How to Do Bayesian Inference Without Likelihoods?

Michael Burkhardt, Fabian Kessler, and Dominik Straub
Technische Universität Darmstadt

February 26, 2018

Abstract

An essential problem in many domains of science is to identify those parameters of a model which are consistent with empirically observed data - the so-called posterior distribution in Bayesian inference. Researchers can now efficiently approximate posterior distributions that were previously not viable because of recent advances in computational resources and developments in sampling algorithms. However, these methods still require calculating the likelihood, i.e. the probability of the data given the model parameters. The likelihood function is intractable or simply unknown in many applications, but one can nevertheless specify a simulator function that generates data from the parameters. Using this simulator function, Approximate Bayesian Computation (ABC) methods bypass calculating the likelihood when computing the posterior distribution. In this paper, we describe the classic sampling-based ABC methods: Rejection-, MCMC- and SMC-ABC. In order to illustrate their differences, we apply these methods to three widely used examples from the ABC literature. Finally, we discuss shortcomings of these methods (their inefficiency, the need for good summary statistics etc.) and how they are addressed by more recent approaches like ABCDE, BOLFI and Regression ABC.

Contribution of Authors

MB wrote Section 3 and ran the experiments in that section

FK wrote Section 2 and 4

DS wrote Section 1, 5 and 6

The rest of the paper was written in collaboration

1 Introduction

In a lot of scientific domains, observed data are often understood as generated by some underlying process, which can be simulated given a set of parameters $\boldsymbol{\theta}$. Once the model has been defined, the researcher is usually interested in inferring the parameter values that are most likely to have generated the observed data $\boldsymbol{x}_{obs}$. A Bayesian approach allows us to not only compute point estimates, but rather infer the distribution over the parameters given the observation (i.e. the posterior distribution $p(\boldsymbol{\theta} \mid \boldsymbol{x}_{obs})$). This is achieved by specifying a prior distribution $p(\boldsymbol{\theta})$ over parameters, which represents the initial belief about plausible parameter values. The prior distribution is then updated by means of the likelihood function $p(\boldsymbol{x}_{obs} \mid \boldsymbol{\theta})$ of the data generating process. Formally, this is realized through Bayes' theorem:

$$\underbrace{p(\boldsymbol{\theta} \mid \boldsymbol{x}_{obs})}_{\text{Posterior}} = \frac{\overbrace{p(\boldsymbol{x}_{obs} \mid \boldsymbol{\theta})}^{\text{Likelihood}} \overbrace{p(\boldsymbol{\theta})}^{\text{Prior}}}{\underbrace{p(\boldsymbol{x}_{obs})}_{\text{Evidence}}}. \tag{1}$$

Determining the posterior distribution from Bayes' theorem requires computing the marginal likelihood (also called evidence) $p(\boldsymbol{x}_{obs}) = \int p(\boldsymbol{x}_{obs} \,|\, \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$. This integral is not generally possible to solve in closed form. Historically, this problem was overcome by relying on a simple class of likelihood functions, for which so-called conjugate prior distributions exist. These models yield an analytically tractable integral. One example is a Gaussian prior for the mean of a Gaussian random variable, which results in a Gaussian posterior. Another approach to Bayesian inference, which is applicable to models for which closed-form solutions do not exist, is to numerically approximate the posterior distribution. These kinds of methods are usually referred to as Markov Chain Monte Carlo (MCMC) algorithms. The basic idea is to circumvent computing the marginal likelihood by sampling from the unnormalized posterior. Due to recent advances in computational resources and the development of efficient MCMC algorithms, researchers can now approximate analytically intractable posterior distributions. For a review of MCMC methods, see Andrieu et al. [2003].

All of these methods still rely on the ability to calculate the likelihood function $p(\boldsymbol{x}_{obs} \,|\, \boldsymbol{\theta})$ in order to assess whether a sample should be retained or rejected. For a lot of real-world problems, however, the likelihood is intractable or computationally expensive. Often, one is still able to simulate the data generating process, i.e. draw samples $\boldsymbol{x}_{\boldsymbol{\theta}} \sim Model(x \,|\, \boldsymbol{\theta})$. Approximate Bayesian Computation (ABC) methods substitute model simulations for likelihood evaluations to allow approximate inference for such simulator-based models. The name ABC was established in population genetics, where these kinds of models are often encountered [Beaumont et al., 2002, Tavaré et al., 1997, Pritchard et al., 1999]. This choice of name is slightly misleading, since most Bayesian inference algorithms yield only approximations of the posterior, even when the likelihood is known. For this reason, ABC algorithms are sometimes also referred to as likelihood-free inference (LFI) methods [Gutmann and Corander, 2016, Papamakarios and Murray, 2016].

As this name suggests, all algorithms in the ABC family share the same basic idea: to substitute model simulations for likelihood evaluations. The following procedure is repeatedly executed: For a parameter vector $\boldsymbol{\theta}^{(i)}$ drawn from the prior distribution $p(\boldsymbol{\theta})$, a simulated data set $\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)} \sim Model(x \,|\, \boldsymbol{\theta}^{(i)})$ is generated. If the simulated data are sufficiently close to the observed data, the proposed parameter vector is accepted. More formally, the distance $\rho(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}, \boldsymbol{x}_{obs})$ needs to be smaller than some threshold ϵ. Hence, ABC methods replace the likelihood function $p(\boldsymbol{x}_{obs} \,|\, \boldsymbol{\theta})$ with an approximation $p(\rho(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}, \boldsymbol{x}_{obs}) < \epsilon \,|\, \boldsymbol{\theta})$. Instead of the true posterior, they sample from the ϵ-approximate posterior $p(\boldsymbol{\theta} \,|\, \rho(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}, \boldsymbol{x}_{obs}) < \epsilon)$. As the dimensionality of the data (or of the parameter space) increases, the probability of simulating a data set, for which $\rho(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}, \boldsymbol{x}_{obs})$ is small, decreases. For this reason, ABC methods suffer severely from the curse of dimensionality. In most practical applications, this issue is addressed by replacing the data with a set of lower dimensional summary statistics $S(\boldsymbol{x})$. These statistics should be chosen in order to capture as much information about the data as possible, i.e. they should be sufficient. For most problems that warrant ABC methods, the distribution of the data is unknown. Thus, finding nearly sufficient summary statistics is a crucial component of ABC.

Introducing summary statistics changes the acceptance criterion to

$$\rho(S(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}), S(\boldsymbol{x}_{obs})) < \epsilon \tag{2}$$

For notational simplicity, we will not distinguish between summarized and raw data in this paper (except when explicitly discussing different summary statistics): $\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}$ and $\boldsymbol{x}_{obs}$ will denote the summarized simulated and observed data. The choice of summary statistic is critically important, as it influences the correctness of the posterior approximation.

The choice of the threshold ϵ also plays a crucial role in the accuracy of the estimation. For $\epsilon = 0$, the ϵ-approximate posterior corresponds to the true posterior (assuming the summary statistics are sufficient). Choosing $\epsilon = 0$ is however only viable for low-dimensional discrete data. For all other applications, choosing a small ϵ, which leads to a good approximation of the true

posterior distribution, is desirable. Decreasing the threshold, however, decreases the probability of accepting the proposed parameters and thus leads to long computation times. Different methods have been suggested to deal with the trade-off between accuracy and computational cost. One approach is to correct the error due to the non-zero ϵ by using post-hoc regression adjustment [Beaumont et al., 2002, Blum, 2017]. A regression model, where the distance between observed and simulated data is the independent variable and the proposed parameter value is the dependent variable, is learned. The weights of this model are then used to correct the parameters towards the observed data. This idea has led to a new family of ABC algorithms called Regression ABC [Blum and François, 2010, Papamakarios and Murray, 2016], which directly approximate the posterior density with a flexible parametric regression model.

Another popular approach is to embed the basic idea of ABC - accepting parameters based on the distance between simulated and observed data - in more sophisticated sampling schemes. For instance, MCMC-ABC increases the efficiency of the basic ABC algorithm described above by sampling preferably in regions of the parameter space that have proven to produce data similar to the observed data [Marjoram et al., 2003]. Sequential Mont Carlo-ABC (SMC-ABC), on the other hand, uses a set of thresholds, which starts with a higher ϵ and decreases to a final small value of ϵ, while using the samples retained from the previous threshold as a proposal distribution [Sisson et al., 2007, Liu, 2001].

The rest of this paper is structured as follows: We will first introduce the most common sampling-based ABC approaches, starting with the basic Rejection ABC algorithm (Section 2.1). We then consider more efficient sampling schemes, in particular MCMC-ABC and SMC-ABC (Sections 2.2 and 2.3). Each of these approaches come with specific advantages and disadvantages, depending for example on the computational cost of the simulator and on properties of the posterior distribution (e.g. multiple modes, heavy tails). These are highlighted using different example problems in Section 3. Section 3 also illustrates the influence of the threshold ϵ and the use of different summary statistics. Finally, we examine more recent approaches to the problem of likelihood-free inference. In Section 4, we discuss a method that extends the SMC approach with ideas from genetic algorithms [Turner and Sederberg, 2012]. We also consider a recent approach by Gutmann and Corander [2016], which uses Bayesian Optimization and a probabilistic model of the distances between observed and simulated data (Section 5). In Section 6, we present a simple version of Regression ABC algorithms. We conclude our work with a discussion section in which we summarize our findings and highlight several open questions that are worth further investigation.

2 Sampling Algorithms

In this Section we introduce the three most common sampling-based ABC algorithms Rejection ABC, MCMC ABC and SMC ABC. We implemented these algorithms in Python as part of our PyABC package, refer to Appendix A for the implementation.

2.1 Rejection ABC

The most basic sampling-based ABC algorithm is called the 'Rejection Sampler' Tavaré et al. [1997], Beaumont et al. [2002] (see Algorithm 1). This algorithm encompasses the fundamental idea of ABC methods in its most simple form following three steps:

1. sample a proposal $\boldsymbol{\theta}^{(i)}$ from a prior distribution (line 3)
2. simulate a dataset $\boldsymbol{x}_\theta^{(i)} \sim Model(x \mid \boldsymbol{\theta}^{(i)})$ (line 4)
3. summarize the data and apply a distance metric comparing it to the observed data - do the following based on the ϵ-threshold (line 5)

Algorithm 1 Basic Rejection Sampler

Require: Data $\boldsymbol{x}_{obs} \sim Model(\boldsymbol{\theta})$, tolerance threshold ϵ, number of desired samples N, summary statistics $S(x)$ and prior distribution $p(\boldsymbol{\theta})$

for $1 \leq i \leq N$ **do**
  while *true* **do**
    Sample from prior: $\boldsymbol{\theta}^{(i)} \sim p(\boldsymbol{\theta})$
    Simulate data: $\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)} \sim Model(x \,|\, \boldsymbol{\theta}^{(i)})$
    if $\rho(S(\boldsymbol{x}_{\theta}^{(i)}), S(\boldsymbol{x}_{obs})) < \epsilon$ **then**
      Accept $\boldsymbol{\theta}^{(i)}$
      break
    end if
  end while
end for

(a) if the distance is smaller then ϵ accept the sample

(b) if the distance is larger then ϵ reject the sample

The algorithm progresses iteratively until N valid samples fulfilling the epsilon criterion have been accepted. Sampling is not altered based on previous samples or iterations and therefore a new sample for $\boldsymbol{\theta}^{(i)}$ is drawn from the prior in every iteration, which is either accepted or rejected, hence the name 'rejection sampler'. Depending on the magnitude of ϵ and also the dimensionality of $\boldsymbol{\theta}$ this can lead to a lot of samples being rejected. Also, the algorithm can potentially get stuck in an infinite loop. Once the algorithm has finished the posterior is approximated by sampling from the following distribution

$$p(\boldsymbol{\theta}|\boldsymbol{x}_{obs}) \propto \int_x p(\boldsymbol{\theta}) Model(x|\boldsymbol{\theta}) I(\rho(S(x_{\boldsymbol{\theta}}), S(\boldsymbol{x}_{obs})) < \epsilon) \; dx \tag{3}$$

Where I denotes an indicator function that is either 1 or 0 depending on the ϵ-criterion from equation (2) and thus determines whether a sample is accepted or rejected.

2.2 MCMC ABC

One idea to overcome the inefficiency of the rejection sampler is to base the current sample $\boldsymbol{\theta}^{(i)}$ on the previously accepted sample $\boldsymbol{\theta}^{(i-1)}$ and to explore the possible parameter space via MCMC generating a chain of dependent samples $[\boldsymbol{\theta}^{(1)} \rightarrow \boldsymbol{\theta}^{(2)} \rightarrow \boldsymbol{\theta}^{(3)} \rightarrow .... \rightarrow \boldsymbol{\theta}^{(N)}]$. This possibly saves countless of rejected samples in zero-density regions of the posterior and therefore expensive simulations.

The most popular algorithm for MCMC is the Metropolis-Hastings algorithm [Hastings, 1970], that computes acceptance probabilities for a proposal $\boldsymbol{\theta}^{(i)}$ based on the ratio of likelihoods of the proposal $p(\boldsymbol{x}_{\theta}^{(i)}|\boldsymbol{\theta}^{(i)})$ and the previous sample $p(\boldsymbol{x}_{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i-1)})$. In the ABC setting, however, we do not have an explicit expression for the likelihood, but the basic idea of Metropolis-Hastings MCMC can still be implemented in the ABC framework as proposed in Marjoram et al. [2003]. This algorithm called 'MCMC ABC Sampler' (see Algorithm 2) uses the following modified Metropolis-Hastings probability, which does not rely on explicitly calculated likelihoods

$$A = \begin{cases} min(1, \frac{p(\boldsymbol{\theta}^{(i)}) q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)})}{p(\boldsymbol{\theta}^{(i-1)}) q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)})}) & \text{for } \rho(S(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}), S(\boldsymbol{x}_{obs})) \leq \epsilon \\ 0 & \text{for } \rho(S(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}), S(\boldsymbol{x}_{obs})) > \epsilon, \end{cases} \tag{4}$$

but instead includes the ϵ-threshold criterion, the prior distribution $p(x)$ and the distribution q, which refers to the proposal distribution also known as transition kernel. Once again it is important

Algorithm 2 ABC MCMC Sampler

Require: Data $\boldsymbol{x}_{obs} \sim Model(\boldsymbol{\theta})$, tolerance threshold ϵ, number of desired samples N, summary statistics $\boldsymbol{S}(x)$, prior distribution $p(\boldsymbol{\theta})$

1: Set the initial value $\boldsymbol{\theta}^{(1)}$
2: **for** $2 \leq i \leq N$ **do**
3: **while** true **do**
4: Perturb $\boldsymbol{\theta}^{(i-1)}$ by transition kernel: $\boldsymbol{\theta}^{(i)} \leftarrow q(\cdot|\boldsymbol{\theta}^{(i-1)})$
5: **if** $p(\boldsymbol{\theta}^{(i)}) = 0$ **then**
6: continue
7: **end if**
8: Simulate data: $\boldsymbol{x}_{\theta}^{(i)} \sim Model(x \,|\, \boldsymbol{\theta}^{(i)})$
9: **if** $\rho(S(\boldsymbol{x}_{\theta}^{(i)}), S(\boldsymbol{x}_{obs})) < \epsilon$ **then**
10: $A \leftarrow min(1, \frac{p(\boldsymbol{\theta}^{(i)})q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)})}{p(\boldsymbol{\theta}^{(i-1)})q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)})})$
11: $u \sim Uniform(0, 1)$
12: **if** $u \leq A$ **then**
13: Accept proposal
14: break
15: **end if**
16: **end if**
17: $\boldsymbol{\theta}^{(i)} \leftarrow \boldsymbol{\theta}^{(i-1)}$
18: **end while**
19: **end for**

to note that proposals in the rejection sampler are independent of each other, as they are always drawn directly from the prior. In comparison in the ABC MCMC algorithm two subsequent samples $\boldsymbol{\theta}^{(i-1)}$ and $\boldsymbol{\theta}^{(i)}$ are dependent on each other, as $\boldsymbol{\theta}^{(i)}$ is based on $\boldsymbol{\theta}^{(i-1)}$ through the transition kernel q. If we assume the transition kernel to be a normal distribution centered on the previous sample $\boldsymbol{\theta}^{(i-1)}$ with variance $\boldsymbol{\sigma}^2$, new proposals are generated in the following manner

$$\boldsymbol{\theta}^{(i)} \sim \mathcal{N}(\boldsymbol{\theta}^{(i-1)}, \boldsymbol{\sigma}^2), \tag{5}$$

where $\boldsymbol{\sigma}^2$ can intuitively be understood as the step size of the Markov Chain. The choice of $\boldsymbol{\sigma}^2$, therefore, is critical for exploring the parameter space, but also for convergence of the chain, which is desirable in all MCMC-based methods. In the ABC case, this means to regions of the parameter-space that satisfy equation (2). Speed of convergence is strongly dependent on the initial proposal $\boldsymbol{\theta}^{(1)}$ as well as the step-size parameter $\boldsymbol{\sigma}^2$. If $\boldsymbol{\theta}^{(1)}$ initially lies in a region of low probability, an initial burn-in period with a potentially higher step-size $\boldsymbol{\sigma}^2$ might be required in order for the algorithm to converge to the higher density regions first. But if the step-size is chosen too small the algorithm can also get stuck in high-density regions, which is problematic in those distributions that are either multi-modal or heavy-tailed. This is highly dependent on the initialization of $\boldsymbol{\theta}^{(1)}$.

What follows is a description of one iteration of the algorithm. The chain starts on one sample $\boldsymbol{\theta}^{(1)}$ (line 1) that either comes from the prior or has been accepted via the rejection procedure under the threshold ϵ described in algorithm 1. A transition kernel based on the previous sample $\boldsymbol{\theta}^{(i-1)}$ with variance $\boldsymbol{\sigma}^2$ gives us a new proposal $\boldsymbol{\theta}^{(i)}$ (line 4). It is first checked whether this proposal is valid under the prior (line 5). Also the proposal $\boldsymbol{\theta}^{(i)}$ can still be rejected as it is dependent on the distance between simulated data and the observed data under the summary statistics and distance function in comparison to the ϵ-threshold (line 9). If this is the case a new proposal is generated until a valid one is found. The central equation in line 10 calculates the acceptance probability, which then decides whether the proposal $\boldsymbol{\theta}^{(i)}$ is accepted or rejected (line 12), in which case the

previous sample $\boldsymbol{\theta}^{(i-1)}$ is taken instead (line 17). This procedure is repeated until N dependent samples are accepted.

2.3 SMC ABC

Algorithm 3 ABC SMC Sampler

Require: Data $\boldsymbol{x}_{obs} \sim Model(\boldsymbol{\theta})$, a set of T decreasing tolerance thresholds ϵ, number of desired samples N, summary statistics $\boldsymbol{S}(x)$, prior distribution $p(\boldsymbol{\theta})$

1: Sample N samples with $\epsilon^{(1)} : \boldsymbol{\theta}^{(1,1:N)} = RejectionSampler(N, \boldsymbol{x}_{obs}, \epsilon^{(1)})$
2: Set weights $w^{(1,1:N)} = \frac{1}{N}$
3: Set $\sigma_1^2 = 2 * WeightedCovar(\boldsymbol{\theta}^{(1,1:N)})$
4: **for** $2 \leq t \leq T$ **do**
5: **for** $1 \leq i \leq N$ **do**
6: **while** true **do**
7: Sample: $\boldsymbol{\theta}^{(t,i)} \sim \boldsymbol{\theta}^{(t-1,1:N)}$ with weights $w^{(t-1,1:N)}$
8: Perturb $\boldsymbol{\theta}^{(t,i)}$: $\tilde{\boldsymbol{\theta}}^{(t,i)} \leftarrow \mathcal{N}(\boldsymbol{\theta}^{(t,i)}, \boldsymbol{\sigma}^2_{(t-1)})$
9: **if** $p(\tilde{\boldsymbol{\theta}}^{(t,i)}) = 0$ **then**
10: continue
11: **end if**
12: Simulate data: $\boldsymbol{x}^{(i)} \sim Model(x \,|\, \tilde{\boldsymbol{\theta}}^{(t,i)})$
13: **if** $\rho(S(\boldsymbol{x}^{(i)}), S(\boldsymbol{x}_{obs})) < \epsilon^{(t)}$ **then**
14: Accept proposal $\tilde{\boldsymbol{\theta}}^{(t,i)}$
15: Set $w^{(t,i)} = \frac{p(\tilde{\boldsymbol{\theta}}^{(i,t)})}{\sum_{j=1}^{N} w^{(t-1,j)} \mathcal{N}(\tilde{\boldsymbol{\theta}}^{(t-1,j)} | \boldsymbol{\theta}^{(t,j)}, \sigma^2_{(t-1)})}$
16: **end if**
17: **end while**
18: **end for**
19: Normalize weights such that $\sum_i^N w^{(t,i)} = 1$
20: Set $\boldsymbol{\sigma}_t^2 = 2 * WeightedCovar(\boldsymbol{\theta}^{(t,1:N)})$
21: **end for**

Another way to overcome the inefficiency of rejecting a lot of samples is by iteratively approximating the posterior distribution through a sequence of decreasing ϵ-thresholds. By placing importance weights on samples from the previous iteration, they serve as proposal distribution for the current iteration and are re-sampled in each step, after which they are perturbed by a transition kernel. This, in essence, is the idea of the so called Sequential Monte Carlo Algorithm or for short SMC ABC first proposed by [Sisson et al., 2007].

SMC ABC tries to overcome some of the limitations of MCMC ABC, for example, the choice of the step-size as a hyper-parameter, and it also allows for an efficient exploration of multi-modal distributions. As in the rejection sampler the initial samples are uncorrelated, so no burn-in period or convergence as in MCMC is required. Samples of $\boldsymbol{\theta}$ in iteration t are thought of as 'particles' $(\boldsymbol{\theta}^{(t,i)}, w^{(t,i)})$ that have an associated weight $w^{(t,i)}$. These weights are calculated according to the following equation

$$w^{(t,i)} = \frac{p(\boldsymbol{\theta}^{(t,i)})}{\sum_{j=1}^{N} w^{(t,i)} \mathcal{N}(\boldsymbol{\theta}^{(t-1,i)} | \boldsymbol{\theta}^{(t,j)}, \boldsymbol{\sigma}^2_{(t-1)})} \tag{6}$$

and specify the probability with which particles are re-sampled in every iteration from the intermediary distributions in the importance sampling step. The set of particles $P = \{(\boldsymbol{\theta}^{(t,i)}, w^{(t,i)})\}$ can be viewed as an approximation of the posterior at time-step t by satisfying equation (2) for the

threshold $\epsilon^{(t)}$. After the re-sampling step, the algorithm uses a transition kernel to perturb these samples slightly in order to explore the parameter space. In the case where this transition kernel is a symmetric distribution, such as the Normal distribution $\tilde{\boldsymbol{\theta}}^{(t,i)} \sim \mathcal{N}(\boldsymbol{\theta}^{(t,i)}, \boldsymbol{\sigma}^2_{(t-1)})$, this variant is called Population Monte Carlo (PMC). Symmetric transition kernels can lead to problems when certain parameters do not have infinite support e.g. can only be non-negative such as the rate parameter of an exponential distribution or the success probability in the case of a binomial distribution.

The SMC ABC algorithm starts out by generating a set of N candidate values $\boldsymbol{\theta}$, which are either drawn from the prior or sampled via a rejection sampler with a large enough threshold $\epsilon^{(1)}$ (line 1). Each particle gets assigned the same weight of $\frac{1}{N}$ in the first iteration (line 2). At the beginning of each iteration importance sampling based on the weights from the previous iteration is performed (line 7). Similar to the MCMC algorithm's proposal distribution a transition kernel is used to perturb the sample $\tilde{\boldsymbol{\theta}}^{(t,i)}$ to further explore the parameter space (line 8). Not always does this perturbation lead to a valid proposal under the prior, which is checked in line 9 and the procedure is repeated until a valid proposal is found. A dataset $\boldsymbol{x}^{(i)}_{\boldsymbol{\theta}}$ is simulated and subsequently compared to the observed data under the summary statistics S and distance function ρ. In the case that the distance is smaller than the threshold of the current iteration $\epsilon^{(t)}$ the sample gets accepted and a new associated weight is calculated based on equation (6) (line 15). With each successive iteration and decreasing $\epsilon^{(t)}$ the set of particles better approximates the posterior and then also serves as a proposal distribution in the next iteration. This can greatly reduce the number of rejected samples. The last iteration with threshold $\epsilon^{(T)}$ then gives a sample representation from the posterior.

3 Comparison and Characteristics of ABC Algorithms

In this section, we look at three common examples to better understand how the different ABC algorithms work, whether they are reliable and "correct", and how different parameter values influence the overall performance of these methods. We start with a simple univariate Gaussian where we want to infer the mean but know the variance. Afterwards, we take a look at a simple yet analytically intractable model by Tanaka et al. [2006] that describes a birth-death-mutation process for the spread of tuberculosis. This example belongs to a series of common benchmarks for assessing new ABC algorithm's performances, which for example can be found by Lintusaari et al. [2017a] and Sisson et al. [2007]. The third example is a classical mixture of two normal distributions with the same mean but different variance which results in a heavy-tailed distribution problematic for some ABC algorithms. Here we want to reconstruct the exact true posterior density's shape. The model was used by Sisson et al. [2007] to demonstrate the advantages of SMC ABC over classical MCMC ABC.

In this section, when we talk about the performance of an algorithm we talk about three measures: the total number of simulations the algorithm needed to gather the required number of samples, the Kullback-Leibler divergence (KL)[1] between the approximated posterior and some reference posterior – if such a reference is available, and the distance between the ABC posterior mean and the reference posterior mean. The KL divergence in the case of an approximation can be understood as a measure of inefficiency caused by the approximation. Please note that for MCMC ABC the reported number of total simulations is not corrected for the fact that the samples are correlated. For SMC ABC the reported number of total simulations is the sum of total simulations for each threshold.

[1]The KL divergence was calculated using scipy.stats.entropy and discretizing the KDE and true posterior distribution.

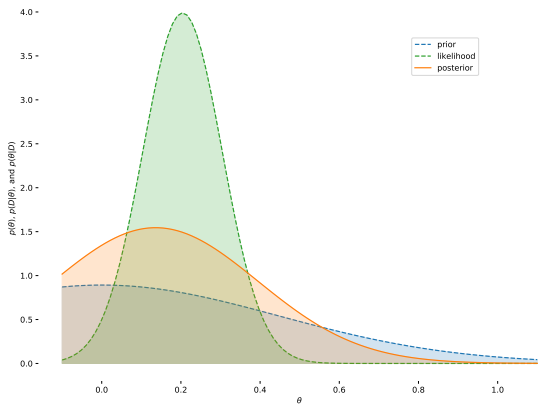


Figure 1: Prior, likelihood and posterior density functions for the univariate Gaussian model for 10 observations drawn from a Gaussian $\mathcal{N}(0.5, 1)$.

3.1 Univariate Gaussian

In this first example, we look at a univariate Gaussian as a generative model with an unknown mean but known variance of 1 (Fig. 1). This example allows us to analytically compute the posterior and compare the performance of the different ABC methods against it.

Our generative model can be stated as

$$x_{obs} \sim \mathcal{N}(\mu, 1).$$

For our observation, we choose a true mean of $\mu = 0.5$ and sample $N = 10$ data points from our generative model. To obtain a posterior, we first have to specify a prior over our parameter $\theta = \mu$. Because we are interested in an analytically closed-form solution, we choose a conjugate prior which is again a Gaussian

$$p(\theta) = \mathcal{N}(\mu_0, \sigma_0^2),$$

with $\mu_0 = 0, \sigma_0^2 = 0.2$.

With the stated prior and likelihood, we are able to compute the posterior for our model

$$p(\theta = \mu | \boldsymbol{x}_{obs}) \propto p(\boldsymbol{x}_{obs}|\mu)p(\mu) = \mathcal{N}(\mu_N, \sigma_N^2) \tag{7}$$

$$\text{with } \mu_N = \frac{1}{N\sigma_0^2 + \sigma^2} \cdot (\sigma^2\mu_0 + N\sigma_0^2\mu), \quad \sigma_N^2 = \frac{\sigma^2\sigma_0^2}{N\sigma_0^2 + \sigma^2}$$

In the following, we are going to analyze the three classical ABC methods with regard to this toy example and compare the ABC posterior density against the analytically computed posterior density in terms of the KL divergence.

3.1.1 General Performance of Different ABC Methods

First, we look at each of the three classical ABC methods Rejection ABC, MCMC ABC and SMC ABC separately (Fig. 2a). The sample mean was chosen as sufficient summary statistic (the variance is not needed as it was said to be known thus is fixed and does not change the distance metrics).

MCMC ABC needed the smallest number of simulations, followed by Rejection ABC and SMC ABC (Tab. 1). Rejection ABC showed the best performance with regard to the KL divergence, followed by SMC ABC and MCMC ABC, sharing the same KL divergence. The mean of the Rejection ABC posterior is also the closest one to the true posterior mean (one order of magnitude better than the other algorithms), followed by MCMC ABC and finally SMC ABC.

Table 1: Performance of three ABC methods

ABC algorithm	# Simulations	KL divergence[1]	$\lvert E[\boldsymbol{\theta}_{\text{true}}] - E[\boldsymbol{\theta}^{(i)}]\rvert$
Rejection ABC	74,000	0.0028	0.0022
MCMC ABC	64,407	0.0181	0.0168
SMC ABC	189,769	0.0181	0.0440

[1] ABC posterior was estimated with KDE using Scott's Rule for bandwidth

Besides the small number of samples and a threshold greater than 0, all methods were able to close in to the right location of the analytical posterior and show a unimodal shape.

3.1.2 Influence of the Threshold

Next, we analyze the influence of the threshold ϵ. As stated in the introduction, with an $\epsilon > 0$ we are no longer sampling from the desired posterior $p(\boldsymbol{\theta}|\boldsymbol{x}_{obs})$ but from the epsilon-approximate posterior (defined in (3)). Thus one of the main open questions in ABC is how to determine a reasonably small ϵ as a trade-off between accuracy and computation time.

The ABC posterior shows greater deviation from the analytical posterior for greater values of ϵ, introducing a bias into the ABC posterior so that the true mean cannot be recovered (Tab. 2 and Fig. 2b). The relationship between the threshold value, the number of simulations and the approximation's accuracy is not linear, that is decreasing the threshold increases the number of total simulations and decreases the KL divergence not in a proportional manner.

Table 2: Influence of the threshold on ABC posterior's accuracy

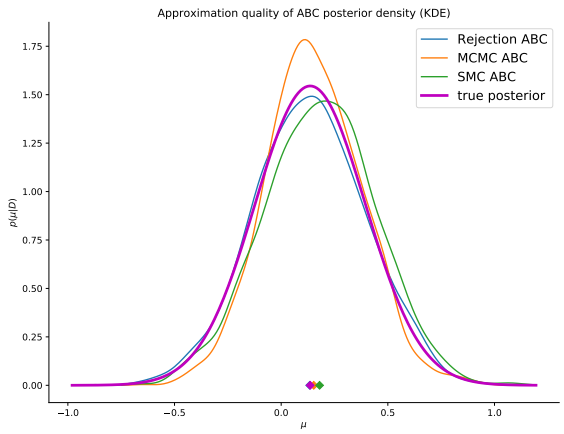
Threshold value	# Simulations	KL divergence[1]	$\lvert E[\boldsymbol{\theta}_{\text{true}}] - E[\boldsymbol{\theta}^{(i)}]\rvert$
$\epsilon = 1$	3,000	0.3403	0.0854
$\epsilon = 0.5$	4,000	0.0610	0.0268
$\epsilon = 0.02$	74,000	0.0028	0.0022

[1] ABC posterior was estimated with KDE using Scott's Rule for bandwidth

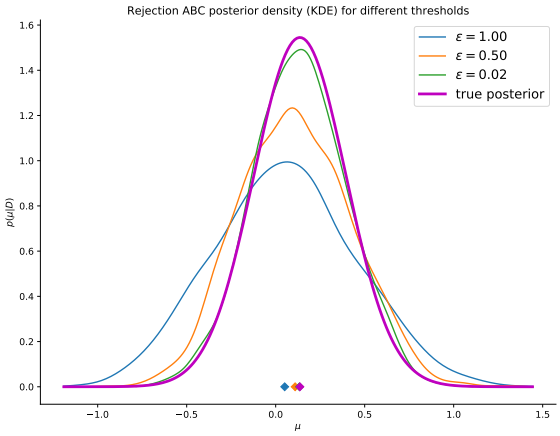
3.1.3 Influence of the Number of Samples

To improve the quality of the posterior approximation one can always increase the number of samples from which the approximation is computed. However, more samples also mean more simulations as well as a longer runtime. More simulations can be problematic whenever they are expensive and a longer runtime is problematic if runtime is precious. On the other hand, a certain number of samples is required to roughly correctly approximate the posterior. Due to the fact that by a given threshold there are always more samples $\boldsymbol{\theta}$ accepted for which the associated distances are close to the threshold than to zero (because the probability of the distance ρ being close to the threshold is higher than close to zero), we need a minimum number of samples to ensure that we have sufficient samples associated with a distance close to zero.

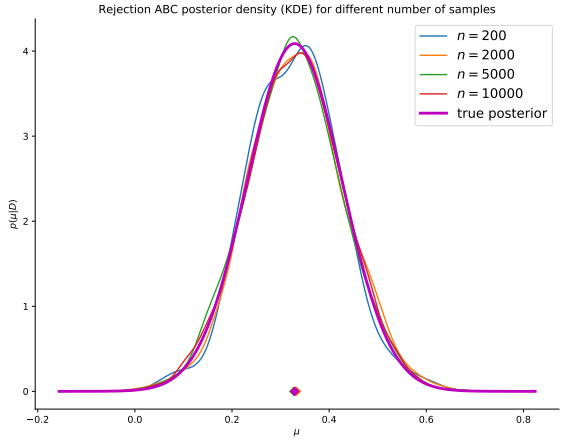
For this example, even 200 samples are enough to roughly approximate the posterior (Tab. 3 and Fig. 2c), whereas the gain in performance from 5,000 to 10,000 seems to be negligible (the negative slope of the regression line is much lower).



a) Accuracy of the three ABC methods Rejection ABC, MCMC ABC, and SMC ABC in comparison to the analytical posterior density. Parameters are 2000 samples, 10 observations and a final threshold value of 0.02. Step size for MCMC ABC was set to 0.1. SMC ABC was given a list of six thresholds ranging from 0.1 to 0.02. The true posterior's mean is $E[\boldsymbol{\theta}_{\text{true}}] = 0.1346$.



b) Accuracy of Rejection ABC with three different threshold values of 1, 0.5, and 0.02, respectively. Parameters are 2000 samples, and 10 observations. The true posterior's mean is $E[\boldsymbol{\theta}_{\text{true}}] = 0.1346$



c) Rejection ABC with four different number of samples: 200, 2,000, 5,000, and 10,000, respectively. There were 100 observations. The true posterior's mean is $E[\boldsymbol{\theta}_{\text{true}}] = 0.3266$

Figure 2: Example 1: Univariate Gaussian. Comparison of Rejection ABC, MCMC ABC, and SMC ABC and analysis of the influence of the threshold and the number of samples on general performance. Shown are the kernel density estimates for the ABC posterior densities and the ground truth posterior. The mean of each posterior density is marked with a diamond on the x axis.

Table 3: Influence of the number of samples on ABC posterior's accuracy

# Samples	# Simulations	KL divergence[1]	$\lvert E[\boldsymbol{\theta}_{\text{true}}] - E[\boldsymbol{\theta}^{(i)}] \rvert$
$n = 200$	$8,000$	0.0061	0.0004
$n = 2000$	$79,000$	0.0045	0.0042
$n = 5000$	$188,000$	0.0021	0.0020
$n = 10000$	$381,000$	0.0017	0.0006

[1] ABC posterior was estimated with KDE using Scott's Rule for bandwidth

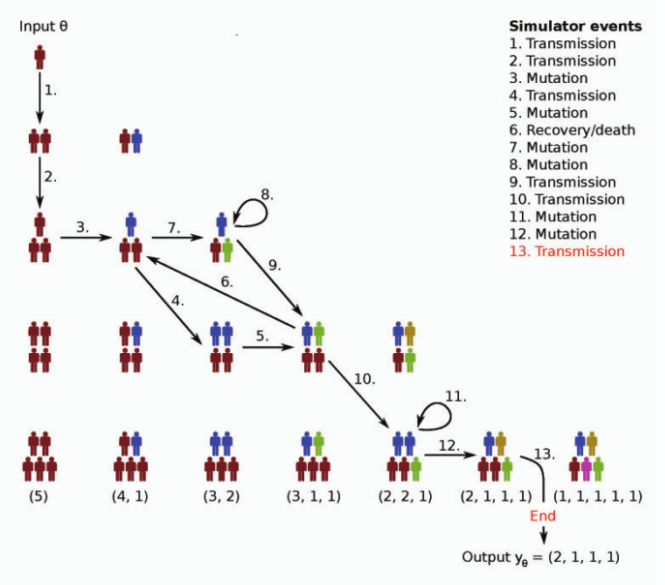


Figure 3: The model from Tanaka et al. [2006] to describe the spread of tuberculosis. There are three events: Transmission, mutation and death. Different colors represent different haplotypes of the pathogen. Arrows indicate the sequence of random events for one simulation. For greater detail see [Lintusaari et al., 2017a, p. e68].

3.2 Spread of Tuberculosis

Our second example is a biological example of the spread of tuberculosis, based on a model proposed by Tanaka et al. [2006]. The model is governed by three events and their according probabilities: an infection or transmission rate α, a recover or death rate δ, and a mutation rate τ. Thus, the model's parameters are $\boldsymbol{\theta} = \{\alpha, \delta, \tau\}$. The output of the simulator is a vector of cluster sizes, with each cluster representing a group of hosts infected by the same haplotype of the pathogen. When a fixed number of hosts m is infected – that is the sum of all cluster sizes reaches m –, the simulation stops. Figure 3 summarizes the model and shows the result of a simulation run as well as alternative states for each time step.

This example provides the rare case of an observation with discrete values, that is a list of cluster sizes like $\boldsymbol{x}_{obs} = (6, 3, 2, 2, 1, 1, 1, 1, 1, 1, 1)$. Whereas for continuous values a threshold of $\epsilon = 0$ is unfeasible, for discrete values it becomes possible to do exact inference by comparing $\boldsymbol{x}_{\text{obs}}$ with $\boldsymbol{x}_{\theta}^{(i)}$ directly (that is without the need of a summary statistic).

3.2.1 General Performance of Different ABC Methods

First we have a look at the exact inference, that is without a summary statistic $S(\boldsymbol{x})$ and with a threshold of $\epsilon = 0$ (Fig. 4a). According to Lintusaari et al. we chose $\delta = 0, \tau = 0.198$ as fixed

and known parameters and only inferred the transmission rate α (Fig. 4b). The observation was obtained with an α value of 0.2. The population size m was set to 20. The prior for α was chosen to be uniform in the interval $(0.005, 2)$. Lintusaari et al. accepted $n = 40,000$ samples for α, which resulted in an acceptance rate of 0.2%. Due to limited computational resources we chose only $n = 10,000$ samples.

Table 4: Performance of three ABC methods

ABC algorithm	# Simulations	$E[\boldsymbol{\theta}^{(i)}]$
Rejection ABC	5,062,000	0.3253
MCMC ABC	1,627,776	0.2947
SMC ABC	3,139,758	0.2866

The found solutions by all three ABC methods are very close to the reference posterior (which was numerically computed for this special case of the model) presented by Lintusaari et al., despite fewer samples. Our acceptance rate for the Rejection ABC sampler is identical to the reported acceptance rate (0.2 %). The algorithms vary in their efficiency (Tab. 4). Our MCMC ABC implementation is the fastest algorithm, followed by SMC ABC and Rejection ABC.

For the analysis of the summary statistics, we stick with the Rejection ABC algorithm because of its high resemblance to the reference posterior and because this algorithm has no further hyperparameters dependent on some optimization.

3.2.2 Influence of Summary Statistics

Up to this point, we did not talk much about summary statistics. In the first example, we knew that the chosen summary statistic was sufficient and for the second example we were not in need of a summary statistic. However, the simulation is computationally expensive and the process itself is highly stochastic, which leads to low acceptance rates (0.20%, 0.61%, and 0.32%, respectively). Sufficient summary statistics can help to reduce the runtime and needed computational resources as they reduce the dimensionality of the data and thus the cost for calculating the distance while keeping all relevant information available.

To analyze the influence of different summary statistics we now introduce two summary statistics T_1 and T_2, such that
$\rho(S(\boldsymbol{x}_\theta^{(i)}), S(\boldsymbol{x}_{obs})) = |T_1(\boldsymbol{x}_\theta^{(i)}) - T_1(\boldsymbol{x}_{obs})|$ (analogously for T_2). T_1 is just the number of clusters contained in the data divided by the sample size n, i.e. $T_1(\boldsymbol{x}) = \frac{dim(\boldsymbol{x})}{n}$. T_2 is a genetic diversity measure defined as $T_2(\boldsymbol{x}) = 1 - \sum_i (\frac{n_i}{n})^2$.

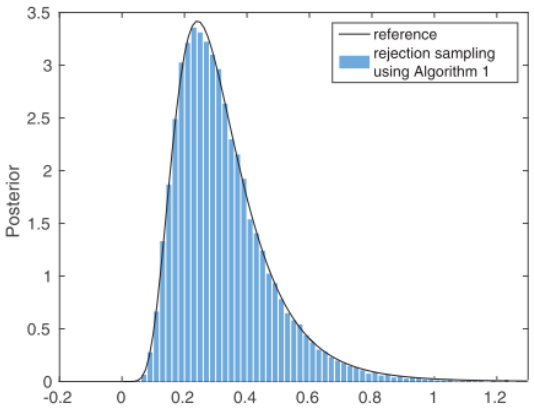
Both summary statistics allow us to accept a sampled value for α even if the simulated data $\boldsymbol{x}_\theta^{(i)}$ does not equal $\boldsymbol{x}_{\text{obs}}$. But we do not know if the summary statistics are sufficient and if that is not the case what bias they introduce into the approximation.

Table 5: Performance of Rejection ABC with summary statistics T_1

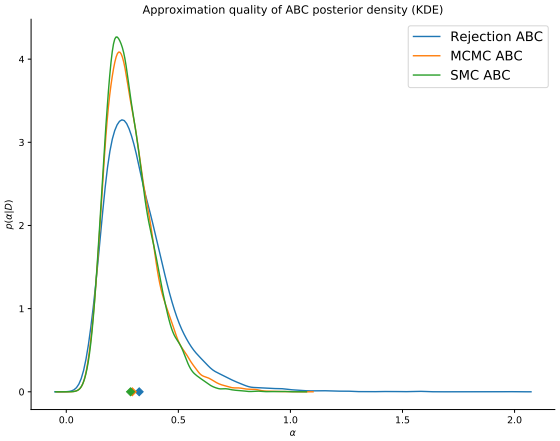
Threshold	# Simulations	KL divergence[1]	$\lvert E[\boldsymbol{\theta}_{\text{ref}}] - E[\boldsymbol{\theta}^{(i)}]\rvert$
$\epsilon = 0.20$	51,000	0.2071	0.0595
$\epsilon = 0.10$	114,000	0.0272	0.0105
$\epsilon = 0.05$	235,000	0.0177	0.0229

[1] ABC posterior was estimated with KDE using Scott's Rule for bandwidth

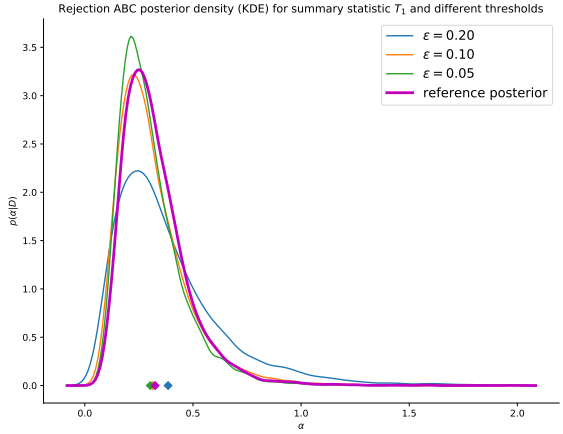
T_1 seems to be a plausible summary statistic as there is no bias introduced into the ABC



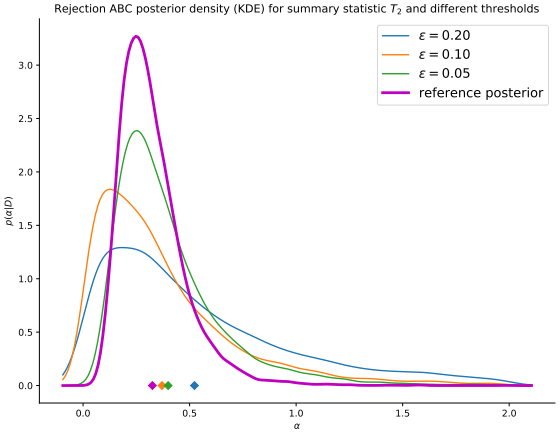
a) Solution of exact inference. Taken from Lintusaari et al. [2017a].



b) Three ABC posteriors for exact inference ($\epsilon = 0$). Number of samples was set to $n = 10000$. MCMC ABC's step size was 0.1. SMC ABC run with $\epsilon \in \{4, 3, 2, 1, 0\}$.



c) Rejection ABC with T_1 as summary statistic and three different threshold values: $\epsilon = 0.2, \epsilon = 0.1$, and $\epsilon = 0.05$, respectively. The reference posterior mean is $E[\boldsymbol{\theta}_{\text{ref}}] = 0.3253$.



d) Rejection ABC with T_2 as summary statistic and three different threshold values: $\epsilon = 0.2, \epsilon = 0.1$, and $\epsilon = 0.05$, respectively. The reference posterior mean is $E[\boldsymbol{\theta}_{\text{ref}}] = 0.3253$.

Figure 4: Example 2: Spread of Tuberculosis. Comparison of Rejection ABC, MCMC ABC, and SMC ABC and analysis of two summary statistics. Shown are the kernel density estimates for the ABC posterior densities and the ground truth posterior. The mean of each posterior density is marked with a diamond on the x axis.

posterior and the shape closely resembles the reference posterior's shape (Fig. 4c and Tab. 5). Using T_1 instead of exact inference brought an advantage with respect to the runtime as the number of total simulation decreased notably. The Rejection ABC posterior for an ϵ of 0.2 is not accurate enough and its shape clearly deviates from the reference posterior. Instead, a threshold value of 0.1 or less produces approximations very close to the reference posterior. Thus using T_1 and an $\epsilon \leq 0.1$ lead to a manifold decrease of the total number of simulations (from $5,062,000$ to $235,000$) without a loss in the approximation's quality.

Table 6: Performance of Rejection ABC with summary statistics T_2

Threshold	# Simulations	KL divergence[1]	$\lvert E[\boldsymbol{\theta}_{\text{ref}}] - E[\boldsymbol{\theta}^{(i)}]\rvert$
$\epsilon = 0.20$	$27,000$	1.4727	0.1975
$\epsilon = 0.10$	$38,000$	0.9711	0.0452
$\epsilon = 0.05$	$86,000$	0.1689	0.0738

[1] ABC posterior was estimated with KDE using Scott's Rule for bandwidth

T_2, on the other side, introduces an approximation error which is clearly visible (Fig. 4d) and is also expressed in a higher KL divergence (Tab. 6). This was also reported by Lintusaari et al.. Only for a threshold value of 0.05 and less the ABC posterior's mode begins to resemble the reference posterior's mode. However, even then the shape of the approximated posterior shows a greater deviation from the reference posterior than compared to the summary statistic T_1 (KL divergence of 0.1689 against 0.0177). Note how halving the threshold value from 0.2 to 0.1 led only to a very small change in the total number of simulations, indicating a possible way to assess a summary statistics' appropriateness.

3.3 Mixture of Gaussians

In this example we choose a particular shape for the posterior that will be more problematic for some of the presented samplers – and less problematic for others (Fig. 5). This will give us the opportunity to analyze advantages and disadvantages of different ABC methods. We assume the posterior to be formed by a mixture of two normal distributions with shared mean but different variances, resulting in a more heavier tailed distribution which makes inference more difficult.

$$f(\theta|x_0) = \frac{1}{2} \cdot \left(\mathcal{N}(0, \frac{1}{100}) + \mathcal{N}(0, 1) \right). \tag{8}$$

The first component has a small variance and produces a sharp peak, whereas the second component with a variance of 1 produces larger regions with relatively low probability mass.

This posterior can be implemented within an ABC setting by drawing from a normal distribution $x_i \sim \mathcal{N}(\theta, 1)$ and then using the following distance function

$$\rho(S(\boldsymbol{x}_\theta^{(i)}, S(\boldsymbol{x}_{\text{obs}})) = \begin{cases} |\bar{x}| & \text{with probability } \frac{1}{2} \\ |x_1| & \text{with probability } \frac{1}{2} \end{cases}. \tag{9}$$

In half the cases the distance is equal to the empirical mean of the simulated data, whereas in the other half of cases the distance equals the absolute distance of the first data point.

3.3.1 Performance of Different ABC Methods

We start our discussion with the most basic method Rejection ABC. Though highly inefficient (over nine million simulations), the final shape of the Rejection ABC posterior density is very close to

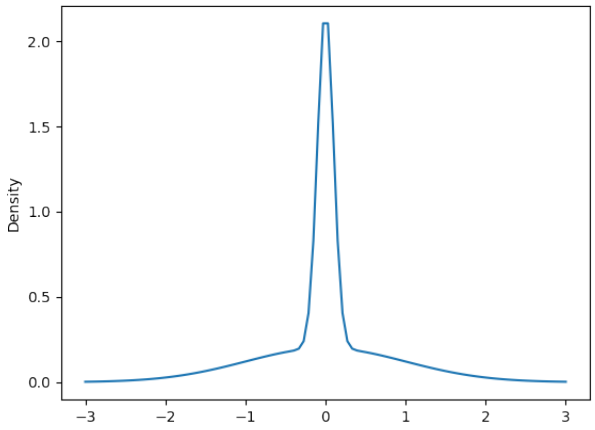


Figure 5: Mixture of two normal distributions with mean $\mu = 0$ and variances $\sigma_1^2 = 1$ and $\sigma_2^2 = 0.01$, respectively.

Table 7: Performance of Rejection ABC, MCMC ABC, and SMC ABC

ABC Method	Parameter	# Simulations	KL divergence	$\lvert E[\boldsymbol{\theta}_{\text{true}}] - E[\boldsymbol{\theta}^{(i)}] \rvert$
Rejection ABC	-	9,952,000	0.0400	0.0067
MCMC ABC	step size: 0.15^2	609,811	0.2969	0.0008
MCMC ABC	step size: 0.5^2	1,141,377	0.1115	0.0017
SMC ABC	# thresholds: 3	2,175,425	0.0367	0.0105
SMC ABC	# thresholds: 6	1,876,087	0.0507	0.0024

[1] ABC posterior was estimated with KDE using Scott's Rule for bandwidth

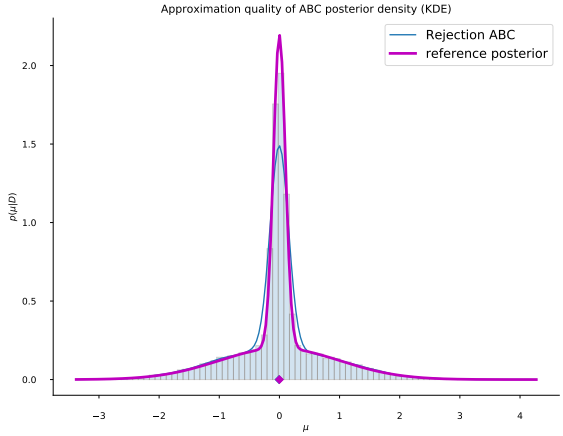
the ground truth (Fig. 6a and Tab. 7). Especially the heavy-tailed regions are well approximated by the ABC method.

Now we compare this result with MCMC ABC's performance. Although MCMC ABC only required 609,811 simulations, the method does not accurately approximate the tail regions of the true posterior (Fig. 6b and Tab. 7). However, the deviation from the true mean is negligible.

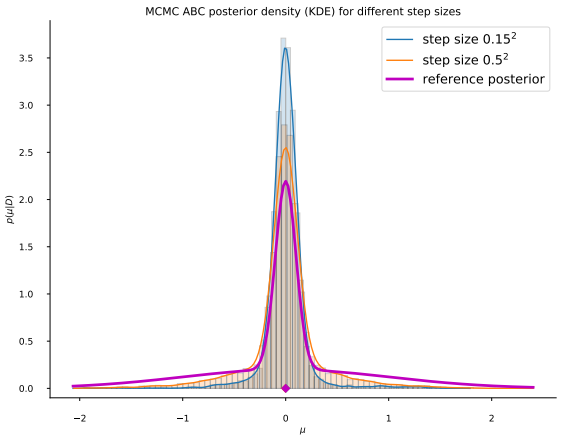
We can try to "fix" and improve this behavior by increasing the step size of the MCMC ABC algorithm allowing it to find more solutions in the tail regions. However, increasing the step size also increases runtime. The gained improvement in the approximation comes for the prize of a roughly doubled runtime. In return, we gained a better approximation of the tail region. This illustrates the importance of the proper choice of step-size as a hyperparameter of MCMC ABC methods.

Finally, we look at SMC ABC's performance. One can see that SMC ABC is better in approximating the tail regions than MCMC ABC (Fig. 6c and Tab. 7). For SMC ABC, the number of thresholds is a critical hyperparameter, influencing the runtime as well as the accuracy of the method.

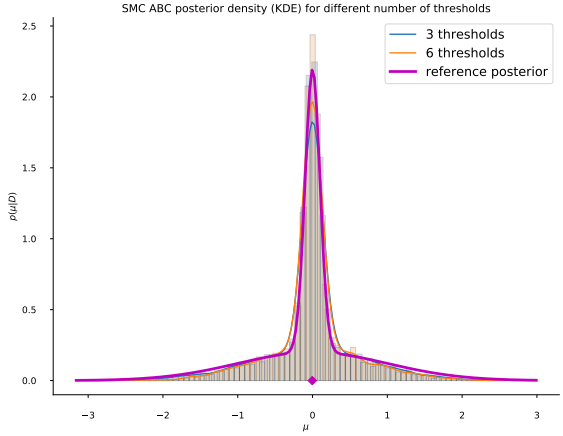
To conclude, this example demonstrated the advantages and disadvantages as well as the appropriateness of different ABC methods.



a) Performance of Rejection ABC. Parameters are $N = 1$ observational data point $\boldsymbol{x}_{\text{obs}} = 0$, $n = 10,000$ samples, $\epsilon = 0.01$, and a uniform prior $\theta \sim U(-10, 10)$.



b) Performance of MCMC ABC for $N = 1$ observational data point, $n = 10,000$ samples, and $\epsilon = 0.01$. We compare the two step sizes 0.15^2 and 0.5^2.



c) Performance of SMC ABC for $N = 1$ observational data point, $n = 10,000$ samples, and $\epsilon = 0.01$. We compare a list of three and six thresholds.

Figure 6: Example 3: Mixture of Gaussians. Comparison of Rejection ABC, MCMC ABC, and SMC ABC. Shown are the kernel density estimates for the ABC posterior densities and the ground truth posterior. The mean of each posterior density is marked with a diamond on the x axis. Because KDE is sensitive to the right choice of the bandwidth, the histogram of the drawn samples is shown as well.

4 ABC with Differential Evolution (ABCDE)

The three ABC methods proposed in section 2 all rely on an explicitly defined threshold ϵ which is thought of as a trade-off between computation time and accuracy of estimation and hence has to be carefully chosen by the user. In cases where simulation is expensive, a simple rejection-based approach leads to a high degree of inefficiency as many proposals get discarded straight away once they do not satisfy equation 2. An additional problem these classical ABC algorithms face is a decrease in inference performance as the dimensionality of $\boldsymbol{\theta}$ increases since the search space of possible parameter combinations becomes increasingly large. One sampling-based algorithm that tries to overcome these limitations is the so-called ABC with Differential Evolution (ABCDE) [Turner and Sederberg, 2012], which combines elements from the SMC and MCMC algorithm

with Differential Evolution, a method from genetic algorithms for efficient proposal generation and parameter space exploration.

4.1 Continuous Evaluation of Proposal Parameters

Similar to the SMC algorithm, samples are realized as particles with associated weights. But in ABCDE, weights are thought of as a continuous measure of a particles 'fitness', indicating how well they perform in terms of approximating values from the posterior. The central goal of the algorithm is to continually improve those fitness values across iterations until convergence. For this purpose the indicator function $I(\rho(S(\boldsymbol{x}_\theta^{(i)}, S(\boldsymbol{x}_{obs})) < \epsilon)$ for rejecting/accepting is replaced with a kernel $\psi(\rho(S(\boldsymbol{x}_\theta^{(i)}), S(\boldsymbol{x}_{obs})) \,|\, \delta)$, which provides a continuous evaluation for a certain proposal parameter $\boldsymbol{\theta}^{(i)}$. It is assumed that the observed data $\boldsymbol{x}_{obs}$ is a realization of a model simulation under the best possible parameter values $\hat{\boldsymbol{\theta}}$ plus some random error ξ

$$\boldsymbol{x}_{obs} = Model(y|\hat{\boldsymbol{\theta}}) + \xi \tag{10}$$

where ξ follows the distribution $\psi(\cdot|\delta)$ which allows modeling it as a continuous random error under the parameter δ. One possibility to model this error is to assume the distribution ψ to be a normal distribution centered around zero such that $\xi \sim \mathcal{N}(\rho(S(\boldsymbol{x}_\theta^{(i)}), S(\boldsymbol{x}_{obs})) \,|\, 0, \delta)$ which will yield higher probabilities for simulated data that are close to the observed data assuming that the summary statistics are sufficient. Equation (3), which represented how samples where distributed in the rejection-based algorithms, now becomes the following:

$$p(\boldsymbol{\theta}|\boldsymbol{x}_{obs}) \propto \int_x p(\boldsymbol{\theta}) Model(\boldsymbol{x}|\boldsymbol{\theta}) \mathcal{N}(\rho(S(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}), S(\boldsymbol{x}_{obs})) \,|\, 0, \delta) \tag{11}$$

These so-called kernel-based ABC methods [Wilkinson, 2013] can potentially improve the computational efficiency of a sampler, but one has to keep in mind that the accuracy of the estimated posterior still relies on the proper selection of the parameter δ. It is however possible to infer δ as a latent variable along with the model parameters, such that only a prior on δ-values has to be selected. A good prior for the selection of δ would be the exponential distribution, in which smaller values are favored over larger values. We can then evaluate $\boldsymbol{\theta}^{(i)}$ in a continuous fashion and therefore calculation of a particles' fitness becomes

$$\pi(\theta) \mathcal{N}(\rho(S(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}), S(\boldsymbol{x}_{obs})) \,|\, 0, \delta) q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)}) \tag{12}$$

with q as in SMC and MCMC ABC referring to a transition kernel indicating the probability to transition from $\boldsymbol{\theta}^{(i-1)}$ to $\boldsymbol{\theta}^{(i)}$. The rule that finally helps us to decide, whether the proposed sample should be accepted or rejected, is a modified version of the Metropolis-Hastings probability from equation 4, which uses the ratio of the fitness values to decide, whether a 'jump' is performed by accepting the proposal:

$$min\left(1, \frac{p(\boldsymbol{\theta}^{(i)})\ \mathcal{N}\left(\rho(S(\boldsymbol{x}_\theta^{(i)})), S(\boldsymbol{x}_{obs})) \,|\, 0, \delta\right)\ q(\boldsymbol{\theta}^{(i)}|\boldsymbol{\theta}^{(i-1)})}{p(\boldsymbol{\theta}^{(i-1)})\ \mathcal{N}\left(\rho(S(\boldsymbol{x}_\theta^{(i-1)}), S(\boldsymbol{x}_{obs})) \,|\, 0, \delta\right) q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^{(i)})}\right) \tag{13}$$

4.2 Mutation, Crossover and Migration Step

In our description of the ABCDE algorithm, we first look at the mutation and crossover step that try to explore the parameter-space by generating new proposals and decide via the Metropolis-Hastings probability (see equation (13)) whether these proposals should be accepted over the previous parameter value for $\boldsymbol{\theta}$. We then look at the migration step which swaps particles across groups in order to diversify the population of particles.

Mutation The mutation step works similar to the proposal generation of both the MCMC algorithm (line 4) and the SMC algorithm (line 8) where a simple gaussian transition kernel centered around the current sample with step-size σ is used to propose a new sample $\boldsymbol{\theta}$, which is accepted according to the Metropolis-Hastings probability from equation 13. This step is performed with probability β.

Crossover In the crossover step proposals for θ are generated via Differential Evolution [Storn and Price, 1997], which can be thought of as a linear combination of the parameters θ, in which the dimensions of θ span a vector space. The idea is that well-performing particles within a group are used, to guide the poorer performing particles into higher-density regions. A new proposal based on the particle of the previous iteration $\theta^{(t-1)}$ then generated according to the following equation

$$\theta_* = \theta^{(t-1)} + \gamma_1(\theta_m - \theta_n) + \gamma_2(\theta_b - \theta_{(t-1)}) + b \tag{14}$$

with $\theta_{(t-1)} \neq \theta_m \neq \theta_n$ where θ_m and θ_n are sampled from the population with uniform probabilities and θ_b, the so called 'base particle', is sampled from the population according to the current weights as probabilities. We obtain scalars from a uniform distribution such that $\gamma_1, \gamma_2 \sim Uniform(0.5, 1)$ and the noise term $b \sim Uniform(-0.001, 0.001)$. To slow the exploration of the parameter-space some of the proposal-parameters are reset to their previous value with probability $(1 - k)$ where k is typically set to 0.9.

Migration Now that we know how new proposals are generated within the groups of particles, we look at how the migration step (see line 5) tries to diversify particles across groups. Within groups particles are thought converge to high-density regions of the posterior. Note that some groups converge faster then others and also some particles perform poor in a particular group, but could still contribute valuable information to other pools of particles that either converge to a different region of the posterior or simply perform worse in general. Hence the migration step performs swapping of particles with low weights between a small subsets of n groups in the following manner:

$$\left\{\theta^*_{g_1}, \theta^*_{g_2}, \dots, \theta^*_{g_n}\right\} \rightarrow \left\{\theta^*_{g_n}, \theta^*_{g_1}, \dots, \theta^*_{g_{n-1}}\right\} \tag{15}$$

4.3 High-level Overview of the Algorithm

Having explained the basic principles, we now take a look at the algorithm (see Algorithm 4) itself, which consists of the following three steps: migration (line 5), mutation (line 10) and the crossover step (line 12). For a full description, we refer the reader to Turner and Sederberg [2012]. The ABCDE algorithm differs from other sampling-based algorithms in three ways.

First, a continuous evaluation of proposal parameters over T iterations, instead of a simple ϵ-threshold criterion, leads to fewer samples being discarded straight away and the algorithm can not get stuck in an infinite loop. Rather each sample of θ can potentially contribute to improving the overall solution in an iterative fashion. The number of simulations performed by the algorithm is known beforehand, as it is simply the product of the number iteration T times the number of samples N. Also due to the iterative nature and continuous evaluation, no explicit threshold ϵ is required beforehand as it is treated as a latent variable δ and inferred across iterations.

Second, the Crossover-Step serves as another mean of generating new proposals for θ. By essentially operating in a vector space and using linear combinations of intermediate solutions of θ to generate new potentially better solutions. The crossover is thought to generalize well, to higher-dimensional parameter spaces, which is demonstrated in the paper [Turner and Sederberg, 2012]

Algorithm 4 ABCDE

Require: Data $x_{obs} \sim Model(\theta)$, summary statistics $s_1, s_2 ... s_n$ and prior distribution $\pi(\theta)$, number of iterations T, number of groups K, group size G and decision tuning parameters α and β

Sample from prior: $\theta_{1:K,1:G,1} \sim \pi(\theta)$
for $2 \leq t \leq T$ **do**
  $p_1^* \sim Uniform(0, 1)$
  if $p_1^* < \alpha$ **then**
    $\theta_{1:K,1:G,t} \leftarrow Migrate(\theta_{1:K,1:G,t-1})$
  end if
  for $1 \leq k \leq K$ **do**
    $p_2^* \sim Uniform(0, 1)$
    if $p_2^* < \beta$ **then**
      $\theta_{k,1:G,t} \leftarrow Mutate(\theta_{k,1:G,t-1})$
    else
      $\theta_{k,1:G,t} \leftarrow Crossover(\theta_{k,1:G,t-1})$
    end if
  end for
end for

on a 20-parameter problem, where they infer the mean of a 20-dimensional multivariate Gaussian distribution.

Thirdly, the samples, which are also referred to as particles, are first evenly divided into G groups of size K, such that $N = G * K$. Multiple groups can help finding different modes or regions of probability (such as the tails) from the posterior, as particles within a group are guided towards higher-densities regions in the crossover and mutation step and swapped across groups in the migration step

We've implemented a basic version of the ABCDE algorithm according to the implementation details given by Turner and Sederberg [2012] and initially evaluated our implementation on the Mixture of Gaussians example of Section 3.3 trying to match the performance of the other ABC samplers presented in Section 2. Using parameters similar to the ones from the paper by [Turner and Sederberg, 2012], except that we used 10.000 samples instead of 100 samples and $\lambda = 4$, our posterior (Fig. 7) visually looks almost identical to the one by Turner and Sederberg [2012] (refer to Figure 3 of the paper). Depending on the hyper-parameters of ABCDE and desired accuracy our implementation needs between 5 and 2 million simulations, indicating a high sensitivity to hyper-parameters and resulting convergence characteristics.

As a second example, we tried the ABCDE algorithm on the 20-dimensional Gaussian example, a high-parameter problem inferring the mean also presented in the original paper, but failed to produce any good approximation of the posterior and therefore were not able to replicate the original claims made by the authors. Unfortunately, the authors had not provided any reference implementation we could have used to rule out problems with our own implementation, which had worked just fine in the example above producing a posterior comparable to the original paper.

While the algorithm has provided some interesting concepts, for example proposal generation via linear combinations of different $\boldsymbol{\theta}$ and the continuous evaluation of errors using kernels, we were not able to achieve a lower number of simulations for a good approximation of the posterior density, than both our SMC and MCMC implementations. Also the fact that up to 7 hyper-parameters have to be tuned before one can expect good approximations, made this algorithm highly complex and hard to use in real applications.

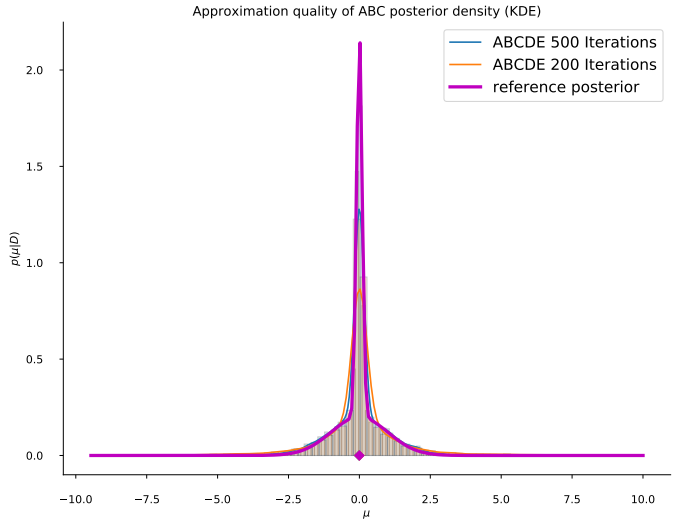


Figure 7: Posterior Distribution of Mixture of Gaussian using two different ABCDE Samplers with n=10.000 and 500/200 iterations (300/150 iterations of burn-in) for a total of 5.000.000/2.000.000 model simulations using G = 500 groups, $\lambda = 4$, $\alpha = 0.1$, $\beta = 0.0$, $\kappa = 0.9$. KL divergence of 0.0746 and 0.6746 respectively. $|E[\boldsymbol{\theta}_{\text{true}}] - E[\boldsymbol{\theta}^{(i)}]| = 0.01300$ and 0.0038 respectively

5 Bayesian Optimization for Likelihood-Free Inference (BOLFI)

Classical ABC methods suffer from computational inefficiency, because they do not use all the information available: They discard proposed parameter values that yield a distance higher than ϵ without considering the relationship between the parameter values $\boldsymbol{\theta}^{(i)}$ and the distances $\rho(\boldsymbol{\theta}) = \rho(\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}, \boldsymbol{x}_{obs})$. Attempts have been made to use this information in order to correct the parameters post-hoc [Beaumont et al., 2002]. This approach still requires simulating a large amount of data, because the relationship between parameters and distances is not used during sampling. For instance, if there are regions in the parameter space, where few simulations determine that the distances are very unlikely to be lower than the threshold, these regions need not be further explored. Thus, a model $J(\boldsymbol{\theta}) = E[\rho(\boldsymbol{\theta})]$, which describes the mapping from parameters to distances, could be used to make ABC more efficient: which regions of the parameter space promise to result in low distance values?

Gutmann and Corander [2016] address this by creating a Bayesian model of the distances and iteratively updating this model using Bayesian optimization. Specifically, they model J as a Gaussian process (GP). For a comprehensive introduction to Gaussian processes, see Rasmussen and Williams [2006]. Essentially, modeling the expected distances as a GP assumes that the joint distribution of $\boldsymbol{J}_n = (J(\boldsymbol{\theta}^{(1)}), ..., J(\boldsymbol{\theta}^{(n)}))$ for any arbitrary evidence set $(\theta^{(1)}, ..., \theta^{(n)})$ is Gaussian with a mean and covariance function.

If we are now interested in the prediction on a test output $J(\boldsymbol{\theta})$, we can model the joint distribution of the evidence set $\boldsymbol{J}_n$ and the test output as Gaussian and condition on the evidence set. Since the conditional distribution of a Gaussian random variable given a set of Gaussian random variables is also Gaussian, this yields

$$J(\boldsymbol{\theta})|\boldsymbol{J}_n \sim \mathcal{N}(\mu_n(\boldsymbol{\theta}), v_n(\boldsymbol{\theta}) + \sigma^2). \tag{16}$$

For the derivations of $\mu_n(\boldsymbol{\theta})$ and $v_n(\boldsymbol{\theta})$, see Rasmussen and Williams [2006].

5.1 Parameter acquisition using Bayesian Optimization

We have now arrived at a posterior distribution over distances given an evidence set of already simulated distances. This initial set is for example attained by repeatedly sampling from the prior, simulating data sets, and computing the distance to the originally observed data. The posterior from equation (16) can then be used to decide from which region of the parameter space the next samples should be drawn. Gutmann and Corander [2016] propose the Bayesian Optimization framework [Snoek et al., 2012] for this purpose. In a nutshell, the next point $\boldsymbol{\theta}^{(n+1)}$ is chosen, such that it *potentially* yields a low distance. This is achieved by means of an acquisition function $\mathcal{A}_n(\boldsymbol{\theta})$, which implements a trade-off between exploration and exploitation. The acquisition function should be low where the mean of the prediction $J(\boldsymbol{\theta})|\boldsymbol{J}_n$ is low, and it should also be low where there is a great uncertainty in the prediction. This minimizer of this acquisition function is then the next parameter value $\boldsymbol{\theta}^{(n+1)}$ (see Fig. 8). A common choice of acquisition function is the lower confidence bound criterion (LCB, introduced by Cox and John [1997]):

$$\mathcal{A}_n(\boldsymbol{\theta}) = \mu_n(\boldsymbol{\theta}) - \kappa v_n(\boldsymbol{\theta}), \tag{17}$$

where κ is a constant that controls the exploration-exploitation trade-off. The LCB criterion is also the acquisition function used by Gutmann and Corander [2016].

This acquisition function is designed with optimization in mind and thus yields good results near the maximum of the likelihood. The low-density regions are often not explored further, which leads to a less accurate ABC posterior estimate. Järvenpää et al. [2017] propose an acquisition function specifically for ABC, which chooses the point which results in the lowest expected ABC posterior uncertainty.

5.2 Likelihood approximation

Once the GP has been fit, its mean and variance (μ_n and v_n) can be used to construct a closed-form approximation $\hat{L}_n(\boldsymbol{\theta})$ of the likelihood. In the ABC setting, a non-parametric likelihood approximation can be understood as an expectation over model simulations:

$$\hat{L}_n(\boldsymbol{\theta}) = \int Model(x|\boldsymbol{\theta})\, \psi(\rho(\boldsymbol{\theta}) \,|\, \epsilon)\, \mathrm{d}x = E[\psi(\rho(\boldsymbol{\theta}) \,|\, \epsilon)] \tag{18}$$

where ψ is a kernel with bandwidth ϵ. If we now assume ψ to be a uniform kernel (i.e. $\psi(u) = const$ if $u < \epsilon$ and $\psi = 0$ otherwise, which is equivalent to the indicator function in Rejection ABC), the likelihood approximation becomes

$$\hat{L}_n(\boldsymbol{\theta}) \propto P(\rho(\boldsymbol{\theta}) < \epsilon). \tag{19}$$

Since $\rho(\boldsymbol{\theta})$ is modeled as a GP (and thus follows a Gaussian distribution) the probability $P(\rho(\boldsymbol{\theta}) < \epsilon)$ can be expressed as

$$\hat{L}_n(\boldsymbol{\theta}) \propto \Phi(\epsilon \,|\, \mu_n(\boldsymbol{\theta}), v_n(\boldsymbol{\theta}) + \sigma^2) \tag{20}$$

where Φ is the Gaussian CDF. This unnormalized likelihood approximation can then be used with standard MCMC algorithms to arrive at a sample-approximation of the posterior.

We performed this procedure to infer the posterior distribution of the transmission rate parameter of the Tuberculosis model from Section 3.2 (see Fig. 9). Using only 1000 model simulations we were able to obtain an estimate of the posterior distribution comparable to the results of the classical ABC methods (which need a considerably larger amount of simulations) The KL divergence w.r.t. the Rejection ABC posterior was 0.23. Thus, BOLFI can be a highly effective tool when the simulations are costly. It however still depends on a good choice of summary statistics and a distance measure, which are central open problems in the ABC literature.

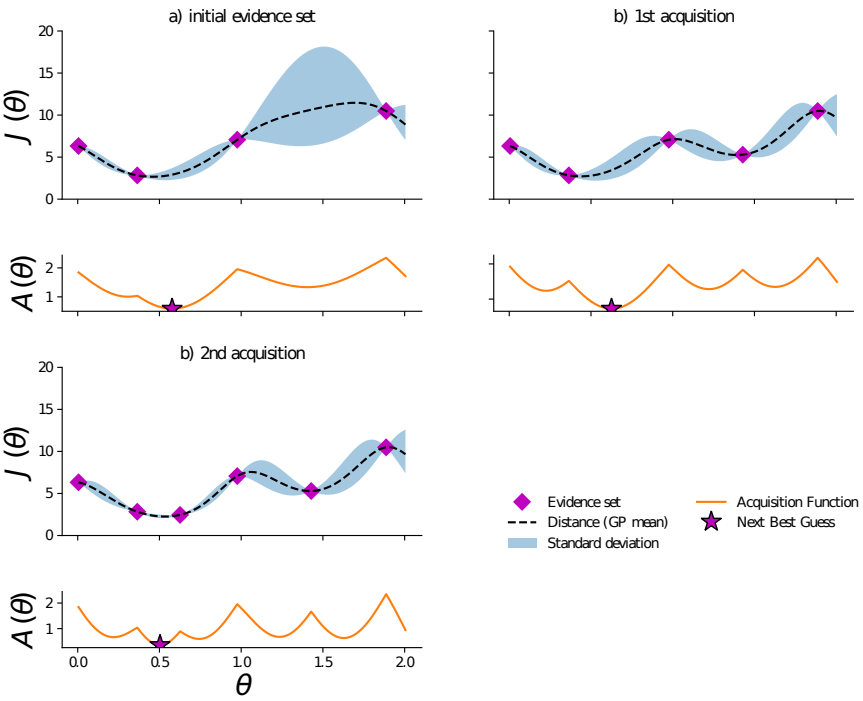


Figure 8: 2 iterations of Bayesian Optimization to estimate the transmission rate parameter (here denoted θ instead of α) of the Tuberculosis model defined in Section 3.2. The GP mean and its standard deviation are shown in the upper panel of each plot. **a)** Starting with an initial evidence set of 4 data points sampled from a uniform prior, **b)**, **c)** 2 additional data points are acquired with the LCB acquisition rule. The acquisition function is displayed in the lower panel of each plot. The parameter value which maximizes this acquisition function (and thus minimizes the LCB criterion) is added to the evidence set. Then, the Gaussian process is updated by conditioning on the new evidence set, according to equation 16. The acquisition rule trades off exploitation (low posterior mean, e.g. **b)**) and exploration (high posterior variance, e.g. **c)**).

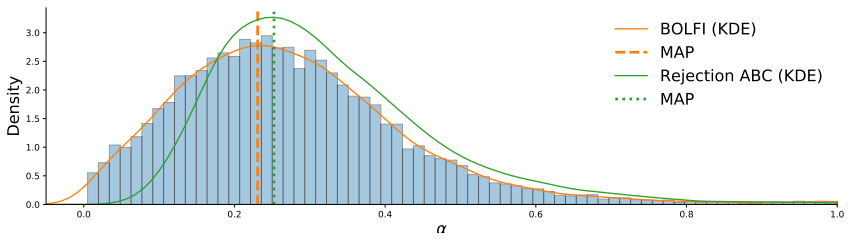


Figure 9: The posterior distribution for the transmission rate parameter α of the Tuberculosis model inferred with BOLFI. The observed dataset was the same as in Section 3.2, generated with $\alpha = 0.2$. We performed 1000 iterations of Bayesian Optimization with an initial evidence set of size 10. A non-parametric likelihood approximation was computed from the GP model according to (20) with $\epsilon = 0.02$. We then ran 4 MCMC chains to obtain 10,000 posterior samples.

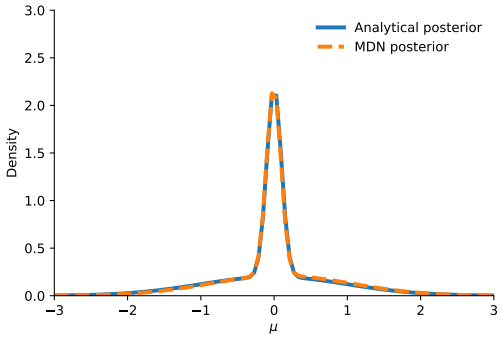


Figure 10: Mixture of 2 Gaussians: Correct analytical posterior and posterior obtained by Regression ABC with MDN.

6 Regression ABC

In a line of recent developments, methods based on post-hoc regression adjustments have gained traction. These methods (usually referred to as Regression ABC) circumvent defining summary statistics and distance measures by approximating the posterior distribution directly. Instead of computing the distance between simulated and observed data, a regression model from simulated data $\boldsymbol{x}_{\boldsymbol{\theta}}^{(i)}$ to the parameters $\boldsymbol{\theta}^{(i)}$ is learned [e.g. Blum and François, 2010]

Papamakarios and Murray [2016] propose an approach based on Mixture Density Networks (MDNs). MDNs approximate probability distributions using mixture models (e.g. Gaussian mixture models) and learn the parameters of the mixture model as the outputs of a neural network [Bishop, 1994]. In the ABC framework, this means that the simulated data are fed into a neural network, whose outputs are the parameters of a mixture model. This enables computing a conditional density estimate $q(\boldsymbol{\theta} \mid \boldsymbol{x})$, which serves as an approximation of the posterior distribution when evaluated at the observed data $\boldsymbol{x}_{obs}$.

We implemented a simple Regression ABC algorithm that uses MDNs to infer the mean parameter of the Gaussian mixture example (Section 3.3). Our MDN has one hidden layer with 20 *tanh*-activated neurons. The dimensionality of the output layer is determined by the number of mixture components, which was two for this example. For each mixture component, the output layer contains a mean, standard deviation and weight unit. The weights are normalized across the mixture components using *softmax*-activations. We created a training set of 10,000 samples from a uniform prior and one corresponding data set of size 1 from the simulator model. After training the MDN with the Adam algorithm [Kingma and Ba, 2014] for 10,000 iterations with a learning rate of 0.001, the posterior was approximated very well (see Fig. 10).

We used a similar network architecture (with 3 mixture components instead of 2 and 50 hidden units instead of 20) to infer the posterior distribution of the transmission rate parameter from Section 3.2. Our results did not yield consistent results across multiple runs. We suspect that the optimization got stuck in bad local optima. As Papamakarios and Murray [2016] note, learning the posterior directly from the prior with a MDN is highly inefficient, since $q(\boldsymbol{\theta} \mid \boldsymbol{x})$ is estimated for all $\boldsymbol{x}$, while we are only interested in $q(\boldsymbol{\theta} \mid \boldsymbol{x}_{obs})$. Thus, adopting their procedure of first learning an approximation of the posterior with one mixture component and then using this approximation as a proposal prior to train the more complex model might make the inference more stable. Implementing this procedure, however, is beyond the scope of this paper.

In addition to being ϵ-free and not requiring summary statistics and a distance measure, the Regression ABC approach has a few advantages over standard ABC methods. First, similar to

BOLFI, it uses information from all samples, without discarding any of them. Second, it provides a parametric form of the approximate posterior, which enables further evaluations and decisions based on the posterior density. While Regression ABC methods dispose of the threshold parameter, they introduce additional hyperparameters like the number of mixture components, the number of hidden layers and hidden units, which influence the quality of the posterior approximation.

7 Discussion

In this paper, we provided an introduction to Approximate Bayesian computation, including the basic sampling-based algorithms. ABC methods are used for models, for which explicit likelihoods are not available. The major challenge for ABC methods is to achieve an accurate approximation of the posterior density while balancing computational cost. In classical ABC approaches, the accuracy is determined by the threshold parameter ϵ, which governs the allowed distance between simulated data and observed data. Another important problem is the proper choice of summary statistics that enable the comparison of datasets for cases where a direct comparison is (numerically or computationally) infeasible.

Rejection ABC uses the most simple and inefficient approach by drawing independent samples from the prior and comparing their simulated outputs to the observed data under the threshold ϵ. In Section 2 we presented two methods (MCMC and SMC), which use more sophisticated sampling schemes that make use of previous samples in order to save simulations and therefore computation.

In Section 3 of this paper, we systematically compared these different sampling schemes using three different toy examples commonly found in the ABC literature.

The first example, a simple univariate Gaussian model, allowed us to compare the different algorithms to an analytically tractable posterior. We saw the importance of the right choice of the threshold for ABC methods. A larger threshold introduces a bias in the ABC posterior and hinders sampling from the true posterior. On the other hand, a desirable small threshold might be infeasible due to limited computational resources as more samples – and potentially expensive simulations – are required. In general, one does not know the degree of the bias that is introduced by a certain choice of a threshold value. Therefore, an open question for ABC methods is how to set the threshold value and how to determine the quality of the posterior approximation – or to find methods that can work without such a threshold.

The second example, the Tuberculosis model, dealt with two important characteristics of ABC methods. First, having access to a simulator-based model with discrete output values allows for exact inference where no summary statistics are needed. The three ABC methods differed in the total number of simulations with MCMC ABC being three times more efficient than Rejection ABC.

Second, for discrete outputs where exact inference is infeasible or for continuous outputs, ABC methods rely on sufficient summary statistics. The crucial point is "sufficient", which is only known for simple standard distributions. For many applications and especially in the realm of ABC methods, we have no clue about the likelihood's shape or to which family of distribution it might belong. But there are methods to estimate the quality of a newly added summary statistic, for example the ratio of posteriors with and without a specific summary statistic (for more detail see Joyce and Marjoram [2008]). As we have seen, using summary statistics can greatly reduce the total number of simulations but at the same time can introduce a bias into the approximation. Depending on the end users goal – approximating the whole posterior distribution or only gaining a valid point estimate – even a non-sufficient summary statistic can be helpful. However, methods that avoid the need for summary statistics are preferable.

That different ABC methods not only differ with respect to the runtime but also the approxi-

mation's quality was the result of the third example. Only the most basic Rejection sampler has no hyperparameters (other than ϵ) but suffers from inefficiency. All other ABC methods try to overcome the inefficiency of the Rejection sampler but introduce new parameters (like step size and number of decreasing thresholds) in their algorithms that have to be fine-tuned and optimized for the specific problem at hand. There is no such thing like a best sampler for all scenarios, just different advantages and disadvantages one should consider and account for.

While more sophisticated sampling methods definitely improve upon the inefficiency of the Rejection ABC algorithm, they still have in common the need for an explicit threshold. All information from simulations that yield distances above this threshold is discarded. We have shown different methods that try to address this. The ABCDE algorithm approaches this problem by treating the threshold as a latent variable, for which a reasonable value is inferred. Then the threshold is fixed and an approximation of the posterior is estimated. Its results are, however, very susceptible to the right choice of hyperparameters and do not improve upon existing sampling-based methods such SMC and MCMC. BOLFI, on the other hand, introduces a probabilistic model of the relationship between the parameters and the distances and chooses the next parameter value via Bayesian Optimization. Thus, instead of discarding any samples, it learns from them. When simulations are expensive, BOLFI can be a very efficient tool for ABC because it requires only a very small number of simulations. For cheap simulator models, BOLFI can be more inefficient than classical ABC methods, since it frequently performs the expensive operation of fitting a Gaussian Process.

Even though these methods overcome the necessity of a fixed threshold, a central problem of ABC remains: One still needs to define summary statistics and a distance function that capture the properties of the model well. Regression ABC methods address this difficulty. Instead of comparing observed to simulated data, they learn a regression model from the data to the simulator's parameters. This model can be used to compute a parametric approximation of the posterior distribution, which can be very helpful for further evaluations. Most importantly, their abandonment of the threshold parameter ϵ makes Regression ABC methods a promising development in the field of likelihood-free inference.

Classical ABC methods use a nonparametric auxiliary likelihood as a replacement for the intractable likelihood. Another very similar approach is called Bayesian synthetic likelihood (BSL) presented by Price et al. [2017] and based on Wood [2010]. In BSL the distribution of a set of summary statistics is approximated by a multivariate normal distribution, and its parameters are estimated by simulating n i.i.d. data sets from the model based on $\boldsymbol{\theta}$ and fitting the auxiliary likelihood to the summary statistics $\boldsymbol{s}_{1:n}$. Thus, BSL relies on the existence of summary statistics. BSL only relies on a single hyperparameter n, which is the number of simulations of the model used to estimate the parameters of the multivariate normal.

Like the threshold in ABC methods, in BSL the choice of n determines the closeness of the approximation to the ideal target. Interestingly, Price et al. [2017] report that the BSL target is "remarkably insensitive to n" compared to the more problematic choice of ϵ for ABC methods. Another finding is that classical ABC is more efficient for one-dimensional summary statistics, equally efficient for two-dimensional summary statistics and "significantly less efficient as d [dimensionality of the summary statistics] increases beyond 2". However, if the distribution of the summary statistics does not follow a normal distribution and is highly irregular, the output of BSL cannot be trusted. The authors give some hints of how to combine the SMC approach with BSL to obtain an algorithm which adaptively selects the value of n.

We have seen throughout this work that finding good summary statistics is a major concern of ABC methods and that some approaches like Regression ABC try to circumvent this problem altogether. Another approach is presented by Gutmann et al. [2017] and is based on classification instead of summarizing the data. The main idea is that two data sets generated with very different

values of $\boldsymbol{\theta}$ should be easier to distinguish from each other than two data sets generated with very similar values of $\boldsymbol{\theta}$. In fact, if two data sets are generated with the same value of $\boldsymbol{\theta}$, then distinguishing both should not be possible better than with mere chance-level. Given an augmented dataset $\mathcal{D}_{\boldsymbol{\theta}}$ with observed and simulated data and some binary label and a classification rule h that maps each feature vector $\boldsymbol{u}$ to its class label $h(\boldsymbol{u}) \in \{0, 1\}$, one can compute the classification accuracy which is the proportion of correct assignments.

Although the optimal Bayes classification rule $h^*_{\boldsymbol{\theta}}$ is not available, one can use any approximation of $\hat{h}_{\boldsymbol{\theta}}$ like LDA, QDA, SVM or many other. In this context, the best parameter value $\hat{\boldsymbol{\theta}}$ is the one which minimizes the average classification accuracy. Gutmann et al. [2017] showed that this approach "yielded accurate posterior inferences and that it defines a consistent estimator".

One thing that the ABC field is still lacking are methods that allow the comparison of different algorithms. We propose a few steps that address this problem. First, some examples are encountered again and again in the ABC literature, e.g. the Tuberculosis [Tanaka et al., 2006], Ricker [Wood, 2010], and Lotka-Volterra [Papamakarios and Murray, 2016] models. However, the implementation details (like the choice of summary statistics or distance and kernel functions) are seldom explicitly stated. We therefore argue for the need of standard implementations of common examples as well as standard implementations of the common ABC algorithms. Second, in order to evaluate new algorithms, a baseline is necessary. In the case of toy examples like the mixture of Gaussians, where a closed-form posterior is known, the choice of baseline is self-evident. For the problems that actually warrant ABC methods, the choice is not so obvious. Should a closed-form likelihood approximation be used? Should a Rejection ABC solution with a very high number of samples and a very small threshold value be used? These approaches come with high computational costs every time a new algorithm needs to be evaluated. Thus, it would be convenient if baseline posterior distributions for commonly used ABC examples were available – either as a sample-based representation or in a functional form (e.g. density estimates). Third, it is not obvious how results from different ABC algorithms should be compared to a baseline solution. At one time the result is a sample-based approximation of a posterior distribution with different amounts of samples (classical ABC) and at another time it is a parametric density (Regression ABC). Is it sufficient if the mean and variance are in the same ballpark? Should KL divergences or other measures for goodness-of-fit [e.g. Chwialkowski et al., 2016] be used? These are still open questions, which need to be addressed to allow for valid comparisons between algorithms for likelihood-free inference.

A Implementation

Our Python implementations of the Rejection-, MCMC-, SMC-ABC, ABCDE, BOLFI and Regression ABC algorithms can be found at our github repository. All presented results were obtained using our own implementations, except for Section 5, where we used ELFI [Lintusaari et al., 2017b], because it yielded more reliable results. In addition to the examples presented in this paper, we implemented some of the most common examples in the ABC literature:

- Ricker model [Wood, 2010]
- Tuberculosis [Tanaka et al., 2006]
- Mixture of Gaussians [Sisson et al., 2007]
- 20-dimensional Gaussian [Turner and Sederberg, 2012]
- REM (Episodic Memory) model [Turner and Zandt, 2012]

We also provide a template for implementing examples for ABC (including simulator, summary statistics and distance functions), which is easy to extend and import.

References

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1/2):5–43, 2003. ISSN 08856125. doi: 10.1023/A:1020281327116. URL http://link.springer.com/10.1023/A:1020281327116.

Mark A. Beaumont, Wenyang Zhang, and David J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002. ISSN 00166731. doi: GeneticsDecember1, 2002vol.162no.42025-2035. URL http://www.genetics.org/content/genetics/162/4/2025.full.pdf.

Christopher M Bishop. Mixture Density Networks. 1994. URL http://www.ncrg.aston.ac.uk/.

Michael GB Blum. Regression approaches for Approximate Bayesian Computation. 7 2017. URL http://arxiv.org/abs/1707.01254.

Michael GB Blum and Olivier François. Non-linear regression models for Approximate Bayesian Computation. *Stat Comput*, 20:63–73, 2010. ISSN 09603174. doi: 10.1007/s11222-009-9116-0. URL http://dx.doi.org/10.1007/s11222-009-9116-0.

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A Kernel Test of Goodness of Fit. 2 2016. URL http://arxiv.org/abs/1602.02964.

Dennis D. Cox and Susan John. SDO: A statistical method for global optimization. *Multidisciplinary Design Optimization*, pages 315–329, 1997. doi: 10.1109/ICSMC.1992.271617. URL http://ieeexplore.ieee.org/document/271617/.

Michael U. Gutmann and Jukka Corander. Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models. *Journal of Machine Learning Research*, 17:1–47, 2016. ISSN 15337928. doi: arXiv:1501.03291v3. URL http://arxiv.org/abs/1501.03291.

Michael U. Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Likelihood-free inference via classification. *Statistics and Computing*, pages 1–15, 2017. ISSN 15731375. doi: 10.1007/s11222-017-9738-6.

W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444. doi: 10.1093/biomet/57.1.97.

Marko Järvenpää, Michael U. Gutmann, Aki Vehtari, and Pekka Marttinen. Efficient acquisition rules for model-based approximate Bayesian computation. *ArXiv*, 4 2017. URL http://arxiv.org/abs/1704.00520.

Paul Joyce and Paul Marjoram. Approximately Sufficient Statistics and Bayesian Computation. *Statistical Applications in Genetics and Molecular Biology*, 7(1), 2008. ISSN 1544-6115. doi: 10.2202/1544-6115.1389. URL https://www.degruyter.com/view/j/sagmb.2008.7.1/sagmb.2008.7.1.1389/sagmb.2008.7.1.1389.xml.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 12 2014. URL http://arxiv.org/abs/1412.6980.

Jarno Lintusaari, Michael U. Gutmann, Ritabrata Dutta, Samuel Kaski, and Jukka Corander. Fundamentals and recent developments in approximate Bayesian computation. *Systematic Biology*, 66(1):e66–e82, 2017a. ISSN 1076836X. doi: 10.1093/sysbio/syw077.

Jarno Lintusaari, Henri Vuollekoski, Antti Kangasrääsiö, Kusti Skytén, Marko Järvenpää, Michael Gutmann, Aki Vehtari, Jukka Corander, and Samuel Kaski. ELFI: Engine for Likelihood Free Inference. 8 2017b. URL http://arxiv.org/abs/1708.00707.

Jun S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2001. ISBN 0387952306.

Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavare. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 100(26):15324–8, 2003. ISSN 0027-8424. doi: 10.1073/pnas.0306899100. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=307566&tool=pmcentrez&rendertype=abstract.

George Papamakarios and Iain Murray. Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation. 5 2016. URL http://arxiv.org/abs/1605.06376.

L F Price, C C Drovandi, A Lee, and D J Nott. Bayesian Synthetic Likelihood, 2017. ISSN 15372715.

Jonathan K. Pritchard, Mark T. Seielstad, Anna Perez-Lezaun, and Marcus W. Feldman. Population growth of human Y chromosomes: A study of y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 12 1999. ISSN 07374038. doi: 10.1093/oxfordjournals.molbev.a026091. URL http://www.ncbi.nlm.nih.gov/pubmed/10605120.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning.*, volume 14. 2006. ISBN 026218253X. doi: 10.1142/S0129065704001899. URL http://www.gaussianprocess.org/gpml/chapters/RW.pdf.

S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2 2007. ISSN 0027-8424. doi: 10.1073/pnas.0607208104. URL http://www.pnas.org/cgi/doi/10.1073/pnas.0607208104.

Jasper Snoek, Hugo Larochelle, and Rp Adams. Practical Bayesian Optimization of Machine Learning Algorithms. *Nips*, pages 1–9, 2012. ISSN 10495258. doi: 2012arXiv1206.2944S. URL https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.

Rainer Storn and Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. ISSN 1573-2916. doi: 10.1023/A:1008202821328. URL http://dx.doi.org/10.1023/A:1008202821328.

Mark M. Tanaka, Andrew R. Francis, Fabio Luciani, and S. A. Sisson. Using approximate bayesian computation to estimate tuberculosis transmission parameters from genotype data. *Genetics*, 173(3):1511–1520, 2006. ISSN 00166731. doi: 10.1534/genetics.106.055574.

Simon Tavaré, David J Balding, R. C. Griffiths, and Peter Donnelly. Inferring coalescence times from DNA sequence data, 1997. ISSN 00166731. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1207814/pdf/ge1452505.pdf.

Brandon M. Turner and Per B. Sederberg. Approximate Bayesian computation with differential evolution. *Journal of Mathematical Psychology*, 56(5):375–385, 2012. ISSN 00222496. doi: 10.1016/j.jmp.2012.06.004. URL http://dx.doi.org/10.1016/j.jmp.2012.06.004.

Brandon M Turner and Trisha Van Zandt. A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, 56:69–85, 2012. URL http://www.elsevier.com/copyright.

Richard David Wilkinson. Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical Applications in Genetics and Molecular Biology*, 12 (2):129–141, 2013. ISSN 15446115. doi: 10.1515/sagmb-2013-0010.

Simon N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 8 2010. ISSN 00280836. doi: 10.1038/nature09319. URL http://www.nature.com/doifinder/10.1038/nature09319.