied and then diverge in function over time, with their protein products losing some of the interactions of the original gene in the process.

Our aim in this section is to estimate the parameters \(p\) and \(r\) given simulated network data \(y\). To simulate \(y\), we begin with a seed network composed of two cliques of fully connected nodes, then generate connections between them formed uniformly out of the set of possible clique to clique connections with fixed probability. Seed networks of this kind were shown in Hormozdiari et al. (2007) to produce networks with markedly different structural properties than other types of seed. From this seed, we then generate a random network with \(d = 100\) nodes using the duplication-divergence process with parameters \(p = 0.5\) and \(r = 0.2\).
Figure 5: Box plots of the estimated posterior means estimated by ABC-SMC and RE-ABC SMC\(^2\) over 50 runs, for the three different scenarios: left \(d = 25\); middle \(d = 50\) and right \(d = 100\).

The full data generation process used as the model within the ABC algorithms first simulates a seed network using an Erdős-Rényi random graph with \(d = 20\) nodes: the existence of each of the \(d_s(d_s - 1)/2\) edges is modelled by independent Bernoulli distributions with probability \(a = 0.3\). We denote the edges in the seed network by \(u_s\). Then, conditional on the seed network, the duplication-divergence process is simulated using parameters \(\theta = (p, r)\). The full details are: given an undirected seed network represented as a set of nodes \(N\) and edges \(E\), at each iteration of the model, we select a node \(x_i \in N\) uniformly at random to duplicate. To do so we first create a new node in the network \(x^*\), with no edges. Then we take all nodes \(x_j \mid (x_i, x_j) \in E\) neighbouring \(x_i\), and attach them to the new node \(x^*\) forming new edges \((x_j, x^*)\), each with probability \(p\). Finally the new node \(x^*\) is connected to the node that was duplicated, \(x_i\), forming an edge \((x_i, x^*)\), with probability \(r\). This process is repeated until the desired number of nodes \((d = 100)\) in the network is reached. We denote the variables generated in the duplication-divergence process as \(u_r\).

To apply the RE-ABC-SMC\(^2\) methodology in this context, we use the idea in section 2.3.4 of dividing the \(u\) variables into two parts: the \(u_s\) variables will be updated using MCMC moves in the internal SMC, and the \(u_r\) variables will not be updated. We make this choice since the dimension of \(u_r\) will change dependant on its value in a complex way, and would be difficult to update effectively using MCMC. Specifically, some of the \(u_r\) will correspond to the choice of whether or not to form an edge between a newly created node \(x^*\), and one of the neighbours \(x_j\) of the existing node \(x_i\) chosen to be duplicated. However the number of such \(u_r\) will depend on the number of edges \(x_i\) has, which in turn could depend on previous values of \(u_r\) used to make decisions on whether edges were formed when node \(x_i\) was created.

We used the uniform prior \(U(0, 1)\) for both \(p\) and \(r\). \(P_e (y \mid x)\) was chosen to be the uniform distribution \(U(d(y, x) - \epsilon, d(y, x) + \epsilon)\), where \(d(y, x)\) is an approximation to the edit distance between the two graphs \(y\) and \(x\). The edit distance is defined as the smallest number of edges that would need to either be added to or deleted from \(y\) or \(x\) for the two graphs to become isomorphic. This is prohibitively computationally expensive to calculate directly, but can be approximated as in Thorne and Stumpf (2012) using the ordered eigenvalues \(\alpha_1, \ldots, \alpha_d\) and \(\beta_1, \ldots, \beta_d\) of the adjacency matrices of \(y\) and \(x\) respectively as

\[
d(y, x) \approx \sum_i (\alpha_i - \beta_i)^2.
\]

To construct an MCMC kernel on \(u^*\), we apply a Metropolis-Hastings sampler with a proposal
that either adds or deletes an edge in the seed network with equal probability \( q_{\text{add}} = q_{\text{del}} = 0.5 \). When an edge addition proposal is chosen, one of the \( N(N-1)/2 - |E| \) possible pairs of unconnected nodes is selected uniformly at random, and an edge added between them. For an edge deletion proposal, one of the \( |E| \) edges in the seed network is chosen uniformly at random and deleted. 2 sweeps of the Gibbs sampler were used at every SMC iteration. For the move step on \( \theta \), we used a Metropolis-Hastings algorithm with a truncated (from 0 to 1) Gaussian proposal with variance chosen to be equal to the variance of the weighted particles after the reweighting step. The number of iterations of the MCMC was again determined adaptively as described in section 2.3.3. In all SMC algorithms again we resample when the ESS drops below the proportion \( \alpha = 0.5 \) of the number of particles, and we chose the sequence of \( \epsilon_t \) adaptively such that \( \epsilon_t \) is as close as possible to giving a CESS of \( \beta N_\theta \), where \( \beta = 0.9 \).

We used \( N_\theta = 500 \) particles in our run of RE-ABC-SMC\(^2\), with \( N_u = 200 \). ABC-SMC was set up to have approximately the same cost per iteration as RE-ABC-SMC\(^2\): we used \( N_\theta = 1.5 \times 10^6 \) external particles, with \( N_x = 1 \). Figure 6 gives a comparison of the adaptive schedules found using ABC-SMC and RE-ABC-SMC\(^2\): again we see that for RE-ABC-SMC\(^2\) the tolerance decreases more quickly, and a smaller tolerance in achieved. Figure 7 shows the box plots of posterior mean estimates from 50 runs of each algorithm, with parameter \( p \) in the left figure and parameter \( r \) in the right. The left box plot for each parameter corresponds to ABC-SMC, and the middle corresponds to RE-ABC-SMC\(^2\) with the parameters described in the previous paragraph. Recall that the true value of \( p \) is 0.5 and the true value of \( r \) us 0.2. We observe that in all cases, the estimated posterior mean from RE-ABC-SMC\(^2\) is closer to the true value with comparatively lower variance than that from ABC-SMC. We also again observe that the estimate from ABC-SMC is significantly biased, particularly for parameter \( r \). Further, we ran RE-ABC-SMC\(^2\) with an increased number of \( N_u \) particles (the box plots on the right of each scenario) and once more observed an improved performance.

4 Conclusions

This paper builds on the particle MCMC methodology of Prangle et al. (2018), introducing an SMC counterpart to the approach in that paper. The advantage of the Prangle et al. (2018) approach over standard ABC techniques is that it uses an SMC, rather than an IS, likelihood estimator. IS requires an exponential number of points in the dimension of the data to control the variance of the likelihood estimator, rendering ABC impracticable unless the dimension of
the data is reduced by considering only summary statistics. The rare event SMC approach requires only a quadratic (Prangle et al. 2018) number of points in the dimension of the data, making it possible to use ABC on larger data sets without taking summary statistics. The new SMC\(^2\) methodology in this paper inherits these advantageous properties, whilst also inheriting the useful properties of the widely-used ABC-SMC algorithms: that the particle population can be used to explore multi-modal targets (Sisson et al. 2007); that it requires few tuning parameters, due to the adaptive techniques outlined in section 2.3; and that the model evidence may be estimated directly from the SMC output (Didelot et al. 2011).

At the core of the approach is the use of the reparameterisation trick described in section 1.3. The real significance of this idea is that it removes the intractable likelihood from ABC, enabling one to use any available Bayesian computation technique for inferring the joint posterior of $\theta$ and $u$. This paper, and Prangle et al. (2018), have in common that they use a pseudo-marginal approach for this task. However, just as outside the ABC context, the most appropriate technique depends on the structure of the posterior: in some cases it may be more effective to use the HMC of Graham and Storkey (2017); in others it may be more effective to use an SMC directly on the joint posterior of $\theta$ and $u$ (as in a parallel work to ours in Zhang (2022)). The use of a pseudo-marginal approach, and hence the algorithm in this paper, is effective when there is a strong posterior dependence between $\theta$ and $u$ and it is not easy to construct an MCMC on this joint space, and when one has available an effective approach to estimating the marginal (with respect to $u$) likelihood of $\theta$ (as is the SMC method used in this paper).

The limits of this reparameterisation trick are that in practice it may be difficult to rewrite a simulator model in this way. Often ABC is used in the context when “unpacking the black box” simulator is problematic for practical reasons: e.g. a model is developed over several years by a researcher in an applied field and is coded in such a way that it is not practicable to recode it in such a way that the sources of randomness $u$ can be isolated as inputs to the simulator. In this case, from an idealistic perspective it would be possible to use the methods from this paper, but in practice this would require substantial effort that may not be available. Another situation is the case mentioned in this paper, where the space of the some of the variables is sufficiently complicated that cannot (currently) hope to design an MCMC move that can explore the space effectively. For these variables the situation remains the same as for the original formulation of ABC: that the only available proposal is the simulator.

Figure 7: Box plots of the estimated posterior means estimated by ABC-SMC and RE-ABC SMC\(^2\) over 50 runs, for the two parameters: $p$ and right $r$. 
Acknowledgements

Richard Everitt’s work was supported by NERC grant NE/T00973X/1, and Ivis Kerama’s work was supported by EPSRC grant EP/L016613/1 (the Centre for Doctoral Training in the Mathematics of Planet Earth).

References

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. (2017). Importance Sampling: Intrinsic Dimension and Computational Cost. *Statistical Science* 32(3), 405–431. Publisher: Institute of Mathematical Statistics.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods (with discussion). *Journal of the Royal Statistical Society: Series B* 72(3), 269–342.

Andrieu, C., Doucet, A., and Lee, A. (2012). Contribution to the discussion of "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation" by Fearnhead and Prangle. *Journal of the Royal Statistical Society Series B* 74(3), 451–452.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics* 37(2), 697–725.

Barabási, A.-L. and Albert, R. (1999). Emergence of Scaling in Random Networks. *Science* 286(5439), 509–512.

Beaumont, M. A. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics* 164(3), 1139–1160.

Beskos, A., Crisan, D., and Jasra, A. (2014). On the Stability of Sequential Monte Carlo Methods in High Dimensions. *The Annals of Applied Probability* 24(4), 1396–1445.

Cérou, F., Moral, P. D., Furon, T., and Guyader, A. (2012). Sequential Monte Carlo for rare event estimation. *Statistics and Computing* 22(3), 795–908.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B* 75(3), 397–426.

Cranmer, K., Brehmer, J., and Louppe, G. (2020, December). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences* 117(48), 30055–30062. Publisher: Proceedings of the National Academy of Sciences.

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.

Didelot, X., Everitt, R., Johansen, A., and Lawson, D. (2011). Likelihood-free estimation of model evidence. *Bayesian Analysis* 6(1), 49–76.

Everitt, R. G. and Rowińska, P. A. (2021). Delayed Acceptance ABC-SMC. *Journal of Computational and Graphical Statistics* 30(1), 55–66.

Fan, Y. and Sisson, S. A. (2019). ABC Samplers. In S. A. Sisson, Y. Fan, and M. A. Beaumont (Eds.), *Handbook of Approximate Bayesian Computation*. Chapman & Hall / CRC Press.

Forneron, J.-J. and Ng, S. (2016). A likelihood-free reverse sampler of the posterior distribution. In *Essays in Honor of Aman Ullah*, pp. 389–415.

Graham, M. M. and Storkey, A. J. (2017). Asymptotically exact inference in differentiable generative models. *Electronic Journal of Statistics*, 11, 5105–5164.
Hormozdiari, F., Berenbrink, P., Pržulj, N., and Sahinalp, S. C. (2007). Not All Scale-Free Networks Are Born Equal: The Role of the Seed Graph in PPI Network Evolution. *PLOS Computational Biology* 3(7), e118.

Marjoram, P., Molitor, J., Plagnol, V., and Tavare, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* 100(26), 15324–15328.

Meeds, E. and Welling, M. (2015). Optimization Monte Carlo: Efficient and Embarrassingly Parallel Likelihood-Free Inference. *Advances In Neural Information Processing Systems*, 2071–2079.

Moreno, A., Adel, T., Meeds, E., Rehg, J. M., and Welling, M. (2016). Automatic Variational ABC. *arXiv* (arXiv:1606.08549), 1–11.

Pastor-Satorras, R., Smith, E., and Solé, R. V. (2003). Evolving protein interaction networks through gene duplication. *Journal of Theoretical Biology* 222(2), 199–210.

Prangle, D., Everitt, R. G., and Kypraios, T. (2018). A rare event approach to high dimensional approximate Bayesian computation. *Statistics and Computing* 28(4), 819–834.

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America* 104(6), 1760–1765.

South, L. F., Pettitt, A. N., and Drovandi, C. C. (2019). Sequential Monte Carlo Samplers with Independent Markov Chain Monte Carlo Proposals. *Bayesian Analysis* 14(3), 753–776. Publisher: International Society for Bayesian Analysis.

Thorne, T. and Stumpf, M. P. H. (2012, October). Graph spectral analysis of protein interaction network evolution. *Journal of The Royal Society Interface* 9(75), 2653–2666.

Vázquez, A., Flammini, A., Maritan, A., and Vespignani, A. (2003). Modeling of Protein Interaction Networks. *Complexus* 1(1), 38–44.

Zhang, C. (2022). *On the Improvements and Innovations of Monte Carlo Methods*. Ph. D. thesis, University of Bristol.

Zhou, Y., Johansen, A. M., and Aston, J. A. D. (2015). Towards automatic model comparison: an adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics* 25(3), 701–726.