An introduction to Approximate Bayesian Computation for epidemics.

Dissertation for MSc. in Statistics 2017-2018

Benjamen Simon
32102717

Supervisor: Prof. Peter Neal

Department of Mathematics & Statistics
Lancaster University
September 7, 2018

Contents

Abstract

The modelling of infectious diseases is vital for their understanding and control. Infectious disease data, however, are often characterised by high dependence and missing information, which make inference on models a difficult task. Even advanced statistical techniques such as Monte Carlo Markov Chain (MCMC) can sometimes fall prey to these challenges and become highly inefficient. In such cases, we require an alternative method for making inference on these models. One such alternative could be Approximate Bayesian Computation (ABC), a likelihood-free (sometimes called simulation) method of inference. In this dissertation we present a series of ABC methods which can be used to make *approximate* inference for epidemic models, and demonstrate their effectiveness on a series of simple case study outbreak datasets. Following this, we apply a selection of the methods to a complex spatio-temporal outbreak dataset for Sugarcane Yellow Leaf Virus. Finally, we discuss some questions that have arisen throughout the dissertation on the use of ABC, and give our judgement on its position as an alternative to MCMC for making inference on epidemic models, given our current knowledge.

---

1. Introduction

The understanding of infectious diseases is vital for their control and, ideally, eradication. The World Health Organisation (WHO) estimates vector-borne diseases alone cause more than $700,000$ deaths annually, and they only account for 17% of human infectious diseases[33]. Statistical modelling has been utilised in the past 20 years to great effect to understand the spread of infectious disease[19], and is now a valuable tool in the arsenal for the fight against disease[15].

There are many ways to model infectious diseases, ranging from simple models that assign an individual a state and model the transitions between the states, to complex agent-based models where every individual is unique, with their own attributes and contact networks. As with all modelling, however, parsimony is a virtue. While a more complex model may imitate reality better than a simple one, it becomes much more difficult to fit the models to data, identify parameter values, and take inference[15]. Fitting these models can be a complex task in even the simplest of cases, due to the nature of infectious disease data. Infectious disease data stand apart from non-communicable diseases because it is both highly dependent and only partially-observable. By this we mean that we cannot observe the transmission process of an infectious disease, only the outcome; we can see who is infected, but not who infected them or when. This missing information often makes computing the likelihood very difficult, and more often than not impossible. In these cases we often turn to advanced statistical methods, such as Monte Carlo Markov Chain (MCMC), to perform Bayesian inference to obtain posterior distributions for the parameters of the model. These methods have revolutionised the analysis of partially observed infectious disease data, and have been successfully applied to a myriad of diseases[19] such as Foot-and-mouth[10] and SARS[20]. Unfortunately these methods do not come without their problems. Firstly, non-standard and problem-specific algorithms have to be designed in each instance to improve efficiency[19], and when the models become more complex, or the population too large, the cost of computing the likelihood can become very high[19]. The algorithms make use of common techniques such as data augmentation to impute unobservable aspects of the data, such as infection times. Sometimes the dependence in these augmented data can be unacceptably high, and as a result the efficiency drops to a point where the algorithm is unusable. In these

cases we are in need of an alternative method for making inference for infectious diseases. One such possible alternative is known as Approximate Bayesian Computation (ABC).

ABC is a likelihood-free method of inference, also known as a simulation method. ABC can provide us with *approximate* inference for the posterior of the model parameters. The concept is very simple. We draw model parameters from a prior distribution, and use them to simulate an outbreak. This is relativly easy to do in the case of infectious diseases, even when the model is complex or the data is only partially observed. If our simulated outbreak is "similar" to our data, then we accept the parameter draws as samples from the *approximate* posterior distribution. The success of ABC methods for epidemic data has been demonstrated on many occasions including for; HIV-Aids in Cuba[6], Equine Influenza[1], Ebola in the Democratic Republic of Congo[32], and Bovine Tuberculosis in both cattle, UK[8], and lions, South Africa[18]. Other sources show that a range of ABC methods work for selection of disease outbreaks[19;22;23].

In this dissertation we will go through a selection of different ABC methods, explain how they work, and then show how they can be applied to some simple datasets. We will then take everything we have learnt and apply it to a complex spatio-temporal outbreak dataset for Sugacane Yellow Leaf Virus. In §2 we explain a selection of models we use for the outbreaks, and how we can simulate them. In §3 we detail a series of case study datasets we will be using to test the ABC algorithms presented throughout this dissertation. A general overview of ABC is given in §4. Then, in §5 to §9, we describe a selection of ABC methods, providing algorithms in order to implement them, and showcasing them on a series of case study datasets. After this we describe a complex spatio-temporal epidemic of Sugarcane Yellow Leaf Virus (SCYLV) in §10, before applying some of the ABC methods to make inference on our model in §11. We detail some possible extensions to our analysis of the SCYLV data in §12 that time did not permit us to undertake. Finally we discuss our thoughts on the methods presented throughout this dissertation in §13, addressing some questions that have been raised, suggesting future work to be undertaken, and giving our current judgements on the use of ABC for inference on epidemics.

2. How do we simulate the data?

There are a multitude of ways to model the spread of disease through a susceptible population, from simple state-transition models such as the well known S-I-R model, to household models, and agent based network models. The type of model we use depends very much on the disease in question, the situation, and the type and detail of data we have available. In this section we will go through some of the more fundamental models that we will later utilise to demonstrate various ABC methods for different datasets.

2.1. S-I-R construction.

One of the simplest ways to model an epidemic is the S-I-R construction[17]. The S-I-R construction involves taking a closed population of N individuals and dividing them into three independent sets;

- S - Susceptible. Individuals in this set are susceptible to infection, and will become infected when they come into contact with an infected individual.
- I - Infected/infectious. Individuals in this set are infected, and in the simplest case, also infectious. If these individuals come into contact with a susceptible individual they will infect them. Individuals will remain in state I until they recover.
- R - Recovered/removed. Individuals in this set have recovered from being infected, have no affect on either of the other two sets if they come into to contact with them, cannot become infected again, and will remain in the recovered state indefinitely. Recovery grants immunity.

The population mixes homogeneously (everyone has equal chance of coming into contact with everyone else), and the S-I-R construction describes the process through which the individuals move between the sets. The process could be considered from a purely deterministic point of view, in which case we could model the transitions between the states at each time-step using differential equations. Let $\beta \geq 0$ be the rate of infection, and $\gamma \geq 0$ be the rate of recovery. For instance, β can be thought of as the proportion of the susceptible population that become infected each time-step due to each infected individual. Then,

$$\frac{dS}{dt} = -\beta S(t)I(t), \qquad \frac{dI}{dt} = \beta S(t)I(t) - \gamma I(t), \qquad \frac{dR}{dt} = \gamma I(t),$$

where $S(t), I(t)$, and $R(t)$ are the number of susceptible, infected, and removed individuals at time-step t, respectively.

Epidemics, however, are not deterministic processes, and while the deterministic construction may give us some insights, it will miss a lot of the subtleties. For instance, given two independent epidemics with the same starting conditions, same rates, and the same assumptions, one may infect a large proportion of the population and one may die off instantly, with probabilities dependent on the initial conditions and parameters. The deterministic construction does not account for both of these outcomes at once, and so it is necessary to use a stochastic construction of the S-I-R process. One of the simplest and easiest to implement is known as the Chain-Binomial model.

2.2. **Chain-Binomial.** The Chain-Binomial[5] algorithm is a simple discrete-time, stochastic SIR model. It works as presented in Algorithm 1.

When demonstrating this algorithm, we will choose β and γ such that the value of the basic reproductive number, R_0, lies between 1 and 3, calculated using the equation $R_0 = \frac{N\times\beta}{(1-\exp(-\gamma))}$. This suggests a density dependent model. We will choose an Exponential prior with mean 1 for both β and γ when simulating further outbreaks, which is equivalent to putting a Uniform[0,1] prior on the infection and removal probabilities, p_{inf} and p_{rem}.

2.3. **The Gillespie Algorithm.** The Gillespie algorithm[14] is the stochastic version of the S-I-R construction. It is a relatively simple continuous time model for simulating an epidemic. We again start with an initial population N which is segregated into 3 sets, S-susceptible, I-infected, and R-removed, with $S + I + R = N$. At each time step, t, we know the probability of an infection event, p_{inf}, and of a removal event, p_{rem}. The probabilities depend on the number individuals in each set, as well as the infection rate, β, and the removal rate, γ. Thus at each time step, t, an event will occur, and we draw whether it is an infection or removal event using their probabilities. If it is an infection event increase the infectious population by one, and if it is removal event reduce the infectious population by one. We continue this process until the infectious population $I = 0$.

The rate at which infection events occur is given by $r_{inf} = \frac{\beta}{N}SI$, and the rate at which removal events occur is given by $r_{rem} = \gamma I$. Thus the time until the next event of any type occurs, τ, is Exponentially distributed with rate $\left(\frac{\beta}{N}SI + \gamma I\right)$. The probability of that event being an infection event is then given by $p_{inf} = \frac{\frac{\beta}{N}SI}{\frac{\beta}{N}SI+\gamma I}$. Similarly the probability of that event being a removal event is then given by $p_{rem} = 1 - p_{inf} = \frac{\gamma I}{\frac{\beta}{N}SI+\gamma I}$. The algorithm is given in Algorithm 2.

When demonstrating this algorithm, we will again choose β and γ such that the value of the basic reproductive number, R_0, lies between 1 and 3, calculated this time using the equation $R_0 = \frac{N\times\beta}{\gamma}$. This again suggests a density dependent model. We will continue

Chain-Binomial model:

Inputs: Population size, N; Infection rate, β; Removal rate, γ.

1. Initialise the process by stating the number of susceptible (S), infected (I), and removed (R) individuals, with $S + I + R = N$ at all times. We usually start with $S = N - 1$, $I = 1$, and $R = 0$.
2. Begin at time-step t = 1 with the above, then,
   (a) Draw the number of newly infected individuals in time-step $(t + 1)$, I^*, using
   $$I^* \sim \text{Binomial}\left(S(t), p_{inf}\right),$$
   where $S(t)$ and $I(t)$ are the number of susceptible and infectious individuals available at the end of time-step t, respectively. The probability that a susceptible individual becomes infected is given by $p_{inf} = 1 - e^{-\beta I(t)}$, where $\beta \geq 0$ is the infection rate of the process.
   (b) Draw the number of newly removed individuals in time-step $(t + 1)$, R^*, using
   $$R^* \sim \text{Binomial}\left(I(t), p_{rem}\right),$$
   where $I(t)$ is the number of infected individuals available at the end of time-step t. The probability that an infected individual is removed is given by $p_{rem} = 1 - e^{-\gamma}$, where $\gamma \geq 0$ is the removal rate of the process.
   (c) Update the states using
   $$\begin{aligned} S(t+1) &= S(t) - I^*, \\ I(t+1) &= I(t) + I^* - R^*, \\ R(t+1) &= R(t) + R^*, \end{aligned}$$
   and set $t = t + 1$.
3. Run the process for the T time-steps, or until the infected state reaches size zero.

Algorithm 1

to choose an Exponential prior with mean 1 for both β and γ when simulating further outbreaks, which is equivalent to putting a Uniform[0,1] prior on the infection and removal probabilities, p_{inf} and p_{rem}.

3. The Case Studies

To demonstrate the various Approximate Bayesian Computation algorithms presented in this dissertation, we will use a number of different case studies which exemplify different features an epidemic could take on. In this section we will describe three case study datasets we use to demonstrate the methods, which will be referred to throughout this dissertation.

3.1. **Smallpox Outbreak, Abakaliki, 1967.** The Abakaliki dataset (see[31], page 125) describes an outbreak of Smallpox in the small village of Abakaliki, Nigeria in 1967. The data set consists of $m = 30$ infected individuals out of a total $N = 120$ susceptible individuals. The data also contains temporal information on when each infected individual was detected (often taken to be the recovery time), but we will often just use the final size data. The detection times are: $(0, 13, 20, 22, 25, 25, 25, 26, 30, 35, 38, 40, 40, 42, 42, 47, 50, 51, 55, 55, 56, 57, 58, 60, 60, 61, 66, 66, 71, 76)$.

Gillespie algorithm:

Inputs: Population size, N; Infection rate, β; Removal rate, γ.

1. Initialise the process by stating the number of susceptible (S), infected (I), and removed (R) individuals, with $S + I + R = N$ at all times. We usually start with $S = N - 1$, $I = 1$, and $R = 0$.
2. Begin at time-step t = 1 with the above, while $I > 0$ or $t < T$.;
   i. Draw the time until the next event, $\tau \sim Exp\left(\frac{\beta}{N}SI + \gamma I\right)$.
   ii. Draw $u \sim Unif[0, 1]$.
   iii. If $u < \frac{\frac{\beta}{N}SI}{\frac{\beta}{N}SI+\gamma I}$, then
   · set $S = S - 1$; $I = I + 1$,
   otherwise,
   · set $I = I - 1$; $R = R + 1$.
   iv. Set $t = t + \tau$, and record (t, S, I, R).

Algorithm 2. The algorithm presented here is adapted from Kypraios et al. (2016)[19].

3.2. **Gastroenteritis outbreak, South Carolina, 1996.** The Gastroenteritis dataset describes an outbreak of Gastroenteritis in South Carolina in January 1996, original reported by Caceres et al. (1998)[9]. While Gastroenteritis is commonly spread by contaminated food, in this case, person-to-person spread is believed to have occurred[?]. Of the $N_s = 89$ staff working on the ward during the study period, there were $m_s = 28$ cases, and of the $N_p = 91$ patients who were hospitalised for more than one day during the outbreak, there were $m_p = 10$ cases. The data contains information on the date of the onset of symptoms for all cases.

Following Britton et al. (2002)[7] and Kypraios et al. (2016)[19], we restrict our attention to only the staff population, as the patient population is open and there were relatively few cases. The outbreak lasted for a total of 7 days, and the number of cases on each day, and all days thereafter, is given by $(1, 4, 2, 3, 3, 10, 5, 0)$.

3.3. **Measles outbreak, Honkajoki, 1989.** The Measles dataset describes an outbreak of Measles in a school in Honkajoki, Finland in 1989[26]. There are three types of students, with a type $k \in (0, 1, 2)$ student having received k doses of Measles vaccination. The data is final size data, and is summarised in Table 1.

Vaccination status, k	0	1	2
Total number of infected individuals, m_k	18	11	6
Total number of individuals, n_k	79	189	149

Table 1. A summary of the Measles dataset described in §3.3.

4. What is Approximate Bayesian Computation?

Real data often come with many complications which make analysis difficult and as such require the use of advanced statistical methods, this is doubly true for epidemic data. In such cases, a widely used tool is Monte Carlo Markov Chain (MCMC), which works well in many instances. In cases, however, that require a great amount of data augmentation,

the MCMC method and algorithms can become quite ineffective. Approximate Bayesian Computation (ABC) is a collection of methods that can be viewed as an alternative to MCMC and other such methods.

The idea behind ABC is deceptively simple. Assume we have data that is particularly complex to analyse, but the process we believe it was generated from is reasonably simple to simulate. Then, if we can simulate a sufficiently similar set of data using that process, the parameters that were used to generate the simulation can be seen as draws from the approximate posterior distribution for the parameters that generated the data.

Epidemics in particular are reasonably easy to simulate. In the simplest case, we merely need to know the number of susceptible, infected, and recovered individuals at each time step, and the probability of moving between each set. On the other hand, the data for epidemics is often very difficult and expensive to obtain, and is often characterised by missing information. For instance, the epidemic process is only partially observable, meaning for example, that we can see who was infected, but not necessarily when they were infected or who infected them. This means that MCMC methods require large amounts of data augmentation, the excess of which leads to high levels of dependency between the accepted draws, which greatly reduces the efficiency of the algorithm. Thus we can see the possible usefulness of ABC methods for epidemics.

5. Exact Bayesian Computation

We begin by looking at Exact Bayesian Computation (EBC), which can provide us with exact draws from the posterior distribution of θ. In this section we go over the ideas behind EBC and present an algorithm for implementing it. We then look at some features of the EBC, some simple non-epidemic examples where EBC can be applied, and its usefulness in the epidemic context.

In EBC, we draw parameter values from a prior distribution and simulate the data using a model that we believe it could be generated from. If the simulation exactly matches the data, we accept the draw as an exact draw from the posterior, otherwise we reject it. Thus an algorithm for the EBC is presented in Algorithm 3.

Exact Bayesian Computation (EBC):

Inputs: Data, $\mathbf{X}$; Model, $\mathsf{M}(\theta)$; Prior on θ, $\pi(\theta)$.

1. Draw a realisation, θ^*, of the parameters from their prior distributions $\pi(\theta)$.
2. Simulate an outbreak, χ, using the model $\mathsf{M}(\theta)$ and the parameter draw θ^*.
3. If $\chi = \mathbf{X}$ then accept the parameters θ^* as an exact draw from the posterior distribution for θ, and record θ^*, otherwise reject θ^*.
4. Repeat steps 1-5 until we have T accepted draws from the approximate posterior distribution for θ.

Algorithm 3. The algorithm presented here is adapted from Kypraios et al. (2016)[19].

The EBC algorithm can provide us with independent samples from the exact posterior distribution for θ, which theoretically makes it a very useful tool, arguably even better than MCMC in one sense since the samples are not dependent. However, because we have to match the data exactly, the EBC can only be used with discrete data, as the acceptance probability with continuous data will be 0. Secondly, the more data we have to match, the smaller the acceptance probability becomes, and it quickly becomes prohibitively small.

Thus, in practice, Approximate Bayesian Computation, which only provides approximate draws from the posterior distribution for θ, is often the preferred choice.

Example 1 (Acceptance Probability of the EBC)**.** We mentioned that the acceptance rate of the EBC can be prohibitively small for use in practice. For instance, assume that we generate an outbreak using the simple chain-binomial algorithm given in Algorithm 1. Let this outbreak begin with 1 infected individual ($m = 1$) and 29 susceptible individuals (so $N = 30$), and we let $\beta = \frac{2}{30}$ and $\gamma = 1$, so that $R_0 \approx 3.16$ using the equation given in §2.2. For a particular epidemic that we generated using these settings, a total of 19 individuals out of 30 became infected. The states at each time-step are given in Table 2. The likelihood of generating this exact epidemic (with infections and recoveries at the correct times) under these conditions was calculated to be 0.000000001781 or 1.781×10^{-09}. Assuming that we fix these parameter values for β and γ, it would take around 500 million more simulations to find even one accepted simulation. Now it is definitely possible for this outbreak to be generated by other sets of parameters, which means the acceptance probability of the EBC isn't quite this low, but it does not increase so much as to make the EBC a viable method for even this simple epidemic.

Time-step	1	2	3	4	5	6	7→
S	29	26	21	13	12	11	11
I	1	3	6	9	2	2	0
R	0	1	3	8	16	17	19

TABLE 2. The states of the simulated ourtbreak from Example 1 at each time-step.

For instance, we ran the EBC algorithm to try and find the posterior distribution for the parameters β and γ. We ran the algorithm using the same model the data was generated from, and made the mean of the priors equal to the true parameter values. This is a far more generous situation than we could ever hope for in reality, but even so we ran the algorithm for 36,545,423 simulations which took 9418 seconds (157 minutes) and we did not even find one accepted simulation. □

Sometimes, however, we do not have to match the entirety of the data exactly. In cases where there exists a sufficient statistic for the process, we can merely find simulations that exactly match the sufficient statistic of the data. Before we continue it is worth defining what a sufficient statistic is.

Definition 1 (Sufficient statistic[29])**.** *Let the random variables $X_1, \ldots, X_n$ have joint probability density function (pdf) given by $f(x_1, \ldots, x_n|\theta)$, that depends on parameter θ. The statistic $T(\mathbf{X}) = g(X_1, \ldots, X_n)$ is a sufficient statistic for θ if and only if the pdf can be factorised as*

$$f(x_1, \ldots, x_n|\theta) = \phi\left\{T(\mathbf{X})|\theta\right\} h(x_1, \ldots, x_n),$$

where $\phi(\cdot)$ is a function that depends on the data, $x_1, \ldots, x_n$, only through $T(\mathbf{X})$, and $h(x_1, \ldots, x_n)$ does not depend on θ. □

Example 2 (Sufficient statistic for a Poisson distribution)**.** For instance, assume that random variables $X_1, \ldots, X_n$ are independent and identically Poison distributed with mean λ, then the sum $T(X) = X_1 + \cdots + X_n$ is a sufficient statistic for λ. It may not be completely obvious at first why this is the case. First consider the joint probability

distribution,

$$\begin{aligned}\mathbb{P}\left[\mathbf{X}=\mathbf{x}\right] &= \mathbb{P}\left[X_1 = x_1, \ldots, X_n = x_n\right], \\ &= \mathbb{P}\left[X_1 = x_1\right] \ldots \mathbb{P}\left[X_n = x_n\right], \\ &= \frac{e^{-\lambda}\lambda^{x_1}}{x_1!} \cdots \frac{e^{-\lambda}\lambda^{x_n}}{x_n!} = \frac{e^{-n\lambda}\lambda^{x_1+\cdots+x_n}}{x_1! \times \cdots \times x_n!}, \\ &= \frac{1}{x_1! \times \cdots \times x_n!} e^{-n\lambda}\lambda^{T(\mathbf{x})},\end{aligned}$$

by independence. Then it can easily be seen that $\phi\left\{T(\mathbf{X})|\lambda\right\} = e^{-n\lambda}\lambda^{T(\mathbf{x})}$ and $h(x_1, \ldots, x_n) = \frac{1}{x_1! \times \cdots \times x_n!}$, so $T(\mathbf{X})$ is a sufficient statistic for λ. □

Unfortunately, sufficient statistics are often only available in simple cases with tractable likelihoods, which is often not the case with problems that require the use of ABC methods. It is worth noting that there are some simple epidemic models where sufficient statistics are available, but we will not be going into details about these[19].

6. Approximate Bayesian Computation

It is fairly clear at this point, that for most practical purposes, EBC is not a viable method. This, however, does not mean that the concepts that it employs are completely useless. If we were to be a bit more lenient in which simulations were acceptable, looking for approximate matches as opposed to exact matches, then we should be able to improve the acceptance rate, and even use these methods in cases with continuous data. This is the idea behind Approximate Bayesian Computation (ABC), which can provide approximate (as opposed to exact) draws from the posterior distribution for the parameters, θ. In this section we will explain what is needed to use the ABC method, and present an algorithm for it, before applying the method to a series of case study datasets.

6.1. ABC: The algorithm.

To use an ABC algorithm we will require extra information and tools compared the EBC. We will replace the sufficient statistics with a set of summary statistics, $S(\cdot)$, which we believe capture the qualities of the data that we are interested in well. We then require a distance function, $d(\cdot)$, to measure how different our simulations are to our data, based on their summary statistics. Finally we need a tolerance, ϵ, which tells us which simulations are acceptable in this circumstance, based on their distance.

Let us say we have data $\mathbf{X}$, which we believe to have been generated by a process $\mathsf{M}(\theta)$. We replace our sufficient statistics, $T(\mathbf{X})$, from the EBC, with summary statistics $S(\mathbf{X})$, and define the distance function $d\left(S(\mathbf{X}), S(\chi)\right)$, to measure how different the summary statistics of the data, $\mathbf{X}$, are from the summary statistics of a simulated observation, χ. We let the tolerance be ϵ. We present an algorithm for the ABC in Algorithm 4.

It makes sense that our choices for the prior distributions on θ, $\pi(\theta)$, the process assumed to generate the original data, $\mathsf{M}(\theta)$, the summary statistics, $S(\cdot)$, the distance function, $d(\cdot)$, and the tolerance, ϵ, all have a great impact on the efficiency and effectiveness of the ABC algorithm.

6.2. ABC: Examples.

To see the effectiveness of the ABC we again begin with an example of simulated data, before moving on to applying the methodology to the Abakaliki and Gastroenteritis datasets.

Approximate Bayesian Computation (ABC):

Inputs: Data, $\mathbf{X}$; Model, $\mathsf{M}(\theta)$; Prior on θ, $\pi(\theta)$; Summary statistics, $S(\cdot)$; Distance function, $d(\cdot)$.

1. Draw a realisation, θ^*, of the parameters from their prior distributions $\pi(\theta)$.
2. Simulate an outbreak, χ, using the model $\mathsf{M}(\theta)$ and the parameter draw θ^*.
3. Calculate the summary statistics, $S(\chi)$, for the simulated data, χ.
4. Using the distance function, $d(\cdot)$, calculate the distance between the summary statistics of the data and simulated observation, $d\left(S(\mathbf{X}), S(\chi)\right)$.
5. If $d\left(S(\mathbf{X}), S(\chi)\right) \leq \epsilon$ then accept the parameters θ^* as approximate draws from the posterior distribution for θ, and record θ^*, otherwise reject θ^*.
6. Repeat steps 1-5 until we have T accepted draws from the approximate posterior distribution for θ.

ALGORITHM 4. The algorithm presented here is adapted from Kypraios et al. (2016)[19].

Example 3 (ABC for simulated data)**.** Let the population consist of $N = 200$ individuals, and we will begin with $m = 1$ infected. We generated an outbreak using the Chain-Binomial algorithm (Alg. 1) with the parameter values $\beta = 0.0015$ and $\gamma = \frac{1}{7}$. This resulted in 157 infected individuals and lasted 79 days.

As stated, we require additional tools to be able to use the ABC, and we have to make certain choices for the form of these tools. In this instance, for the summary statistics, $S(\cdot)$, we choose the final size of the epidemic, M, and the time when the final infected individual recovers, $T_{(I=0)}$, also called the duration of the epidemic. These are common features of epidemics that might be of interest. Ideally we want a set of summary statistics that are correlated with the parameters of interest[22], though it may not always be obvious which summary statistics are best. As a minimum we need at least as many summary statistics as parameters in the model, but too many can also lead to greatly reduced acceptance rates, as well as distorting the approximation of the posterior[28].

For the distance function, $d(\cdot)$, we choose the L^2-norm= $\sqrt{(x-y)^2}$, and apply it to each summary statistic separately, meaning we have multiple tolerances. When considering which distance metric we use, the key feature is that it gives the correct weight to each summary statistics based on their importance. A summary statistics importance could be categorised by how well it infers the parameters we are interested in. It is also important to balance scale. For instance, if one summary statistic ranges from (0,1), and another has average values in the hundreds, then the second will dominate any distance function that combines the summary statistics equally and only requires one tolerance, unless they are rescaled.

The Chain-Binomial algorithm in this case is very easy and efficient to simulate, so as a preface to help us identify a suitable tolerance, ϵ, we generate 1 million simulations and identify different tolerances based on their acceptance rate and ability to identify the correct parameter values. We choose $Exp(1)$ priors for both β and γ, as this is mathematically justified as explained in §2.2, and would be our choice if we knew nothing about the parameters.

It took around 557 seconds to generate these 1 million simulations, and we present the results of the pilot run in Table 3 below.

Tolerance	(10,10)	(20,20)	(30,30)	(40,40)	(50,50)
Accepted samples	9	36	106	337	147871
Acceptance rate	0.0009%	0.0036%	0.0106%	0.0337%	14.7871%
$\mathbb{E}[\beta\|\mathbf{X}]$	0.001929	0.002132	0.002440	0.003072	0.979855
$\mathbb{E}[\gamma\|\mathbf{X}]$	0.2298	0.2434	0.2575	0.26972	0.15001

TABLE 3. A summary of the pilot run for the ABC for the simulated data. We are aiming for $\beta = 0.0015$ and $\gamma = 0.1429$.

In this special case we know the values of the parameters that we are aiming for, $\beta = 0.0015$ and $\gamma = 0.1429$. From Table 3 we can see that none of the tolerances we have selected give posterior means equal to these parameters, but we can see that the smaller the tolerance the closer we are. The error at the smaller tolerances may be in part due to the small number of samples, and the error at the larger tolerances is due to accepting more and more samples that look less and less similar to the real data. For instance, at the (50,50) tolerance, we would accept even outbreaks that infected 50 fewer individuals than the real data and last 50 time steps longer. Since the disease infected 157 individuals and lasted 79 days, one that infected 107 individuals and lasted 129 days would look rather different. Also note that the population size is 200, and a (50,50) tolerance will accept simulations that infect less than 207 individuals, so all parameter sets that infect the whole population between 29 and 129 time-steps will be accepted. This essentially means there is no upper limit on the β values that are accepted, as long as the γ values compensate by being smaller. We can see this evidenced in the results.

In the tolerances we have chosen to display, we have given equal weighting to both the summary statistics, but it may be that we believe that the number of infected individuals is better at isolating the correct posterior distributions than the epidemic duration, or vice versa. We can see from Figure 1 the distances of the most relevant samples which may help us pick tolerances, for instance, we can see if a small increase to a tolerance will make a large increase in the acceptance rate. As it stands, we believe that the final size will infer the ratio between β and γ, and the duration will infer the scale of the parameters, so we give equal weighting to both. The other benefit of a pilot run is that we get an initial idea for what the posterior mean may be, and we can update our prior with this information. For instance, we are currently using $Exp(1)$ priors for both β and γ, which assumes they have mean 1, however this pilot runs shows us that even at large tolerances, the mean of β is less than 0.005 and the mean of γ less than 0.25. Adopting these estimates into our prior should improve our acceptance rate.

We now wish to run our ABC algorithm with a chosen tolerance to obtain 1000 samples from the approximate posterior distribution for β and γ. We adopt the findings of the pilot run into our priors, and select an $Exp(2000)$ prior on β and an $Exp(4)$ prior on γ. Since this should improve our acceptance rate, we choose our tolerance to be (10,10). This ABC algorithm took 428,988 simulations to find the 1000 accepted samples, taking 263 seconds. The results are presented in Table 4. We can see instantly that updating our priors has made the acceptance rate jump from 0.0009% to 0.23%, which is over 250 times larger. With our knowledge of the true parameters, we can also see that the estimates of the posterior means are closer to the true estimates, still slightly higher, but well within one standard deviation for both β and γ. Overall the ABC seems have done a fine job of approximating the posterior distribution of β and γ in this case.

□

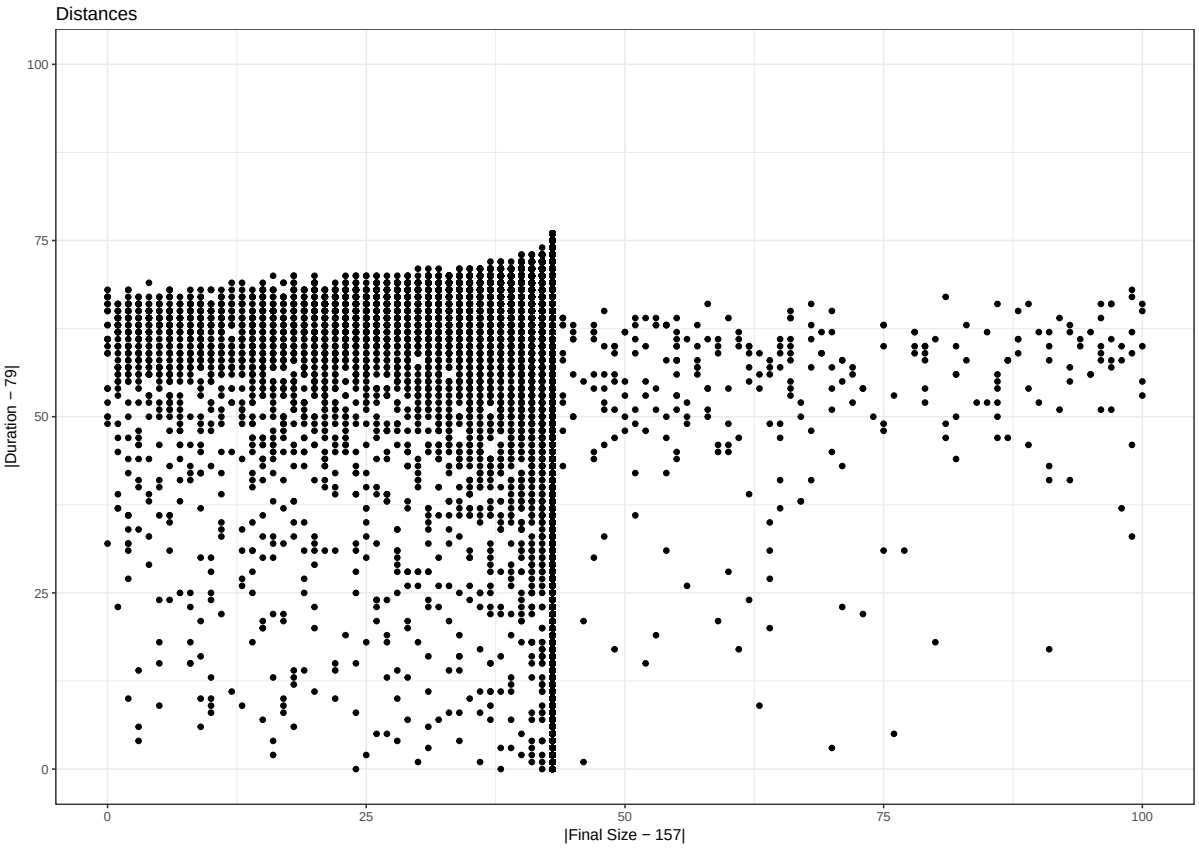


FIGURE 1. A plot of the distances for the best matching simualted outbreaks. We have chosen our tolerance to be (10,10)

$\mathbb{E}[\beta\|\mathbf{X}]$	$\text{sd}(\beta\|\mathbf{X})$	$\mathbb{E}[\gamma\|\mathbf{X}]$	$\text{sd}(\gamma\|\mathbf{X})$	Simulations	Acceptance rate	Time
0.0017	0.0004	0.1892	0.051	428,988	0.23%	263 seconds

TABLE 4. A summary of the ABC results for the simulated data. We are aiming for $\beta = 0.0015$ and $\gamma = 0.1429$.

Example 4 (ABC for Abakaliki data)**.** We next wish to perform inference on a Smallpox outbreak in Abakaliki, Nigeria, as detailed in §3.1. We will make use of the temporal data of the outbreak, which contains information on the detection times, which we will take to be the recovery times following Kypraios et al. (2016) [19] and O'Neill et al. (1999) [25], of all 30 infected individuals from the population of 120. The detection (recovery) times are detailed in §3.1.

Since we are now utilising different information, it makes sense to choose new summary statistics that utilise this new information. To require all the recovery events to occur at around the same time as in the data would be very restrictive on the number simulations we could accept. What may be a better metric, as far as the ABCs acceptance rate is concerned, is whether or not the simulated outbreak is progressing at a similar rate as the data. For this reason, following Kypraios et al. (2016) [19], we take the summary statistics, $S(\cdot)$, to be the number of recovery events in a series of time periods, and the time when the final infected individual recovers, $T_{(I=0)}$. We take the time intervals to be $[0, 13], (13, 26], (26, 39], (39, 52], (52, 65], (65, 78], (78, \infty]$. Thus, the summary statistics for the Abakaliki dataset are given by:

$$S(\mathbf{X}_{\text{Abakaliki}}) = (2, 6, 3, 7, 8, 4, 0, 76).$$

For the distance function, we wish to combine the 8 summary statistics into one number, weighted on their importance and their scale. We note that the final summary statistic, $T_{I=0}$, is anywhere between 10 and 50 times larger than all the other summary statistics, so we may wish to scale this down. Again following Kypraios et al. (2016) [19], we choose the distance function to be:

$$d(\mathbf{X}_{\text{Abakaliki}}, \chi) = \left[\sum_{i=1}^{7} (b_i - b_i^{\chi})^2 + \left(\frac{T_{(I=0)} - T^{\chi}_{(I=0)}}{50} \right)^2 \right]^{\frac{1}{2}},$$

where b_i is the observed number of recoveries in time interval i, and $T_{(I=0)}$ is the observed epidemic duration, for the Abakaliki data. The superscript, χ, denotes similar notions for the simulated data.

Given that we only have temporal information on when the first recovery occured, we do not know how many infected individuals there were at the beginning of the epidemic. With this in mind, we wish to edit the model to reflect our uncertainty in this area. To do this, we begin the epidemic with one infected individual and run the process until the first recovery. We then discard everything that happened before the first recovery, and take the current states as the starting conditions of the epidemic, before letting it continue. For instance, we start with 1 infected individual, we may then have 5 infected individuals by time step 10, say, when the first recovery occurs. So we discard time steps 0 to 9, make time step 10 the new first time step, time step 1, and the epidemic we are interested in now starts with 1 recovered individual and 4 infected.

Again, we do a pilot run of one million simulations in order to choose an appropriate tolerance, ϵ, and again we choose $Exp(1)$ priors for both β and γ. The results of the pilot run are presented in Table 5.

Tolerance	5	5.428	6	8	10	12
Accepted samples	7	115	8069	8697	11943	245969
Accepted rate	0.0007%	0.0115%	0.8069%	0.8697%	1.1943%	24.5969%
$\mathbb{E}[\beta\|\mathbf{X}]$	0.0011	0.0021	0.0069	0.0069	0.2033	0.9118
$\mathbb{E}[\gamma\|\mathbf{X}]$	0.0995	0.3718	1.3244	1.3329	1.0782	0.2738

TABLE 5. A summary of the pilot run for the ABC for the Abakaliki data.

While we may not know the true values of the parameters in this case, we can see from the pilot run a pattern in the estimates, similar to that which we saw in the Example 3. The estimates for both β and γ mostly get larger and further from zero as the tolerance gets larger.

This time we will not alter our prior distributions since the estimates of the posterior mean are much more varied across the tolerances, but we will use tolerance 5.428. We choose this tolerance since the estimates at the next tolerance are more than tripled, which given we have 115 samples seems a bit extreme. We ran the ABC algorithm to obtain 1000 samples from the approximate posterior distribution for β and γ, the results of which are presented in Table 6.

$\mathbb{E}[\beta\|\mathbf{X}]$	$\text{sd}(\beta\|\mathbf{X})$	$\mathbb{E}[\gamma\|\mathbf{X}]$	$\text{sd}(\gamma\|\mathbf{X})$	Simulations	Acceptance rate	Time
0.002	0.002	0.368	0.33	7,685,538	0.013%	5408 seconds

TABLE 6. A summary of the ABC results for the Abakaliki data.

The ABC algorithm took 7,685,538 simulations to find the 1000 accepted samples, taking 5408 seconds (90 minutes). The results have stayed fairly consistent with the estimates from the pilot run at this tolerance, and the acceptance rate has also stayed approximately the same. We notice that γ has a very large variance, suggesting we are rather uncertain about our estimate. It may be that we need a smaller tolerance in order to be more confident in our estimate of γ.

We are fortunate in this case that the Abakaliki dataset is well known, and has been analysed many times before. We compare our results to those of Bailey & Thomas (1971)[2] who utilised likelihood based methods to estimate the parameters of their model. Their mean (standard deviation) estimates were $\beta = 0.00168(0.00047)$ and $\gamma = 0.162(0.050)$. Since they used a different method and most probably assumed a slightly different model with slightly different characteristics, we would not expect our results to match exactly, however, the fact that our estimate for β is close is encouraging. Our estimate for γ is a bit large, however, the posterior distribution for γ is quite skewed, with the median being 0.2626, which is much closer. It may be that our tolerance is too large as we also notice that, due to our uncertainly in γ, the Bailey & Thomas estimate is well within one standard deviation of our estimate.

□

Example 5 (ABC for Gastroenteritis data)**.** Finally, we wish to perform inference on a Gastroenteritis outbreak in South Carolina, that was belived to have been spread by human contact, as detailed in §3.2. Similiarly to the Abakaliki dataset, the Gastroeneritis dataset contains temporal data on the outbreak, which we can utilise in our analysis. The temporal data in this case is the onset of symptoms for all infected individuals, but we will agian take this to be the recovery times, following Kypraios et al. (2016)[19] and O'Neill et al. (1999)[25]. The detection (recovery) times are detailed in §3.2.

Following the Abakaliki example, we take the summary statistics, $S(\cdot)$, to be the number of recovery events in a series of time periods, and the time when the final infected individual recovers, $T_{(I=0)}$. Following Kypraios et al. (2016)[19], we take the time intervals to be $[0,1],(1,2],(2,3],(3,4],(4,5],(5,6],(6,7],(7,\infty]$. Thus, the summary statistics for the Gastroenteritis dataset are given by:

$$S(\mathbf{X}_{\text{Gastro}}) = (1,4,2,3,3,10,5,0,7).$$

For the distance function also, we follow the example of the Abakaliki dataset. We see this time, however, that the duration of the outbreak is not an order of magnitude larger than the other summary statistics, however we will use the same distance function for the sake of convenience and so that we can compare our results to Kypraios et al. (2016)[19] who did the same. We choose the distance function to be:

$$d(\mathbf{X}_{\text{Gastro}},\chi) = \left[\sum_{i=1}^{8}(b_i - b_i^{\chi})^2 + \left(\frac{T_{(I=0)} - T^{\chi}_{(I=0)}}{50}\right)^2\right]^{\frac{1}{2}},$$

where b_i is the observed number of recoveries in time interval i, and $T_{(I=0)}$ is the observed epidemic duration, for the Gastroenteritis data. The superscript, χ, denotes similar notions for the simulated data.

For exactly the same reasons we edit our model to start with a randomly assigned number of infected individuals, and choose $Exp(1)$ priors for both β and γ. The results of our pilot run are presented in Table 7.

The pilot run took 347 seconds to simulate its one million outbreaks. Again we do not know the true values of the parameters, but again we see the patterns with tolerance evidenced. The estimates of the posterior means seem to get smaller as the tolerance

Tolerance	4	4.5	5	6	8
Accepted samples	27	255	900	11748	17829
Acceptance rate	0.0027%	0.0255%	0.0900%	1.1748%	1.7829%
$\mathbb{E}[\beta\|\mathbf{X}]$	0.0117	0.0107	0.0093	0.0095	0.0272
$\mathbb{E}[\gamma\|\mathbf{X}]$	1.8105	1.8353	1.6307	1.3913	1.4199

TABLE 7. A summary of the pilot run for the ABC for the Gastroenteritis data.

increases, however we note that there is little change between the estimates for tolerances 4 and 4.5, suggesting that we can use the larger tolerance 4.5 to increase the acceptance rate without losing accuracy. It is also worth noting that the acceptance rate does not increase linearly with the tolerance, as in all the other examples, so there is no reason to believe that the accuracy will increase linearly as the tolerance decreases, either.

Since the acceptance rates are reasonable at relatively low tolerances, we will keep our priors the same, and will use a tolerance of 4.5. Our ABC algorithm took 1518 seconds (25 minutes) to find 1000 accepted samples from the approximate posterior distribution for β and γ from $4,333,099$ simulations. The results are presented in Table 8. The results have again remained fairly consistent with the pilot run, including the acceptance rate. We notice that the estimate for γ is slightly higher than in the pilot however. It is also worth noting that the median for γ was 1.6797, so the posterior distribution is quite skewed.

$\mathbb{E}[\beta\|\mathbf{X}]$	$\text{sd}(\beta\|\mathbf{X})$	$\mathbb{E}[\gamma\|\mathbf{X}]$	$\text{sd}(\gamma\|\mathbf{X})$	Simulations	Acceptance rate	Time
0.0107	0.0035	1.9506	1.1577	4,333,099	0.023%	1518 seconds

TABLE 8. A summary of the ABC results for the Abakaliki data.

This dataset has been analysed before, but previous analyses have used a range of methods quite different to our own. For this reason we would not expect our inference to match exactly, and it may be that we cannot compare our estimates directly as our parameters are not equivalent. Kypraios et al. (2016) [19] used a different model such that their infection rate β^* was equivalent to $N \times \beta$ in our notation. Converting their estimates to our notation they had posterior mean estimates of $\beta = 0.0153$ and $\gamma = 1.14$. Britton & O'Neill (2002) [7] on the other hand used a model which included a random social structure (as opposed to one with a homogeneously mixing population), and obtained estimates of $\beta = 0.061$ and $\gamma = 1.47$. We can see that in general these estimates from all three models are similar, and the models are quite different so we would expect reasonable differences, but the similarities are still reassuring. We can also compare the results based on the value of R_0 that each model estimates. Kypraios et al. (2016) [19] estimates their own R_0 value to be 1.14, and Britton & O'Neill's to have a mean of 1.17 and a median of 1.14. We calculated our own R_0 value to have a mean of 1.67 and a median of 1.03, using the equation provide in §2.2. As we can see there is some general agreement between the different analyses, based on this summary statistic, which lends credence to our results. However, our inference may be able to be improved by re-running the algorithm with a stricter tolerance.

□

7. Coupled Approximate Bayesian Computation

The ABC algorithm, while performing reasonably well in terms of estimating the parameter values of the simple models we have presented it, does have an issue with its

acceptance rate. We can improve on the acceptance rate of the ABC by utilising a different algorithm known as Coupled Approximate Bayesian Computation. In Coupled ABC, we can seperate the parameters from the simualtion, such that we can simulate one outbreak, and choose the parameter values that make it best match the data. In this case, we do this by utilising a different model for simulating the outbreaks, known as the Sellke construction[30]. Unfortunately, this model does not concern any temporal elements of the data, so can only be used in cases where the final size is the only data we have/use. Since it is not uncommon for the only data available to be final size data, this coupled ABC algorithm is still useful in practice. It is possible to use the Coupled ABC and utilise temporal information, but it is a different process than what is presented here. We begin by explaining how this new model for simulating epidemics works, before explaining how the coupled ABC utilises its benefits. We then present an algorithm for coupled ABC, and apply the coupled ABC algorithm to the case study datasets.

7.1. **Sellke construction.** The S-I-R construction of the epidemic process may be one of the most intuitive, but it is far from the only option. In cases where we are not interested in the temporal development of an epidemic, but merely the final size, we can make use of the Sellke construction[30]. The Sellke construction is mathematically equivalent to the Gillespie algorithm, it just goes about things in a different way[16].

The basic concept behind the Sellke construction is that every individual has a personal threshold, T_i, $i \in (1, \ldots, N)$. If an individual becomes infected they have a personal infectious period, I_i. During this period they are infected/infectious, and after this period they are removed. While an individual is infectious, they put out infectious pressure λ per unit time step, which is split between all other individuals. If a susceptible individuals total amassed infectious pressure goes above their personal threshold, T_i, then they become infected for a period of length I_i. When it comes to simulating an epidemic in this way, there are lots of simplifications that can make the process more computational efficient.

We begin with an initial total population size of N individuals, with m of those being infectious, leading to a susceptible population size of $N - m$ individuals. We often take $m = 1$.

We first draw independent infectious periods for every individual from the infectious period distribution $f(I)$, which in this case we take to be $I \sim$ Exponential(1). In total we will draw N infectious periods; m for the initial infected individuals, and then an additional $N - m$ for the susceptible individuals for if they become infective. These will be indexed by $I_1, \ldots, I_m, I_{m+1}, \ldots, I_N$.

We next draw the thresholds of the susceptible individuals independently from the threshold distribution, which we again take to be $T \sim$ Exponential(1). We will draw thresholds for all N individuals in the population, $T_1, \ldots, T_N$, but since the first m are already infected, we will set their thresholds to zero, $T_1 = \cdots = T_m = 0$. These thresholds are unordered. We move forward with the assumption that $m = 1$, so only $T_1 = 0$.

Let λ be the infection rate. While an individual is infectious (i.e. they have been infected for less time than their infectious period) they contribute λ infectious pressure per unit time-step, which is split evenly between all the other individuals in the population. So for each unit time-step when individual i is infectious, they contribute λ/N infectious pressure to each other individual in the population. Arguably, this should be $\lambda/(N-1)$, however, in sufficiently large populations this makes little difference, and using N makes the equations neater. For the infected or recovered individuals, (individuals are recovered when they have been infected longer than their infectious period), the infectious pressure does not matter, but for the susceptible individuals their 'pool' of infectious pressure

increases, moving them closer to becoming infected. Individual i is then infected if their total 'pool' of infectious pressure exceeds their personal threshold T_i.

Let us now consider the population ordered ascendingly by their thresholds, we have $T_{(1)} \leq T_{(2)} \leq \cdots \leq T_{(N)}$. So we can view the above as an individual becomes infected if their threshold is less than the total sum of the infectious periods that have come before multiplied by λ/N. That is, individual (i) becomes infected if

$$T_{(i)} \leq \frac{\lambda}{N} \sum_{j=1}^{(i-1)} I_j.$$

As a simple example, six sequential infectious individuals with infectious periods of one unit time-step, $I_1 = \cdots = I_6 = 1$, is equivalent to one infectious individual with an infectious period of six unit time-steps, $I_1' = 6$. The next individual, in the sequence of ascendingly ordered thresholds, will become infected if their threshold $T_{(i+1)} \leq \frac{\lambda}{N} \sum_{j=1}^{6} I_j = \frac{\lambda}{N} I_1' = \frac{\lambda}{N} 6$. Since we are not interested in any temporal aspects of the epidemic, it does not matter whether all the infectious periods happen at overlapping times or sequentially.

Let $\tilde{L}_i$ be the additional infectious pressure necessary to infect individual $(i+1)$ given that individual (i) is infected, that is $\tilde{L}_i = T_{(i+1)} - T_{(i)}$. Then $\tilde{L}_i \sim Exp(N-i)$. Thus, in the case when $m = 1$, a second person is infected if

$$\tilde{L}_1 = T_{(2)} - T_{(1)} = T_{(2)} \leq \frac{\lambda}{N} I_1,$$

and a third person is then infected if

$$\tilde{L}_1 + \tilde{L}_2 = (T_{(2)} - T_{(1)}) + (T_{(3)} - T_{(2)}) = T_{(3)} \leq \frac{\lambda}{N}(I_1 + I_2),$$

and so on. If we times both sides by N we can let $L_i = N\tilde{L}_i$ so that now $L_i \sim Exponential(\frac{N-i}{N})$ and a second individual becomes infected if $L_1 \leq \lambda I_1$ and so on.

Thus, we can calculate that there are M new people infected if for all $k \in (1, 2, \ldots, M)$,

$$\sum_{i=1}^{k} L_i \leq \lambda \sum_{i=1}^{k} I_i. \tag{1}$$

Alternatively, we could think of this as

$$\min \left\{ m : \; T_{(m+1)} = \sum_{i=1}^{m} L_i > \lambda \sum_{i=1}^{m} I_i \right\} \tag{2}$$

is the total number of infected individuals in the epidemic. Thus the simplest way of simulating an epidemic under the Sellke construction is given in Algorithm 5.

Sellke Construction Epidemic:

Inputs: Population size, N; Infection parameter, λ.

1. Initialise the epidemic with $m = 1$ infectious individuals and $N - m$ susceptible individuals.
2. Draw $I_i \sim Exponential(1)$ and $L_i \sim Exponential(\frac{N-i}{N})$ for $i \in (1, ...N)$.
3. Calculate the final size of the epidemic using Eq.1 or Eq.2 for a given λ.

ALGORITHM 5. The algorithm presented here, and Equations 1, 2, and 3, are adapted from Kypraios et al. (2016)[19].

We can see that using Eq. 2 we can calculate the final size of the epidemic for any value of λ. If we are interested in a particular final size, we can find all the values of λ that produce that final size for a given simulation. That is for

$$\lambda \in \left[\max_{1 \le k \le M-1} \left\{ \frac{\sum_{i=1}^{k} L_i}{\sum_{i=1}^{k} I_i} \right\}, \frac{\sum_{i=1}^{M} L_i}{\sum_{i=1}^{M} I_i} \right), \tag{3}$$

M total people are infected. It may be that the upper limit of the interval is smaller than the lower limit; this represents that when M individuals are infected, at least $M+1$ individuals will be infected.

7.2. **Coupled ABC: The algorithm.** The Sellke construction benefits from the idea of coupling. We can separate the parameter, λ, from the simulation. Thus we only have to simulate once and can test different λ values to see what final size they result in (as opposed to having to run a new simulation for every λ value). In addition to this, we can even use analytical results (Eq.3) to 'bin' λ and show the final size for each interval of λ. For instance it may be that for all values of $\lambda \in (2, 4)$ we have a final size of 10, and that if we go even slightly above 4 we have a final size of 15. Figure 2 demonstrates the final size of a simulated outbreak for each interval of λ.

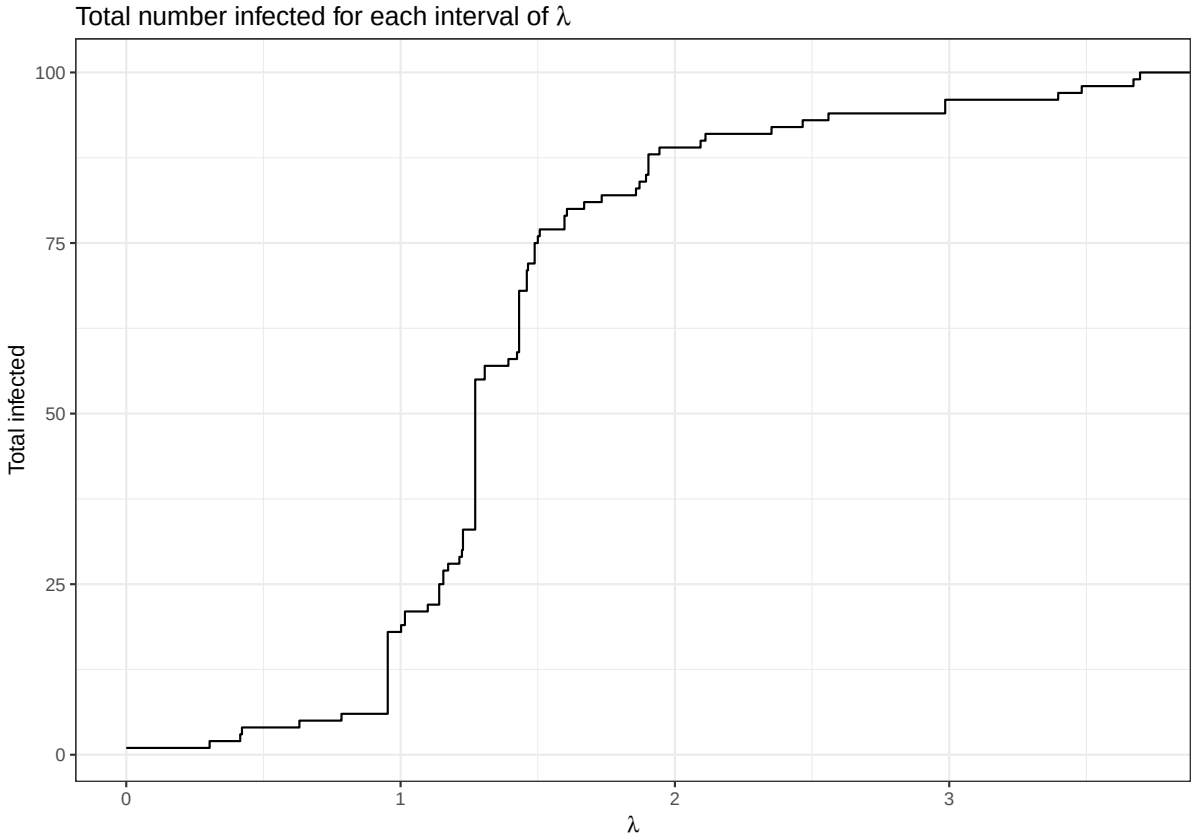


FIGURE 2. The final size of a simulated epidemic for each interval of λ.

As far as the ABC is concerned, this is extremely useful, as instead of drawing the parameter from a prior distribution, we can simply simulate an outbreak and choose the value (or range) of the parameter that best matches our data, thus significantly reducing the computational burden and increasing the acceptance rate. We call this algorithm the coupled ABC (cABC). Technically, the cABC algorithm presented here is the 2-stage

coupled ABC algorithm of Neal (2012)[23], however we will refer to it as the coupled ABC algorithm. Stage 1 of algorithm is presented in Algorithm 6.

Coupled ABC (cABC) - Stage 1:

Inputs: Data, $\mathbf{X}$; Model, $\mathsf{M}(\lambda)$; Summary statistics, $S(\cdot)$; Distance function, $d(\cdot)$; Kernel function, $K_\epsilon(\cdot)$.

1. Simulate an outbreak with $\lambda = 1$ using Algorithm 5.
2. Calculate the intervals for λ for which M individuals are infected using Eq.3, with $M \in (1, \ldots, N)$. Note that there may be less than N intervals, when M individuals infected implies infecting atleast $M + 1$.
3. Record the intervals of λ, A_i, which give final sizes for which $d(S(X), S(\chi_i)) \leq \epsilon$, as well as their weighting, k_i calculated using the kernel $K_\epsilon(d(S(X), S(\chi_i)))$ (details below). Call B the union of all the accepted intervals A_i.
4. Repeat steps 1-3 until there have been T accepted simulations (the set B is non-empty).

ALGORITHM 6. The algorithm presented here is adapted from Neal (2012)[23].

To use this algorithm we will require a summary statistic $S(\cdot)$, a distance function $d(\cdot)$, a tolerance ϵ, and we introduce a kernel function $K_\epsilon(\cdot)$, which is used to weight each simulated outbreak/interval for λ. A kernel function can be used in any ABC algorithm to quantify how much we penalise divergence from the data, though we have omitted to do so thus far. In this case the obvious choice for the summary statistic is the final size of the epidemic, as the Sellke construction does not consider any temporal elements of the outbreak, and it allows us to take advantage of the coupled property. For the distance function, unless there are any special requirements of the data, we will normally just use the $L^2 - norm$, with tolerance $\epsilon \geq 0$ chosen appropriately for the data. There are two main choices for the kernel, $K_\epsilon(\cdot)$. The first being a simple step function which gives 1 if $d(S(X), S(\chi_i)) \leq \epsilon$, and 0 otherwise. The other is the discrete Epanechnikov kernel given by

$$K_\epsilon(d(S(X), S(\chi_i))) \propto \begin{cases} 1 - \left|\frac{S(X) - S(\chi_i)}{\epsilon}\right|, & \text{if } d(S(X), S(\chi_i)) \leq \epsilon \ , \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Stage 1 provides us with independent and identically distributed sets $B^1, \ldots, B^T$ [23]. If the prior on λ, $\pi(\lambda)$, is a uniform distribution, and we use a step kernel function, K_ϵ, then the sets are independent intervals drawn from $\pi(\lambda|X, \epsilon)$ [23]. Stage 2 is used to find draws from the approximate posterior distribution for λ given a specific choice of prior and kernel. Let $L_\pi = \max_{1 \leq t \leq T} \{\sup_{\lambda \in B^t} |\pi(\lambda)|\}$. Stage 2 of the algorithm is then given in Algorithm 7.

For clarity, χ_i^t is the simulation generated by the interval, A_i^t, in B^t which contained the sampled λ value, and $K_\epsilon(d(S(X), S(\chi_i^t)))$ is the weighing, k_i^t, attached to that, which was calculated and recorded in Stage 1.

7.3. **Coupled ABC: Examples.** To see the effectiveness of the coupled ABC we again begin with an example of simulated data, before moving on to applying the methodology to the Abakaliki dataset. This time, however, we cannot take into account any temporal aspects of the data, so we will be purely looking at the final size data.

Coupled ABC (cABC) - Stage 2:

Inputs: Data, $\mathbf{X}$; Parameter sets, B; Prior on λ, $\pi(\lambda)$; Interval weightings, k_i; and L_π.

5. Sample Q from $Q \in (1, \ldots, T)$ with $\mathbb{P}[Q = t] = \frac{|B^t|}{\sum_{k=1}^{T} |B^t|}$.
6. Sample λ from B^Q uniformly.
7. Accept λ as a draw from the approximate posterior distribution for λ with probability $\frac{1}{L_\pi} K_\epsilon(d(S(X), S(\chi_i^t)))\pi(\lambda)$. Otherwise, reject λ.
8. Repeat until there has been U accepted samples.

ALGORITHM 7. The algorithm presented here is adapted from Neal (2012)[23].

Example 6 (Coupled ABC for simulated data)**.** Let the population consist of $N = 200$ individuals, and we will begin with $m = 1$ infected. We generated an outbreak using the Sellke construction algorithm (Alg. 5) with the parameter value $\lambda = 2$. This resulted in 167 infected individuals.

As suggested in §7.2, we will use the final size as our summary statistic, $S(\cdot)$, and the $L^2 - norm$ as our distance function. For the coupled ABC, however, we have an additional choice to make. We require a kernel function which is used to weight the different simulations, and thus the different intervals of λ, A_i, based on the distance of their summary statistics from those of the true data. In this instance we will use a simple step function, giving a set of λ values, A_i, the weighting of 1 if $d(S(X), S(\chi_i)) \leq \epsilon$, and 0 otherwise. Finally we choose our prior on λ to be an $Exp(1)$.

Since the coupled ABC allows us to greatly increase our acceptance rate, and due to the nature of the algorithm, it is no longer necessary/possible for us to have a pilot run. Thus, we jump into applying the algorithm with a tolerance that we believe would be acceptable, $\epsilon = 2$. Since the summary statistic, $S(\cdot)$, is interpretable in this case, this tolerance means, in stage 1, we will accept any intervals of λ that result in a final size between 165 and 169, and in this case, they will all be equally weighted because of our step function kernel. The coupled ABC took around 79 seconds in total to generate these $10,000$ samples from the approximate posterior distribution for λ. We present the results of the run in Table 9 below.

$\mathbb{E}[\lambda\|\mathbf{X}]$	$\text{sd}(\lambda\|\mathbf{X})$	Sims: Stage 1	Samples: Stage 2	Acceptance rates	Time
2.165	0.254	19,038	25,600	53% and 39%	79 seconds

TABLE 9. A summary of the cABC results for the simulated data. We are aiming for $\lambda = 2$. The acceptance rates are for stage 1 and stage 2, respectively.

The first thing we notice is that the acceptance rate of the algorithm overall has dramatically increased, thus drastically reducing our computation time. Our estimate of the approximate posterior mean is slightly higher than the true value, but well within one standard deviation. Given that the acceptance rate was so high it may even be justifiable to make our tolerance even smaller to increase our accuracy. Also bear in mind that we are working with very little information, only the final size data, and this limits what inference is possible, even if the data was generated from the model we are using.

□

Example 7 (Coupled ABC for Abakaliki data)**.** The Abakaliki data consists of $N = 120$, 30 of which become infected with smallpox by the end of the epidemic.

Again, we will use the final size as our summary statistic, $S(\cdot)$, and the $L^2 - norm$ as our distance function. For the kernel function in this instance we will use the discrete Epanechnikov kernel detailed in §7.2. This will give larger weights to sets of λ values which result in simulations with final sizes closer to 30. Finally we again choose our prior on λ to be an $Exp(1)$.

A pilot run would again be inappropriate for this algorithm, so we go straight to applying the algorithm with a tolerance that we believe would be acceptable, again $\epsilon = 2$. In this case, in stage 1, we will accept any set of λ values, A_i, which result in an outbreak with final size between 28 and 32, with greater weight given to sets closer to 30. The coupled ABC took around 71 seconds in total to generate these $10,000$ samples from the approximate posterior distribution for λ. We present the results of the run in Table 10 below.

$\mathbb{E}[\lambda\|\mathbf{X}]$	$\text{sd}(\lambda\|\mathbf{X})$	Sims: Stage 1	Samples: Stage 2	Acceptance rates	Time
1.1621	0.30	76,513	73,794	13% and 14%	199 seconds

TABLE 10. A summary of the cABC results for the simulated data.

The acceptance rates are much lower than for the simulated dataset, but this may be expected, since we know that the model we chose for the coupled ABC in that case was the one we used to generate the simulated dataset that we are trying to analyse. It could also be down to our choice of prior for Stage 2.

Since we are using a model construction which is very different to the chain-binomial, it is more difficult to compare our results to the previous methods. However, λ in this case can be interpreted as an estimate of R_0, so we can calculate R_0 for our ABC estimates in Example 4 and compare them that way. Our posterior mean and median estimates for R_0 using the ABC were 1.0880 and 0.8712 respectively. These are quite a bit lower than the estimate from our cABC (but still within 1 standard deviation), and the median suggests an R_0 value less than 1, which may be possible but is unlikely. On the other hand, we calculate $R_0 = 1.2444$ for Bailey & Thomas (1971) [2], which is well within one standard deviation of our estimate, and seems more likely since $R_0 > 1$. We can also compare our results to Kypraios et al. (2016) [19], who used a coupled ABC on the same dataset using a larger tolerance, $\epsilon = 10$, and returned an estimate of the posterior mean (standard deviation) of λ to be 1.16 (0.29), which exactly agrees with our estimates. They also ran an ABC and sequential ABC algorithm (§9) to make direct inference on λ, which both obtained a posterior mean of 1.16 also.

□

8. Semi-Coupled Approximate Bayesian Computation

Coupled ABC works well when we have one parameter which has some form of natural ordering, for instance, the larger the parameter, the greater the final size of the epidemic. Sometimes though, the questions presented by the data require the use of additional parameters, even when we do not consider a temporal characteristic of the data. It may not be possible to disentangle all of these parameters from the simulations. In these cases it often works well to use a mixed approach. We can still take advantage of the coupled nature of some of the parameters, while drawing the non-coupled parameters from their prior distribution in the traditional way. This is known as semi-coupled ABC. It works in much the same way as the coupled ABC, except that the simulation is generated using random draws from the prior distribution of the non-coupled parameters, and then the coupled parameter values are chosen to fit the data best.

In this case, we will begin by going over the data used in this instance, and then explain how we can build a semi-coupled ABC algorithm for this specific case, before applying it.

8.1. **Setting up the Vaccination data.** Recall from §3.3 the outbreak of Measles in a school in Honkajoki, Finland. There were a total $N = 417$ individuals in the population, which were split into 3 sets based on their vaccination history. A student of type $k \in (0, 1, 2)$ had received k doses of Measles vaccination prior to the outbreak. The data is final size data, with a total 35 students infected, the details of which are given in Table 1. The model presented in this section is adapted from Neal (2018) [22].

We wish to infer the infection parameter, λ, as well as an individual of type k's probability of being infected if contacted by any infectious individual, $q_0, q_1, q_2 \in [0, 1]$. Since we only have final size data, there is not enough information to infer how the vaccine affects both the infectivity and susceptibility of an individual, so following Neal (2018) [22] we will assume the vaccine has no affect on infectivity. Thus q_k can be thought of as the protective effect of having had k measles vaccinations. We will also assume, as Neal did, that being unvaccinated conferred no protective benefit, so we set $q_0 = 1$.

We will use a Sellke construction (§7.1) for the basis of the model, but also introduce new elements which take into account the protective effect of the vaccines, q_0, q_1, q_2. We know that 35 individuals were infected, and we also know how many of each type of student were infected, and we would ideally want our simulations to match that. To do this, we select values for q_1 and q_2, and generate a random infection order for all the individuals in the population, based on q_0, q_1, and q_2. We then look at the first 35 infected individuals, as they are all we are interested in, and if the number of each type of student matches that of the data, then we accept the infection order, otherwise we discard it and generate a new one until we have one that matches.

Now that we have our infection order, for $j \in (0, 1, 2)$, let $s_{j,i}$ denote the total number of susceptible individuals of type j after the i^{th} infection. We use the convention that $s_{j,0} = n_j$, the total number of individuals of each type. Note that $\mathbf{s}$ depends on the infection order, which we will call ω.

For $i \in 1, 2, \ldots,$ let $\alpha_i = \frac{1}{N} \sum_{j=0}^{2} s_{j,i} q_j$. This is the probability that, following the i^{th} infection, an infectious contact will result in an infection [22]. This can also be thought of as the probability that the contact is with a susceptible individual and succeeds in infecting them. We then generate $L_i \sim Exp(\alpha_i)$ for each individual, where L_i is the additional infectious pressure needed after the i^{th} infection for the $(i+1)^{\text{th}}$ infection to take place [22].

We can now use the identities outlined in §7.1, namely Eq. 3, to calculate the final size of the epidemic for each interval of λ. The algorithm then proceeds in much the same way as the 2-stage coupled ABC algorithm. We lay out the algorithm for this case in the next section.

8.2. **Semi-coupled ABC: The algorithm.** Stage 1 of the algorithm is presented in Algorithm 8.

Stage 2 of the algorithm is as in §7.2, but we will reprint it here for ease as Algorithm 9. Let $L_\pi = \max_{1 \le t \le T} \{\sup_{\lambda \in B^t} |\pi(\lambda)|\}$.

Again for clarity, χ_i^t is the simulation generated by the interval, A_i^t, in B^t which contained the sampled λ value, and $K_\epsilon(d(S(X), S(\chi_i^t)))$ is the weighing, k_i^t, attached to that, which was calculated and recorded in Stage 1.

8.3. **Semi-coupled ABC: Example.** We will now put the algorithm laid out above into action for the Measles vaccination dataset. As stated we will use a modified Sellke

Semi-coupled ABC (scABC) - Stage 1:

Inputs: Data, $\mathbf{X}$; Model, $\mathsf{M}(\theta)$; Prior on q_k, $\pi(q_1, q_2)$; Summary statistics for infection order, $S_\omega(\cdot)$; Summary statistics for epidemic, $S(\cdot)$; Distance function, $d(\cdot)$; Kernel function, $K_\epsilon(\cdot)$; Tolerance for infection order, ϵ_ω; Tolerance for epidemic, ϵ.

1. Set $\lambda = 1$ and $q_0 = 1$. Draw realisations of q_1 and q_2 from their prior, $\pi(q_1, q_2)$.
2. Using (q_0, q_1, q_2), generate an order of infection for all the susceptible individuals, ω^*.
3. Calculate the summary statistic for the order of infection, $S_\omega(\omega^*)$, the number of each type of individual in the first 35 infections. Repeat steps 1-3 until $d(S_\omega(X), S_\omega(\omega^*)) \leq \epsilon_\omega$.
4. For $j \in (0, 1, 2)$ and $i \in (1, \ldots, N)$, let $s_{j,i}$ denote the total number of susceptible individuals of type j after the i^{th} infection with $s_{j,0} = n_j$.
5. Calculate $\alpha_i = \frac{1}{N} \sum_{j=0}^{2} s_{j,i} q_j$ for all $i \in (1, \ldots, N)$.
6. Generate $L_i \sim Exponential(\alpha_i)$ for all $i \in (1, \ldots, N)$.
7. Calculate the intervals for λ, using Eq.3, for which M individuals are infected, with $M \in (1, \ldots, N)$. Note that there may be less than N intervals, when M infected individuals implies infecting atleast $M + 1$.
8. Record the intervals of λ, A_i, which give final sizes for which $d(S(X), S(\chi_i)) \leq \epsilon$, as well as their weighting, k_i calculated using the kernel $K_\epsilon(d(S(X), S(\chi_i)))$ (details as in §7.2), and (q_1, q_2). Call B the union of all the accepted intervals A_i.
9. Repeat steps 1-8 until there have been T accepted simulations (the set B is non-empty).

ALGORITHM 8. The algorithm presented here is adapted from Neal (2012) [23].

Semi-coupled ABC (scABC) - Stage 2:

Inputs: Data, $\mathbf{X}$; Parameter sets, B; Prior on λ, $\pi(\lambda)$; Interval weightings, k_i; and L_π.

10. Sample Q from $Q \in (1, \ldots, T)$ with $\mathbb{P}[Q = t] = \frac{|B^t|}{\sum_{k=1}^{T} |B^t|}$.
11. Sample λ from B^Q uniformly.
12. Accept λ as a draw from the approximate posterior distribution for λ with probability $\frac{1}{L_\pi} K_\epsilon \left(d(S(X), S(\chi_i^t)) \right) \pi(\lambda)$, and record (λ, q_1, q_2). Otherwise, reject λ.
13. Repeat until there has been U accepted samples.

ALGORITHM 9. The algorithm presented here is adapted from Neal (2012) [23].

construction, and we will choose $Uniform[0, 1]$ priors for the q_k, this allows the possibility that $q_1 < q_2$, meaning that a second vaccination shot has a detrimental effect and increases an individuals probability of being infected. The summary statistics for infection order, $S_\omega(\cdot)$, is as stated in the algorithm, the number of infected individuals of each type in the first 35 infecteds. We are aiming for (18,11,6) for types (0,1,2) respectively. We set the tolerance for this, $\epsilon_\omega = 0$ since the infection orders are very easy to generate. The summary statistics for the epidemic, $S(\cdot)$, is again as stated, the final size of the epidemic, which is 35 in our data. We set the tolerance of this, $\epsilon = 2$, meaning we can choose λ values that infect a little less or a little more than 35 people. For the distance function for both we will just use the $L^2 - norm$, and for the kernel weighting function, $K_\epsilon(\cdot)$, we will use the step function. Finally in stage 2 we will use an $Exp(1)$ prior on λ.

The algorithm took 987 seconds to run (16.45 minutes), and returned $10,000$ samples from the approximate posterior for (λ, q_1, q_2). The results are presented in Table 11.

$\mathbb{E}[\lambda\|\mathbf{X}]$	$\text{sd}(\lambda\|\mathbf{X})$	$\mathbb{E}[q_1\|\mathbf{X}]$	$\text{sd}(q_1\|\mathbf{X})$	$\mathbb{E}[q_2\|\mathbf{X}]$	$\text{sd}(q_2\|\mathbf{X})$
2.4152	0.6976	0.3409	0.1300	0.2433	0.1143
Sims: Stage 1	Samples: Stage 2	S1: Acc. rate	S2: Acc. rate	S1: Time	S2: Time
7,298,884	55,526	0.01%	18.00%	976 seconds	11 seconds

TABLE 11. A summary of the Semi-coupled ABC results for the Measles vaccination data.

As we can see the acceptance rates are still much higher than we might expect to get using the basic ABC algorithm, especially with such a small tolerance. The 0.01% acceptance rate in stage 1 could be attributed to the fact that a simulation is counted as soon as an infection order is drawn, and the tolerance on matching those is zero, so a lot of them will be discarded. However, relative to a full simulation of the data, they take very few computational resources to generate. We can compare our results to Neal (2018) [22], who analysed this dataset using a range of ABC and alternative methods, such as MCMC. We will compare our results to their MCMC results, as MCMC in theory should return samples from the true posterior. They had a posterior mean (standard deviation) for each parameter of $\lambda = 2.780(0.691)$, $q_1 = 0.303(0.116)$, and $q_2 = 0.220(0.098)$. We see that our posterior mean estimate for λ is lower, though within one standard deviation, and our estimates for q_1 and q_2 are higher. We interpret this as our Semi-coupled ABC inferring that the disease was less infectious, but that the vaccines were also less effective at protecting against the disease. Neal's (2018) [22] basic ABC methods inferred similar results for q_1 and q_2, with posterior mean (standard deviation) estimates of $\lambda = 2.851(0.691)$, $q_1 = 0.321(0.116)$, and $q_2 = 0.240(0.098)$. Their λ result being much greater than our own could be due to their choice of prior. They used an $Exp(0.1)$ as Measles is estimated to have an R_0 value around 10 [13]. Rerunning our own Semi-coupled ABC algorithm with this same prior, we get estimates much closer to theirs, with posterior mean (standard deviation) estimates of $\lambda = 2.970(0.890)$, $q_1 = 0.291(0.122)$, and $q_2 = 0.210(0.101)$, and make the opposite inference, with λ now being higher than their MCMC estimates, and q_1 and q_2 being lower. This demonstrates the effect that our choice in prior can have, with the posterior mean of λ increasing by 23% just based on our choice of prior. Another benefit of using the Semi-coupled ABC is that our prior choice for λ only comes into effect in stage 2, so to get 10,000 samples using the alternative prior only took 2.45 seconds.

9. Sequential Approximate Bayesian Computation

We have thus far seen that Approximate Bayesian Computation can be extremely inefficient if we make "poor" choices when specifying the components of the algorithm. In particular, the prior distribution we choose for our parameters can greatly affect our acceptance rate, and our tolerance can greatly affect our accuracy. We have seen that using the Sellke construction in the Coupled ABC algorithm we can improve the acceptance rate and not rely so heavily on our choice of prior, however, the Sellke construction is not useful when our data contains temporal information that we wish to utilise in our analysis. In this case, one possible alternative is to update our prior, and tolerances, sequentially as we go through the process. This helps the posterior find its mode quicker, thus increasing the acceptance rate. This is known as Sequential Approximate Bayesian Computation. The particular algorithm we will demonstrate is known as Particle Monte

Carlo Approximate Bayesian Computation (PMC-ABC) [19], which is one of many possible sequential ABC algorithms possible. During this dissertation we will refer to it as Sequential-ABC for ease.

The Sequential-ABC algorithm essentially works by running a series of shorter ABC algorithms, at each step using a stricter tolerance, and a prior based on the accepted draws of the previous iteration. There are many ways to update the prior and the tolerance, but in this dissertation we will follow the guidance of Kypraios et al. (2016) [19] and Beaumont et al. (2009) [3]. In this section we will explain how to implement the Sequential-ABC algorithm, and then demonstrate its use on the Abakaliki dataset.

9.1. **Sequential-ABC: The algorithm.** The Sequential-ABC algorithm takes the same inputs as the basic ABC algorithm; A set of summary statistics, $S(\cdot)$, a distance function, $d(\cdot)$, and an initial prior, $\pi(\theta)$, but has a sequence of tolerances, $(\epsilon_1, \ldots, \epsilon_L)$, and updates the prior after each iteration. The algorithm is presented in Algorithm 10.

The accepted samples (particles) from the final iteration, L, will be our sample from the approximate posterior distribution of θ. In the weighting, $\phi(\theta^*|\theta, 2\Sigma)$ is the density of a Multivariate Normal distribution with mean θ and variance-covariance matrix 2Σ, evaluated at θ^*.

9.2. **Sequential-ABC: Example.**

Example 8 (Sequential-ABC for Abakaliki data)**.** We now apply the Sequential-ABC algorithm to the Smallpox outbreak in Abakaliki. When compared to our inference using the ABC, we should be able to get greater accuracy and a higher acceptance rate using the Sequential-ABC. To ensure the results are comparable we need to maintain the same model and summary statistics. We will also use the same distance function and prior distribution. As a reminder, for the summary statistics, $S(\cdot)$, we use the number of recovery events in a series of time periods, and the time when the final infected individual recovers, $T_{(I=0)}$. We take the time intervals to be $[0, 13], (13, 26], (26, 39], (39, 52], (52, 65], (65, 78], (78, \infty]$. For the distance we choose:

$$d(\mathbf{X}_{\text{Abakaliki}}, \chi) = \left[\sum_{i=1}^{7} (b_i - b_i^{\chi})^2 + \left(\frac{T_{(I=0)} - T_{(I=0)}^{\chi}}{50} \right)^2 \right]^{\frac{1}{2}},$$

where b_i is the observed number of recoveries in time interval i, and $T_{(I=0)}$ is the observed epidemic duration, for the Abakaliki data. The superscript, χ, denotes similar notions for the simulated data. The model will generate a random number of initial infected individuals, as described in §4, and we will use $Exp(1)$ priors on both β and γ.

We do not require a pilot run as we will be using a sequence of tolerances. We should also be able to have a smaller final tolerance than for the basic ABC, and we can use the pilot run from our ABC to inform how low that could be. We choose our set of tolerances to be $(12, 10, 8, 6, 5.428, 5, 4.5)$. We ran the sequential-ABC algorithm to obtain 1000 samples from the approximate posterior distribution for β and γ, the results of which are presented in Table 12.

$\mathbb{E}[\beta\|\mathbf{X}]$	$\text{sd}(\beta\|\mathbf{X})$	$\mathbb{E}[\gamma\|\mathbf{X}]$	$\text{sd}(\gamma\|\mathbf{X})$	Simulations	Acceptance rate	Time
0.0010	0.0004	0.1223	0.0483	2,658,935	0.038%	1713 seconds

TABLE 12. A summary of the ABC results for the Abakaliki data.

The ABC algorithm took 2,658,935 simulations to find the 1000 accepted samples, and was completed in 1713 seconds (28.55 minutes). Comparing this to the basic ABC

Sequential ABC (seqABC):

Inputs: Data, $\mathbf{X}$; Model, $\mathsf{M}(\theta)$; Prior on θ, $\pi(\theta)$; Summary statistics, $S(\cdot)$; Distance function, $d(\cdot)$; Tolerances, $(\epsilon_1, \ldots, \epsilon_L)$.

1. Let $l = 1$ and choose an initial prior, $\pi(\theta)$, for the parameters θ.

While $l = 1$;

2. Draw parameters $\theta^* \sim \pi(\theta)$.
3. Simulate an epidemic, χ, using parameters θ^*
4. Calculate the summary statistics for χ using $S(\chi)$.
5. If $d(S(X), S(\chi)) \leq \epsilon_1$ then accept the draw θ^*, otherwise reject.
6. If θ^* was accepted, set the weight of θ^*, ω, equal to 1, and record both θ^* and ω.
7. Repeat steps 2-6 until there have been N accepted samples.
8. Set $l = l + 1$.

While $l \in (2, \ldots, L)$;

2. Sample parameters θ' from the previous iterations accepted samples (particles), $\theta^{(l-1)}$. The probability of choosing any sample (particle), θ'_i, is proportional to its weight, $\omega_i^{(l-1)}$.
3. Set Σ equal to the empirical weighted variance matrix of the accepted samples (particles) from the previous iteration, $\theta^{(l-1)}$, with weights, $\omega^{(l-1)}$. Generate parameters $\theta^* \sim MVN(\theta', 2\Sigma)$. Repeat until $\pi(\theta^*) > 0$.
4. Simulate an epidemic, χ, using parameters θ^*
5. Calculate the summary statistics for χ using $S(\chi)$.
6. If $d(S(X), S(\chi)) \leq \epsilon_l$ then accept the draw θ^*, otherwise reject.
7. If θ^* was accepted, set the weight of θ^*,
$$\omega \propto \frac{\pi(\theta^*) \sum_{i=1}^{N} \omega_i^{(l-1)}}{\sum_{i=1}^{N} \omega_i^{(l-1)} \phi(\theta^* | \theta_i^{(l-1)}, 2\Sigma)},$$
and record both θ^* and ω.
8. Repeat steps 2-7 until there have been N accepted samples.
9. Set $l = l + 1$.

Algorithm 10. The algorithm presented here is adapted from Kypraios et al. (2016)[19] and Beaumont et al. (2009)[3]. See §9.1 for details on $\phi(\cdot)$.

algorithm in §4, we can see that the acceptance rate has increased dramatically from 0.013% to 0.038%, which is almost 3 times as large. The number of simulations required (and thus the computation time) was also vastly reduced, from 7,685,538 to 2,658,935 (5408 seconds to 1713 seconds), which is a reduction of 65% (68%). On top of this we also have samples from the approximate posterior with a smaller tolerance, dropping from 5.428 to 4.5, and these are directly comparable tolerances.

As for the estimates, we can first see that our standard deviations are much smaller, 0.0004 compared to 0.002 for β (a decrease of 80%), and 0.0483 compared to 0.33 for γ (a decrease of 85%), suggesting we are also much more confident in our inference. The posterior mean estimate for β is half of what the ABC gave (0.001 compared to 0.002) and the estimate for γ is a third of what the ABC produced (0.1223 compared to 0.368).

On top of this the approximate posterior for γ is far less skewed than it was from the ABC, with the median now being 0.11572, compared to 0.2626.

The results are also much closer to the Bailey & Thomas (1971)[2] estimates. Their mean (standard deviation) estimates were $\beta = 0.00168(0.00047)$ and $\gamma = 0.162(0.050)$. Again we would expect differences since they assumed a different model and used a likelihood based approach to make inference, but we are now within one standard deviation of their γ estimate, and our own standard deviations are much closer to theirs. One possible reason for our increased accuracy is our stricter tolerance, which was only really viable using the Sequential-ABC in this case.

□

10. Sugarcane Yellow Leaf Virus

We have successfully demonstrated that various ABC methods can be successfully applied to a series of simulated and simple epidemic datasets to provide approximate inference. As an extension we wish to investigate how effective our various ABC methods are at making inference on a more complex dataset/epidemic. In this section we give the details of an outbreak of Yellow Leaf for Sugarcane plants in Guadeloupe, and explain how we intend to model the temporo-spatial spread of this disease in a field.

10.1. **SCYLV: The data.** Sugarcane Yellow Leaf Virus (SCYLV) is spread by aphids, and is especially prevalent in the Caribbean Islands, particularly on Guadeloupe[11]. Yellow Leaf can cause severe yield losses, so understanding it is important for Sugarcane production, which has major cultural and economical implications in Guadeloupe[11]. The dataset comes from one of four distinct trials set up to investigate the spread of the disease performed by Daugrois et al. (2011)[11].

The study consists of 1742 Sugarcane plants arranged on a lattice. There are 17 rows, with inter-row spacing of 1.5m, and the plants within each row are 0.5m apart. Most rows contained 103 plants, with the last row containing 94. The data we have consists of a series of snapshots of the field at weeks 6, 11, 15, 19, 23, and 30, which contain information on the position of all the plants in the field, and their infection status. The trial began with an disease free population at week 0. When a plant is infected with Yellow Leaf, it does not die, and it cannot recover. The spatial progress of the disease is presented in a series of plots in Figure 3 and the growth of the infectious population is presented in Figure 4.

Before progressing to our own analysis, the results of Daugrois et al. (2011)[11] may be of interest. They found that aphids appeared on plants within the first week (they were not artificially introduced) and that aphids were present on all plants by week 22. For the spatial dispersion of SCYLV they found that it had two stages; the first was random, caused by winged aphids from outside the field in the early stages of plant growth, before the soil was covered by a leaf canopy. The second stage was non-random, with a significant neighbourhood effect for plants within 0.5m to 2m, caused by wingless aphids. They found that disease spread was strongly correlated with aphid dynamics, that the time of the first arrival was very important to the rate of spread, that cumulative rainfall in the first few weeks was negatively correlated with aphid dispersion, but that an aphids per plant metric did not necessarily explain variation in incidence.

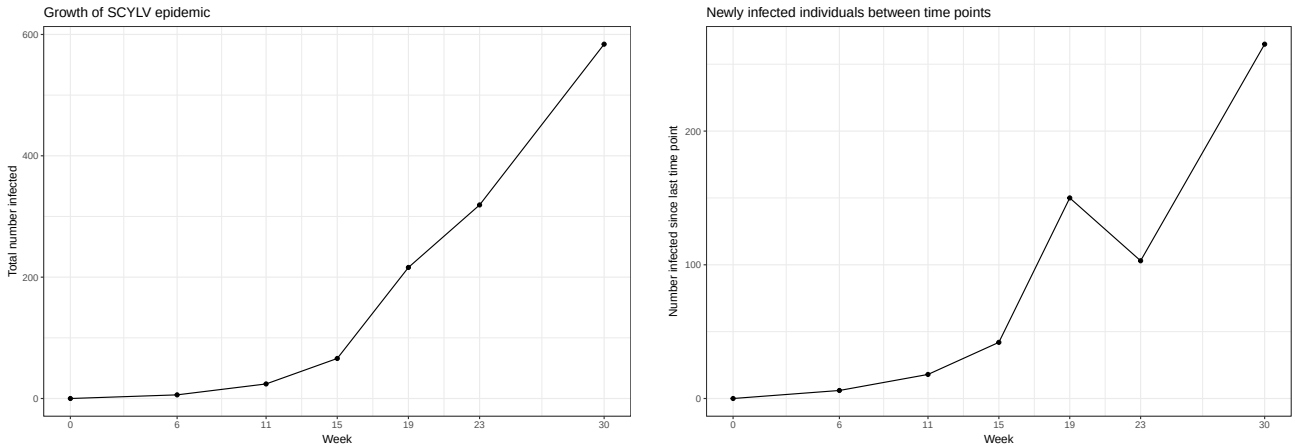
We now go on to explain how we intend to model the spread of the disease in this trial, using a model with both spatial and temporal aspects, before using the model in a series of ABC methods to make approximate inference in the next section.

10.2. **SCYLV: The model.** In contrast to all our other datasets thus far, we do not only know the final size and some information on the temporal spread of the disease, we

FIGURE 3. The spatial progress of the disease at (A) Week 6, (B) Week 11, (C) Week 15, (D) Week 19, (E) Week 23, (F) Week 30, in the SCYLV dataset.

also have information on the location of individuals (Sugarcanes) and the spatial spread of the disease. To improve our inference we will naturally want to include this in our model. The model presented in this section is adapted from Neal & Xiang (2017) [24].

To model the spatial spread of the disease we use a modified Gillespie algorithm, as detailed in §2.3. Firstly we note that there are only two states in this epidemic, S-susceptible and I-infected. There is not recovery/removal, so this is an SI epidemic. Given enough



(A) (B)

FIGURE 4. (A) The growth of the infectious population through time for the SCYLV dataset. (B) The number of newly infected individuals, n_i, at each time point t_i, since the last time point t_{i-1}.

time and no intervention, the whole population of sugar canes will become infected. At any given time in our simulations, we will know the position of every individual on the lattice, and who is infected. We make the assumption of a constant underlying rate of infection, λ, but also non-homogeneous contact between individuals. We assume that when an individual is infected, they put out infectious pressure on those around them. This infectious pressure has no affect on other infected individuals, but it increases the probability that a susceptible individual will become infected. If a susceptible individual becomes infected, they become immediately infectious, and put out infectious pressure of their own on those around them. We make the assumption that the force of infection, $F(\cdot)$, that an infected individual, x, puts on a susceptible, y, is a function of the Euclidean distance between them, $\sqrt{(x-y)^2}$. In our case we take $F_\alpha(x,y) = \frac{1}{\sqrt{2\pi\alpha^2}} \exp\left\{-\frac{(x-y)^2}{2\alpha^2}\right\}$. That is, we assume an isotropic Gaussian decay in the force of infection with increasing Euclidean distance, with the variance given by α^2. For a given pair of individuals, x and y, the distance between them remains constant at all times, in any simulation, so the force of infection between them is only depends on α in a simulation. Finally we assume an independent background infectious pressure of λr, with $r \in [0,1]$, which is how we believe the disease was introduced to the population initially.

Thus if we let the set of locations of infected individuals at time t be denoted by $\mathcal{I}$, then at time t, the infectious pressure that individual y is subject to is given by,

$$R_y = \lambda r + \sum_{x \in \mathcal{I}} \lambda F_\alpha(x,y), \tag{5}$$

$$= \lambda \left(r + \sum_{x \in \mathcal{I}} F_\alpha(x,y) \right). \tag{6}$$

We can also look at this as the rate at which individual y becomes infected at time t (as the rate is non-homogeneous between individuals and non-constant with time). Similarly, let $\mathcal{S}$ denote the set of locations of susceptible individuals at time t. Thus, at time t, the overall rate of infection is given by,

$$A = \sum_{y \in \mathcal{S}} \left\{ \lambda \left(r + \sum_{x \in \mathcal{I}} F_\alpha(x, y) \right) \right\}, \tag{7}$$

$$= \lambda \sum_{y \in \mathcal{S}} \left\{ r + \sum_{x \in \mathcal{I}} F_\alpha(x, y) \right\}, \tag{8}$$

$$= \lambda \left\{ \sum_{y \in \mathcal{S}} r + \sum_{y \in \mathcal{S}} \sum_{x \in \mathcal{I}} F_\alpha(x, y) \right\}. \tag{9}$$

Thus, using our Gillespie construction, we assume that the time until the next infection event, from the current time t, is distributed $Exp(A)$. The probability then, that the next infected individual is y is,

$$\frac{R_y}{A} = \frac{\lambda \left(r + \sum_{x \in \mathcal{I}} F_\alpha(x, y) \right)}{\lambda \left\{ \sum_{y \in \mathcal{S}} r + \sum_{y \in \mathcal{S}} \sum_{x \in \mathcal{I}} F_\alpha(x, y) \right\}}, \tag{10}$$

$$= \frac{r + \sum_{x \in \mathcal{I}} F_\alpha(x, y)}{\sum_{y \in \mathcal{S}} r + \sum_{y \in \mathcal{S}} \sum_{x \in \mathcal{I}} F_\alpha(x, y)}, \tag{11}$$

and we can see that this does not depend on λ.

Thus our modified Gillespie algorithm is given in Algorithm 11.

Modified Gillespie algorithm for SCYLV data:

Inputs: Background infection parameter, r; underlying infection parameter, λ; Force of infection function, $F_\alpha(\cdot)$; Standard deviation of the force of infection, α.

1. Initialise the process by defining the set of locations of initial susceptible individuals, $\mathcal{S}$, and the set of locations of initial infected individuals, $\mathcal{I}$.
2. Begin at time-step t = 6 with the 6 known initial infected individuals, while $I < 700$ or $t < 35$;
   (a) Calculate the overall rate of infection, A, given in Eq. 9.
   (b) Draw the time until the next event, $\tau \sim Exp(A)$.
   (c) For all the susceptible individuals, y, in $\mathcal{S}$, calculate the probabilities that they are the next infected individual, $\frac{R_y}{A}$, as given in Eq. 11.
   (d) Using the above set of probabilities, choose which susceptible individual, y, becomes infected.
   (e) Update the sets $\mathcal{S}$ and $\mathcal{I}$ to reflect the infection of the new individual.
   (f) Set $t = t + \tau$, and record the i.d. of the newly infected individual and the time, t.

ALGORITHM 11. The algorithm presented here is adapted from Neal & Xiang (2017)[24].

The algorithm can return a list of all the individuals in the population and the times at which they became infected, if they did. We assume the initial infected individuals were infected at time 0, before the process began. We know that if we let the algorithm run indefinitely then all the individuals will become infected, but we are aiming for epidemics with a final size of 584 in 30 weeks. Thus we can stop simulating the epidemic shortly after these milestones occur, so as not to waste resources, as we will only be interested in the first 30 weeks or first $584 + \epsilon$ individuals anyway.

11. Applying the methods

In this section we apply the ABC, Semi-coupled ABC, and Sequential-ABC algorithms to the the Sugarcane Yellow Leaf Virus dataset, utilising the model laid out in the previous section. We cannot use the EBC as we have a very large amount of data and no sufficient statistics, and we cannot use the Coupled ABC algorithm because we have more than one parameter, and most of them can't be decoupled.

11.1. **ABC for SCYLV data.** We start by applying the basic ABC algorithm to the Sugar Cane Yellow Leaf Virus (SCYLV) dataset to find samples from the approximate posterior distribution for (α, r, λ). We will use the model as described above in §10.2.

For our summary statistics there are many considerations. Firstly, as stated this is an SI epidemic, and will infect the entire population given enough time, so simply looking at final size will not give us much information. We have temporal data, the details of which individuals are infected at a series of time points, $t_0 = 0 < t_1 < \cdots < t_m$. This allows us to match simulations that progress at a similar rate. Similarly to the model, let $\mathcal{I}_i$ denote the set of locations of infected individuals at time point t_i, and a similar concept for $\mathcal{S}_i$ and the susceptible individuals. Then, following Neal (2017)[24], for $i \in (1, 2, \ldots, m)$, we define $\mathcal{W}_i = \{\mathcal{I}_i\} \backslash \{\mathcal{I}_{i-1}\}$, to be the set of locations of individuals infected between time points t_{i-1} and t_i. We also set $n_i = |\mathcal{W}_i|$, which we can interpret as the number of new infected individuals since the last time we looked. We can then take the number of new infected individuals between each time point, $(n_1, n_2, \ldots, n_m)$, as the first of our summary statistics.

As mentioned, the key benefit of this data is the spatial information it contains. We wish to use summary statistics that take advantage of this, and match simulations with similar spatial spread of the disease. To match the specific plants that became infected and their locations would make the acceptance rate prohibitively small. Instead, we consider the spatial distribution of the infected individuals, as opposed to their actual location. For instance, are all the infected individuals on one side of the lattice? If so, then we would want simulations which also had all of their infected individuals one side of the field. We would not, however, distinguish between whether they were on the left half or the right half. Following Neal (2018)[22], we have chosen to use the statistic Moran's $\mathtt{I}$[21] to measure the spatial distribution of the infected plants. Moran's $\mathtt{I}$ takes a value between -1 and 1, -1 representing perfect mixing (imagine a chess board where infected plants are the black squares and the susceptibles the white), 1 represents perfect segregation (all the black squares on one side and all the white squares on the other), and 0 represents perfectly random mixing.

Thus, for this data set, our time points are given by $(6, 11, 15, 19, 23, 30)$, with the convention that $t_0 = 0$. Our summary statistics are given by

$$\begin{aligned} S(\cdot) &= (n_1, \ldots, n_6, \mathtt{I}_1, ..., \mathtt{I}_6), \\ &= (6, 18, 42, 150, 103, 265, -0.00136, 0.00095, 0.00231, 0.00831, 0.01300, 0.01870) \end{aligned}$$

where $\mathtt{I}_i$ is the value of the Moran's $\mathtt{I}$ statistics at time point t_i.

We have chosen our distance function, $d(\cdot)$, to be the $L^2 - norm$, with a tolerance for each type of summary statistic, $(\epsilon_n, \epsilon_{\mathtt{I}})$. We do not have any intuition about the relative importance of each type of summary statistic, so we will treat them separately instead of taking a scaled weighted average.

Since $r \in [0, 1]$, we choose a $Uniform[0, 1]$. We initially have no intuition for α, thus we give it a $Uniform[0, 5]$ prior. Similarly we have no intuition for λ, but we ran a series of pilot runs of the model with randomly drawn r and α values, and an $Exp(30)$ prior for λ seemed appropriate. We can think of λ as a speed parameter, that dictates how fast

the epidemic spreads, and an $Exp(1)$ prior, as would usually be our first choice, resulted in epidemics that were far too fast, lasting only a few weeks at most before reaching the desired number of infected individuals.

We ran the ABC algorithm to obtain 250 samples from the approximate posterior distribution for (α, r, λ). We use the set of tolerances, $(50, 0.02)$. The results are presented in Table 13.

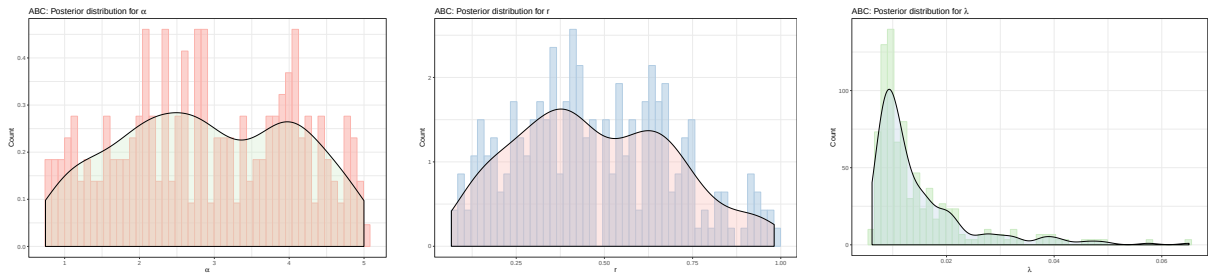
Sims	Acceptance rate	Time
92771	0.27%	331748 seconds

$\mathbb{E}[\alpha\|\mathbf{X}]$	$\text{sd}(\alpha\|\mathbf{X})$	$\mathbb{E}[r\|\mathbf{X}]$	$\text{sd}(r\|\mathbf{X})$	$\mathbb{E}[\lambda\|\mathbf{X}]$	$\text{sd}(\lambda\|\mathbf{X})$
2.8929	1.1639	0.4735	0.2227	0.0148	0.0095

TABLE 13. A summary of the ABC results for the SCYLV data.

The algorithm took 92,771 simulations to find 250 accepted samples from the approximate posterior distribution for (α, r, λ), taking a total of 331,748 seconds (around 92 hours). We see from Table 13 that our acceptance rate is rather high, 0.27%, which is most likely due to our very lax tolerance of 50 on the n_i. This tolerance means that in any given time period there could up to 50 greater, or fewer, infected individuals than in the data. For instance, in the first time period we are aiming for 18, but would accept anything between 0 and 68. The severely increased processing time on the other hand is due to the complexity of the model. For the posterior means of α and r, we notice that they are very close to the means of their prior distributions. We can see from Figure 5 that they are not completely uniform, and they do have peaks, but as we can also see from Table 13 their standard deviations are rather large. We would not have much faith that the algorithm has been able to approximate the posterior well. This may be because of the very lax tolerance, but it could also be due to the summary statistics.

For λ, we can see from Figure 5 and Table 13 that it has a distinct peak, with a posterior mean of 0.0148 which is rather different from the prior mean of 0.0333. This does not mean, however, that we can trust in our inference for λ, as it may have strong correlation with α and r. Ideally we would want to run the algorithm with a much stricter tolerance to have more certainty, but the time to run the algorithm for even this many samples, at such a large tolerance, prohibits greatly what we can do. We next intend to try other ABC methods that could increase our acceptance rate and/or reduce our tolerance.



(A) (B) (C)

FIGURE 5. A histogram of the approximate posterior density for (A) α, (B) r, and (C) λ, that was produced by the ABC algorithm for the SCYLV dataset.

11.2. **Semi-coupled ABC for SCYLV data.** As we saw in §8, we can improve on the acceptance rate of the ABC algorithm using the semi-coupled ABC. In the previous example about Measles with vaccinations, we used a Sellke construction, however, this is not necessary to be able to use the semi-coupled ABC. We notice that in Eq. 11, the probability that individual y gets infected next, R_y/A, does not depend on λ, and that in the overall rate of infection in Eq. 9, λ can come to the front. As we said before, we can think of λ as a speed parameter, its only effect is to change the duration of the epidemic. Notice that using the rescaling property,

$$Exp(A) = Exp\left(\lambda\left\{\sum_{y\in\mathcal{S}} r + \sum_{y\in\mathcal{S}}\sum_{x\in\mathcal{I}} F_\alpha(x,y)\right\}\right) \stackrel{D}{=} \frac{1}{\lambda} Exp\left(\left\{\sum_{y\in\mathcal{S}} r + \sum_{y\in\mathcal{S}}\sum_{x\in\mathcal{I}} F_\alpha(x,y)\right\}\right), \tag{12}$$

where $\stackrel{D}{=}$ implies is equal in distribution, and $Exp(A)$ is the distribution of τ, the time until the next event. Thus we can separate λ from the simulations by simulating with $\lambda = 1$, and the then choosing the value of λ that ensures the 584^{th} individual is infected at $t = 30$.

We use a reduced set of the summary statistics from the ABC in the previous section. Namely, the number of new infected individuals between each time point $(n_1, \ldots, n_6)$, and Morans `I` statistic, though this time we only look at it for the last time point. We do this under the assumption that if the spatial distribution of infected individuals is similar at the end of the epidemic, then chances are it won't be too dissimilar throughout. So our set of summary statistics is $S(\cdot) = (n_1, n_2, n_3, n_4, n_5, n_6, \mathtt{I}_6)$. We again use the $L^2 - norm$ as our distance function, but this time we will use a separate tolerance for every summary statistic, $(\epsilon_{n_1}, \ldots, \epsilon_{n_6}, \epsilon_{\mathtt{I}_6})$. This gives us the ability to allow some time intervals more lenience, if the algorithm is having trouble finding matches.

We again chose a $Uniform[0, 5]$ prior for α and a $Uniform[0, 1]$ prior for r. As stated we have decoupled λ and will choose the value that ensures the 584^{th} individual is infected at $t = 30$. We ran the Semi-coupled ABC algorithm to obtain 250 samples from the approximate posterior distribution for (r, α, λ). We used the set of tolerances, $(30, 30, 30, 50, 30, 30, 0.02)$. The results are presented in Table 14.

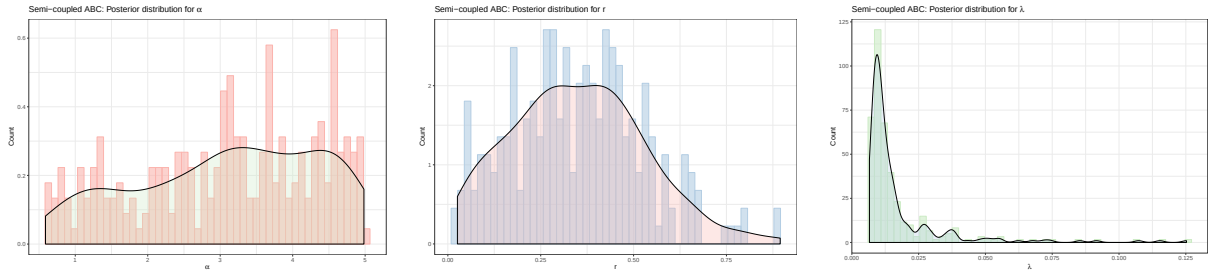
Sims	Acceptance rate	Time
50692	0.49%	379407 seconds

$\mathbb{E}[\alpha\|\mathbf{X}]$	$\text{sd}(\alpha\|\mathbf{X})$	$\mathbb{E}[r\|\mathbf{X}]$	$\text{sd}(r\|\mathbf{X})$	$\mathbb{E}[\lambda\|\mathbf{X}]$	$\text{sd}(\lambda\|\mathbf{X})$
3.0987	1.2444	0.3499	0.1781	0.0173	0.0172

TABLE 14. A summary of the ABC results for the SCYLV data.

The algorithm took a $50,692$ simulations to find 250 samples from the approximate posterior distribution, almost doubling the acceptance rate of the ABC. We notice from Table 14, however, that the algorithm took longer to run, $379,407$ (around 105 hours) seconds compared to $331,748$ seconds (around 92 hours). This will be because in the ABC algorithm we could stop the simulation as soon as it diverged too far from the data. In the Semi-coupled ABC on the other hand, we have to generate the full epidemic in order to choose the λ value that fits the data best. This would still be better if the inference were improved. We note that we were able to have a much stricter tolerance, much higher acceptance rate, and it did not take much extra time, so in those regards this algorithm is probably still better. As for the estimates, we first see that the posterior means for α and r are a bit different to their prior means this time, though the standard deviations are about the same as they were in the ABC. Figure 6 shows that the posterior for r is

much more peaked, though for α it is still rather uniform. Our estimate for the posterior mean and standard deviation of λ is similar to what we saw from the ABC algorithm. In total the results aren't too different to those from the last algorithm, though this may not be a positive attribute. Finally, we wish to apply the Sequential-ABC algorithm.



(A) (B) (C)

FIGURE 6. A histogram of the approximate posterior density for (A) α, (B) r, and (C) λ, that was produced by the Semi-coupled ABC algorithm for the SCYLV dataset.

11.3. **Sequential-ABC for SCYLV data.** We saw in §9 that we can improve on both the acceptance rate and the accuracy of our algorithms by using a Sequential-ABC algorithm. The Sequential-ABC algorithm works by running a series of basic ABC algorithms with smaller and smaller tolerances, and priors based on the accepted samples of the previous iteration.

We again use the set of summary statistics $S(\cdot) = (n_1, n_2, n_3, n_4, n_5, n_6, \mathtt{I}_6)$, where n_i is the number of newly infected individuals at time point t_i since time point t_{i-1}, and $\mathtt{I}_6$ is the value of Moran's `I` statistic at time point t_6. Similarly, for the distance function, $d(\cdot)$, we again use the $L^2 - norm$, with a tolerance for every summary statistic, $(\epsilon_{n_1}, \ldots, \epsilon_{n_6}, \epsilon_{\mathtt{I}_6})$, to allow us greater control for matching epidemics.

From a series of pilot simulations, we noted that the Moran's `I` statistic is always quite close to zero, and we have also chosen a tolerance that still excludes a proportion of simulations, so we do not feel the need to reduce it further from 0.02, and it does not cause the rejection of so many simualtions as to feel the need to start it larger. Thus we keep $\epsilon_{\mathtt{I}_6} = 0.02$ for all iterations of the Sequential-ABC algorithm, the series of choices for $(\epsilon_{n_1}, \ldots, \epsilon_{n_6})$ are given in Table 15. We initially choose a $Uniform[0, 5]$ prior for α, a $Uniform[0, 1]$ prior for r, and an $Exp(30)$ prior of λ.

We initially intended to run the Sequential-ABC algorithm for 7 iterations, with the last iteration having the tolerance set $(20, 20, 20, 50, 20, 20, 0.02)$. The algorithm ran for 120 hours but was unable to complete before it had to be stopped due to time constraints. At this point it had generated 80 samples from approximate posterior distribution under tolerance set 7. Unfortunately these were not recoverable. We estimate it would have taken around 170 hours more to find the remaining samples. This raises an issue with the Sequential-ABC, that it is rather difficult to do pilot runs and estimate run times, as the prior at each stage is dependent on the posterior of the previous.

We ran the Sequential-ABC algorithm to obtain 250 samples from the approximate posterior distribution for (α, r, λ). The algorithm went through 5 iterations, using a total of 184339 simulations, which took a total of 118126 seconds (around 33 hours). We generated 250 samples in each iteration, and use the set from the final iteration as our samples from the approximate posterior distribution for (α, r, λ), the results of which are presented in Table 16.

We immediately see from Table 16 that this algorithm requires the greatest number of simulations so far, double what the ABC required, and almost four times as many

Iteration	ϵ_{n_1}	ϵ_{n_2}	ϵ_{n_3}	ϵ_{n_4}	ϵ_{n_5}	ϵ_{n_6}	$\epsilon_{\mathtt{I}_6}$
1	100	100	100	100	100	100	0.02
2	80	80	80	80	80	80	0.02
3	70	70	70	70	70	70	0.02
4	60	60	60	60	60	60	0.02
5	40	40	40	60	40	40	0.02

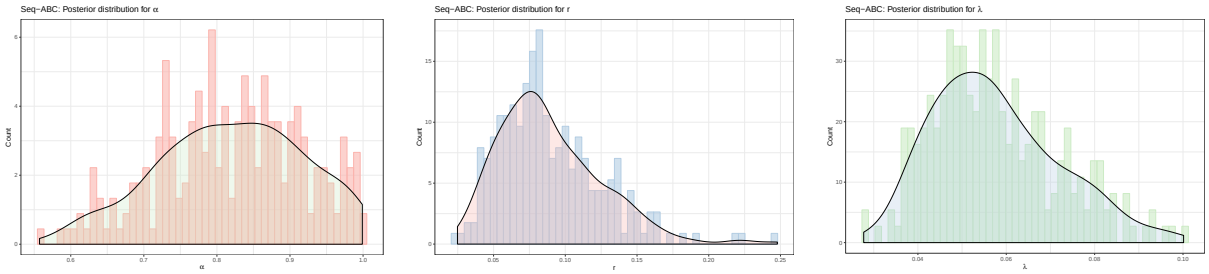
TABLE 15. The series of tolerances used in the Sequential-ABC algorithm for the SCYLV data.

Sims	Acceptance rate	Time
184339	0.14%	118126 seconds

$\mathbb{E}[\alpha\|\mathbf{X}]$	$\text{sd}(\alpha\|\mathbf{X})$	$\mathbb{E}[r\|\mathbf{X}]$	$\text{sd}(r\|\mathbf{X})$	$\mathbb{E}[\lambda\|\mathbf{X}]$	$\text{sd}(\lambda\|\mathbf{X})$
0.8191	0.0993	0.0894	0.0364	0.0575	0.0143

TABLE 16. A summary of the ABC results for the SCYLV data.

as the Semi-coupled ABC. We also notice that because of this the acceptance rate is the lowest so far. The stated acceptance rate was calculated using only the accepted samples from the final iteration as a proportion of the total simulations, as this arguably is all we are interested in. If, however, we calculated the acceptance rate based on all accepted samples, then it is around 0.68%, the highest of any of the algorithms. Despite the much larger number of simulations required, the Sequential-ABC algorithm was the fastest of the three, taking only a third of the time of the other two to find its samples. Turning to the approximate posterior, despite having arguably a more lax tolerance than the Semi-coupled ABC, the Sequential-ABC seems to have done a much better job at approximating the posterior distribution. The posterior means for α and r no longer appear to just be distortions of their prior means, and from Figure 7 we can see that the posterior of α appears to look less uniform, and r has a distinct peak. The standard deviations of α and r are also dramatically improved. As for λ, its posterior mean is almost 4 times as large as the previous estimates, but its standard deviation remains about the same. It must be said, however, that our tolerance is still rather large, so it is hard to have confidence in our inference at this point.



(A) (B) (C)

FIGURE 7. A histogram of the approximate posterior density for (A) α, (B) r, and (C) λ, that was produced by the Sequential-ABC algorithm for the SCYLV dataset.

12. Extensions to our SCYLV investigation

The fact that all three of our algorithms struggled to make inference on the data, even with such large tolerances, is disconcerting. As we have stated previously, our choices

can greatly effect the efficiency of the ABC algorithms. We have seen previously how our choice of prior can greatly affect our accuracy and acceptance rate, but in this case the most important aspect may be our model. We begin by investigating the data and showing where our ABC algorithms may be running into issues, from there we detail some possible alternatives that could be investigated given more time.

We stated originally that our model could affect the efficiency of our ABC algorithms, but up until this point we have only been using simple datasets, which make our choice of model, and model assumptions, a less important factor. This dataset however is far more complex, not just in the type of data it contains, but also in the pattern of the outbreak. We have assumed in our model, a constant underlying infection rate, λ. This may not have been appropriate, as we can see in Figure 8 that the number of newly infected individuals has a sharp spike between weeks 15 and 19, and then goes back down in the subsequent period, weeks 19 to 23. Since we assumed a homogeneous infection rate across the whole process, it is our belief that the our model simply cannot reproduce this pattern. We can see from Figure 8 the progress of all the accepted samples from the three different algorithms. We note that none of them particularly match the data, but all try to minimise the distance from it. They do this by having a quicker infection rate at the beginning, and a slower one at the end. The model simply cannot reproduce the same spike between weeks 15 and 19. We highlight in Figure 8b, in an emboldened red, one trajectory that manages to match the data at the beginning of the outbreak and at the end, but we can see in doing so it cannot match the weeks 15 to 19 interval. In fact it is a distance of 50 away at that point, the maximum the tolerance will allow.

On the other hand, the models assumption of a homogeneous infection rate may not be appropriate in the long run, but it may be applicable in short periods. For this reason we believe it would be beneficial to estimate the parameters and infection rate for each period independently. This would mean we were now working with final size data which has a spatial element. This reduces the information we have, which will reduce the amount of inference we can make, however, we should still be able to adapt the algorithms to make inference on the parameters we are interested in.

We will run the algorithms for each time period $tp_i = (t_{i-1}, t_i), i \in (1, \ldots, 6)$, where $t_0 = 0$ weeks. Since we no longer have temporal information, we cannot use any summary statistics relating to the temporal progress of the outbreak, namely, the number of new infected individuals between time point t_{i-1} and t_i, n_i. Running the ABC and seqABC algorithms for tp_i, we can simply use the final size of the outbreak at t_i weeks as one of our summary statistics. Recall, however, that we need at least as many summary statistics as we have parameters in order to make inference on all of them. Final size data does not provide us with any other viable summary statistics, so we need other summary statistics based on the spatial aspects of the data. Any spatial summary statistics should also help infer α, which the previous algorithms struggles with. We suggest using the number of infected individuals on each row, for all 17 rows, $(c_1, c_2, \ldots, c_{17})$. Combined with the Moran's `I` statistic, this should give a good picture of the spatial distribution of the epidemic. We would have to be careful with the tolerance however, as being too strict could dramatically reduce the acceptance rate, as there is a lot of information to match. If this was causing an issue, we could also consider the sum of the differences between the number of plants in each row in the simulation and those in the data, $\mathcal{C} = \sum_{i=1}^{17} |c_i - c_i^{\chi}|$. We could interpret this as how many infected plants are in the wrong row, though note that assuming our final sizes match, the effect of having a infected plant in row j that should be in row i so to speak, will increase this metric by 2, not just 1. We suggest that a good starting point for a tolerance could be aiming for 75% of the infected plants in the data to be in the right row, plus the tolerance of the final size. This is because, if we need

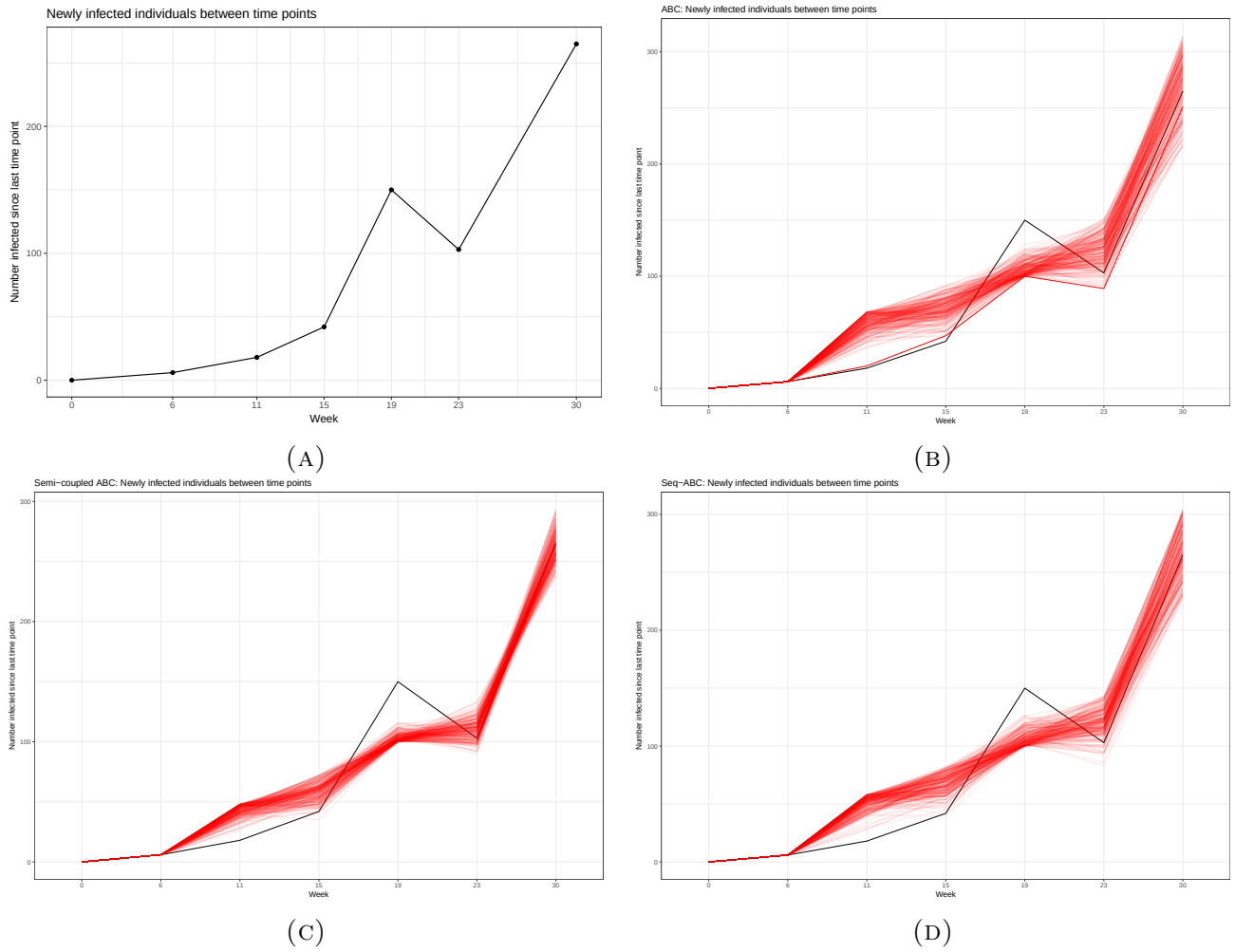


FIGURE 8. (A) The number of newly infected individuals since the last timepoint for the SCYLV dataset, with the accepted samples from the (B) ABC, (C) Semi-coupled ABC, (D) Sequential-ABC overlaid.

a final size of 100 say, and our simulation matches perfectly except it has 101 infected individuals, then that additional 1 infected will increase $\mathcal{C}$ by 1 regardless of which row it is in. For instance, for a final size of 100, a starting point for the tolerance could be $((1 - 0.75) \times 100) \times 2 + \epsilon_M$. The tolerance can then be adapted from there to attain the desired acceptance rate. Other summary statistics could include similar notions for the number of plants in each column, or splitting the field into sectors and counting the number of plants in each sector. We intended to pursue this line of investigation, however, unfortunately we did not have the time. We do note however that it would be much faster to run these algorithms than the previous set.

13. Discussion

Throughout this dissertation we have seen a myriad of ABC methods applied to real and simulated datasets with at least some degree of success. We have seen that some of the issues regarding acceptance rates and accuracy for these simple examples can be alleviated with the use of more advanced methods. This at least shows that ABC methods will provide use with useful inference in simple cases. It would be foolish, however, to declare that ABC can be used to make accurate inference of epidemics in general. There

is a long way to go before ABC should be considered a viable alternative to MCMC and other such methods.

We have preached throughout this dissertation about the importance of our choices for the ABC. The efficiency and accuracy of the algorithm can be greatly affected by our choice of summary statistics, distance function, prior distribution, tolerance, and model.

In §8.3 we saw an example of how our choice of prior distribution could affect our inference when replacing our $Exp(1)$ prior with an $Exp(0.1)$ prior increased our estimate of the posterior mean by 23%. Nothing else about the algorithm was changed. In this case we have no way of knowing which of the results is correct. If ABC were to be used in practice, and this were a fresh dataset that had never been analysed before, how could we assess our accuracy when such a simple change drastically changes our inference? In this case we used MCMC estimates as our benchmark, but in cases where MCMC is a viable option it is unlikely we would choose the approximate inference of ABC methods. In §6.2 Ex. 3, we saw the effect that our choice of prior distribution could have on the acceptance rate, when after the pilot run we changed from our $Exp(1)$ prior for both β and γ to $Exp(2000)$ and $Exp(4)$ priors, respectively. At the same tolerance as in the pilot run, our acceptance rate jumped from 0.0009% to 0.23%, which is over 250 times greater. Now many algorithms are not unfamiliar with the concept of tuning, we would not get around this particular problem by using MCMC, however, ABC methods come with additional issues in this regard. In MCMC methods we may wish to tune our parameter in such a way as to give us a desired acceptance rate, in fact, MCMC methods have an optimal acceptance rate. This, however, may not be appropriate for ABC methods. Neal (2018) [22] recommends an acceptance rate of 1%, however, we have seen from results throughout this dissertation that a 1% acceptance rate would lead to wildly inaccurate results. But it is not just a matter of choosing a smaller acceptance rate to be our rule of thumb. If we choose an acceptance rate of 1%, then the samples generated from a prior which is centred around the true values will be much more representative of the posterior than an uninformative prior would produce. Thus it is not just the acceptance rate that depends on the prior, but also the accuracy, which means this method of choosing a desired acceptance rate is not applicable across ABC methods. To add to these problems, the tolerance is also a major factor in the acceptance rate and accuracy of the algorithm, and the two are not independent.

To choose a tolerance, it is often desirable to complete a pilot run to identify acceptance rates at different tolerances, given our choices. This however is not always a viable method. For instance, in the coupled-ABC (§7.2) we could not run pilot simulations since the variable λ was chosen post-hoc to be best match the data. It would also not have been possible for the SCYLV data (§11) since the model takes so long to simulate compared to the simple models of the previous sections, that we would not have been able to run enough simulations to understand what tolerances are appropriate. Even in cases where we can run pilot studies, if we have no intuition about the parameters, then our metric for choosing a tolerance must be the acceptance rate. The other issue that arises is that tolerances, in most cases but the simplest, are not interpretable. They depend on both the summary statistics and the distance function. Thus a tolerance of 10 can be large in one algorithm and extremely small in another. This, arguably, is to be expected, but it means that without extensive data exploration or pilot runs it is very difficult to choose a tolerance based on intuition. In algorithms such as the Sequential-ABC this is often a difficult issue, as we should be able to achieve a lower final tolerance, but it is difficult to tell how low that can be, and how big the jumps should be. For instance, our Seq-ABC algorithm for the SCYLV data (§11.3) orignially had a smaller tolerance, but it proved to be too big a jump and meant the algorithm could not complete in its

maximum allotted run time. There are, however, methods which choose the tolerances automatically at each stage, which can alleviate this issue to some degree. A common method by Drovandi & Pettitt (2011)[12] is to choose a tolerance based of the α quantile of the distances, calculated using the distance function $d(\cdot)$, for the accepted samples from the previous iteration. In this case α has to be tuned. It is also worth noting that Kypraios et al. (2016)[19] pointed out that this method can have issues with discrete summary statistics, in which case minor modifications to the algorithm need to be made.

We did not see any explicit examples in this dissertation of how the choice of summary statistics can affect the efficiency of the ABC algorithms, however the question still remains of how do we pick summary statistics in the first place? Most of the summary statistics used in this paper were either adapted from the previous inference of others, or chosen based on intuition. These summary statistics may not have been the best possible however, they may have even been quite poor, but the simplicity of the models meant that any damaging effect was mitigated. Not only do we have to find summary statistics that explain the data well, we also cannot have too few, or too many. Having too many summary statistics not only reduces the acceptance rate, but also allows for minor discrepancies to be accepted, which distort the approximation of the posterior[28]. Prangle (2015)[28] lays out multiple methods that can be used to select a set of summary statistics, the general approach being to start with a large set of summary statistics, and then choose the best subset. They also detail other methods that might be of interest. Similarly the distance function plays an important role, though less work has been done to investigate its effect[27]. Prangle (2015*)[27] talks about common choices for the distance function, like those we have used in this dissertation, but also point out that these may not be appropriate for iterative algorithms such as our Sequential-ABC. In this case they suggest some methods for updating (or tuning) their distance function as the algorithm iterates.

Finally, the model we choose has a great effect on the quality of our inference. For the simple examples the assumptions of our models made little difference, even though in some cases they were inaccurate, such as in Ex. 5 with the Gastroenteritis data, which is known to have a incubation period of 1-3 days[19]. In the complex case of the SCYLV data (§10), however, we noted in §11 that the model had a great effect on our inference, as its assumptions were clearly incorrect since the simulations did not match the data as seen in Figure 8. This caused the acceptance rate to be extremely low, even using large tolerances with the more sophisticated algorithms, and for the inference they made to be somewhat worthless. We would have have similar issues if we specified an incorrect model using MCMC, but since MCMC gives exact inference, we could be more certain that it is our model that is at fault.

There are many more considerations that we have not even considered that would affect the efficiency of ABC algorithms, but what they all amass to is this: How can we trust that the inference we have made with ABC is correct? ABC methods only approximate the posterior, if the values look reasonable, there is nothing about the outputs that can say whether the approximation is good or bad. The tolerance is usually non-interpretable, so its size cannot be used as a measure of accuracy, apart from knowing the accuracy increases as the tolerance tends to 0, though not linearly. Whereas, if we were to use MCMC methods, we know we would be finding draws from the exact posterior distribution for the parameters of our model, even if they are dependent, and even if the model has incorrect assumptions.

Before we were to use these methods in practice we would wish to do a much deeper investigation into the effect that our choices have, and the methods that have been developed to counteract these. For instance, Beaumont et al. (2002)[4] introduced the method

of local linear regression, which helps correct the bias that is introduced when taking the tolerance $\epsilon > 0$. We would initially wish to do a formal treatment of the effect that different choices have on our inference, using a series of simulations and real data. For instance, we would wish to run a large number of ABC algorithms using different prior distributions to see how they shifted the posterior, and measure the effect on the acceptance rate. We would wish to test different summary statistics and distance functions to attempt to measure the bias that different choices introduced. Then perhaps we could look into implementing local linear regression to see to what degree it helped alleviate the issue. For all of these investigations we could also look at a range of tolerances, to investigate how the acceptance rate and accuracy may develop as the tolerances get closer to 0. These investigations would obviously not give definitive answers on the efficiency of ABC in general, but it would give us a much better idea of what it is capable of, and how robust it actually is. There are many ways of improving the algorithms, and new methods are being developed regularly, but we have yet to read about any methods that can inform us of how confident we should be in our approximate inference.

References

[1] M. Baguelin, J. R. Newton, N. Demiris, J. Daly, J. A. Mumford, and J. L. N. Wood. Control of equine influenza: scenario testing using a realistic metapopulation model of spread. *Journal of The Royal Society Interface*, 7(42):67–79, 2010.

[2] Norman T.J. Bailey and Anthony S. Thomas. The estimation of parameters from population data on the general stochastic epidemic. *Theoretical Population Biology*, 2(3):253 – 270, 1971.

[3] MARK A. BEAUMONT, JEAN-MARIE CORNUET, JEAN-MICHEL MARIN, and CHRISTIAN P. ROBERT. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[4] Mark A. Beaumont, Wenyang Zhang, and David J. Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[5] Niels Becker. A general chain binomial model for infectious diseases. *Biometrics*, 37(2):251–258, 1981.

[6] Michael G. B. Blum and Viet Chi Tran. Hiv with contact tracing: a case study in approximate bayesian computation. *Biostatistics*, 11(4):644–660, 2010.

[7] TOM BRITTON and PHILIP D. O'NEILL. Bayesian inference for stochastic epidemics in populations with random social structure. *Scandinavian Journal of Statistics*, 29(3):375–390.

[8] Ellen Brooks-Pollock, Gareth O Roberts, and Matt Keeling. A dynamic model of bovine tuberculosis spread and control in great britain. 511:228–31, 07 2014.

[9] Victor M. Cceres, David K. Kim, Joseph S. Bresee, John Horan, Jacqueline S. Noel, Tamie Ando, Connie J. Steed, J. John Weems, Stephan S. Monroe, and James J. Gibson. A viral gastroenteritis outbreak associated with person-to-person spread among hospital staff. *Infection Control and Hospital Epidemiology*, 19(3):162–167, 1998.

[10] Irina Chis Ster and Neil M. Ferguson. Transmission parameters of the 2001 foot and mouth epidemic in great britain. *PLOS ONE*, 2(6):1–12, 06 2007.

[11] Jean Heinrich Daugrois, Carine Edon-Jock, Sandrine Bonoto, Jean Vaillant, and Philippe Rott. Spread of sugarcane yellow leaf virus in initially disease-free sugarcane is linked to rainfall and host resistance in the humid tropical environment of guadeloupe. *European Journal of Plant Pathology*, 129(1):71–80, Jan 2011.

[12] C. C. Drovandi and A. N. Pettitt. Estimation of parameters for macroparasite population evolution using approximate bayesian computation. *Biometrics*, 67(1):225–233.

[13] W.J. Edmunds, N.J. Gay, Mirjam Kretzschmar, Richard Pebody, and H Wachmann. The pre-vaccination epidemiology of measles, mumps and rubella in europe: Implications for modelling studies. 125:635–50, 01 2001.

[14] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[15] T Déirdre Hollingsworth. Controlling infectious disease outbreaks: Lessons from mathematical modelling. *Journal of Public Health Policy*, 30(3):328–341, Sep 2009.

[16] Thomas House, Joshua V. Ross, and David Sirl. How big is an outbreak likely to be? methods for epidemic final-size calculation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 469(2150), 2013.

[17] Matt Keeling, Pejman Rohani, and Babak Pourbohloul. Modeling infectious diseases in humans and animals. 47:864–865, 10 2008.

[18] Margaret Kosmala, Philip Miller, Sam Ferreira, Paul Funston, Dewald Keet, and Craig Packer. Estimating wildlife disease dynamics in complex systems using an

approximate bayesian computation framework. *Ecological Applications*, 26(1):295–308.

[19] Theodore Kypraios, Peter Neal, and Dennis Prangle. A tutorial introduction to bayesian inference for stochastic epidemic models using approximate bayesian computation. *Mathematical Biosciences*, 287:42 – 53, 2017. 50th Anniversary Issue.

[20] E. S. McBryde, G. Gibson, A. N. Pettitt, Y. Zhang, B. Zhao, and D. L. S. McElwain. Bayesian modelling of an epidemic of severe acute respiratory syndrome. *Bulletin of Mathematical Biology*, 68(4):889–917, May 2006.

[21] P. A. P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.

[22] Peter Neal. *Approximate Bayesian Computation methods for epidemic models.* To appear in - Handbook of Infectious Disease Data Analysis. Editors: Leonhard Held, Niel Hens, Phil O'Neill and Jacco Wallinga.

[23] Peter Neal. Efficient likelihood-free bayesian computation for household epidemics. *Statistics and Computing*, 22(6):1239–1256, Nov 2012.

[24] Peter Neal and Fei Xiang. Collapsing of non-centred parameterized mcmc algorithms with applications to epidemic models. *Scandinavian Journal of Statistics*, 44(1):81–96.

[25] P. D. ONeill and G. O. Roberts. Bayesian inference for partially observed stochastic epidemics. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 162(1):121–129.

[26] M. Paunio, H. Peltola, M. Valle, I. Davidkin, M. Virtanen, and O. Heinonen. Explosive school-based measles outbreak: Intense exposure may have resulted in high risk, even among revaccinees. *Amer. J. Epidemiology*, 1998.

[27] D. Prangle. Adapting the ABC distance function. *ArXiv e-prints*, July 2015.

[28] D. Prangle. Summary Statistics in Approximate Bayesian Computation. *ArXiv e-prints*, December 2015.

[29] John A. Rice. *Mathematical Statistics and Data Analysis.* Belmont, CA: Duxbury Press., third edition, 2006.

[30] Thomas Sellke. On the asymptotic distribution of the size of a stochastic epidemic. *Journal of Applied Probability*, 20(2):390–394, 1983.

[31] Norman T.J Bailey. The mathematical theory of infectious diseases and its applications. 34, 01 1975.

[32] McKinley Trevelyan, Cook Alex R, and Deardon Robert. Inference in epidemic models without likelihoods. *The International Journal of Biostatistics*, 5(1):1–40, 2009.

[33] World Health Organisation. Vector-borne diseases: Key facts. `http://www.who.int/en/news-room/fact-sheets/detail/vector-borne-diseases`, 2017. [Online; accessed 2018-09-01].

14. Appendix

The R code used in this dissertation can be found at `https://github.com/BenjamenSimon/Approximate-Bayesian-Computation-for-Epidemics`.