Big Data Epidemics

Benjamen Simon, BSc (Hons), MSc

School of Mathematics and Statistics

Lancaster University

A thesis submitted for the degree of

Doctor of Philosophy

January, 2024

Big Data Epidemics

Benjamen Simon, BSc (Hons), MSc.

School of Mathematics and Statistics , Lancaster University

A thesis submitted for the degree of *Doctor of Philosophy*. January, 2024.

Abstract

Epidemic data inference is a key tool for the control and eradication of infectious disease spread. In the modern data age, where epidemic surveillance makes data abundant, the current methods of epidemic inference are no longer sufficient. Bovine Tuberculosis is endemic in the UK and affects tens of millions of cattle each year, with data available spanning decades (APHA, 2023c). There were 21 million confirmed cases of COVID-19 in England, from a population of roughly 56 million people, over a 3 year period (UK Health Security Agency, 2023). There are also around 1 billion cases of seasonal Influenza per year worldwide, resulting in up to $650,000$ deaths (World Health Organisation, 2023). The current gold-standard methods are incapable of making timely and efficient inference on big data epidemics at the individual level. In this thesis we introduce novel methodology that uses discrete-time population-aggregated approximations of epidemic data to make accurate and efficient inference for complex large-scale epidemics, whilst vastly reducing the computational burden. We apply these methods to a case study of Bovine Tuberculosis in England and Wales, including a novel method of incorporating movement data. We believe the methods developed in this thesis could form part of a multi-pronged approach for understanding and combating epidemics and pandemics of the scale we are now experiencing.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the bibliography. A rough estimate of the word count is: 55149

Benjamen Simon

COVID-19 Statement

This work was undertaken during the COVID-19 Pandemic. A 3 month funding extension was granted on the grounds of supervisor unavailability due to being co-opted onto the government SPI-M modelling committee, a month of technical issues that halted work that could not be fixed due to lockdown, mental health issues and challenging personal circumstances resulting from the pandemic, and familial health issues. In addition I undertook a 3 month secondment to the UKHSA to aid in the modelling of COVID-19.

Benjamen Simon

Contents

List of Tables

List of Figures

List of Algorithms

List of Subroutines

Chapter 1

An Introduction to Epidemic Modelling

1.1 Introduction

The ability to produce timely, accurate, and insightful inference for epidemic and pandemic data is of vital importance in modern society (Epstein, 2008, Isham and Medley, 1996, Woolhouse, 2003). When performed correctly, in conjunction with policy and communication, the insights gained through data can have a tremendous impact on society's ability to control and eradicate disease in the population (McBryde et al., 2020, Kao, 2002). The recent COVID 19 pandemic has demonstrated the usefulness and necessity of inference for epidemic data (McBryde et al., 2020), but it has also highlighted the many challenges that exist (Xiang et al., 2021, Shinde et al., 2020, Brunsdon, 2020).

Epidemic data are highly interdependent, with events that occur in the epidemic dependent on the infectious status of the individuals during the epidemic, however the status of individuals is only partially observed in many cases. We can perhaps know when an individual recovers, but not when they were infected or who infected them. This missing data complicates the process of deriving insights from the data, leading to the need of advanced statistical methodologies. These methodologies

have the potential to provide useful insights, however, each epidemic requires a bespoke solution, and for epidemic data on the scale of COVID-19 and other big-data epidemics, combined with the complexity of the disease dynamics, these standard methods can become prohibitively computationally expensive and inefficient. These challenges lead to the need for new methodologies and frameworks that can make accurate and efficient inference on complex large data epidemics.

The core concept of epidemic modelling involves dividing a population into distinct states related to disease status, and modelling the transitions of individuals between these states. They exist within the general class of State Transition Models (STM). The form of the STM is determined by the application. Some use agent-based models that treat every individual in the population distinctly, and some concern themselves with population-level dynamics (Ajelli et al., 2010). Some take into account complex social network or other covariate information (Ajelli et al., 2010), whilst others deal with simple population counts (Zhou, Ma, and Brauer, 2004). The states of the models are sometimes simplified to a small selection (Kamrujjaman et al., 2022), and others have a large number of states to represent different infectious pathways and histories (Overton et al., 2022).

The model we choose is often based on our assumptions about the disease and population in question, and possibly dictated by the resolution and scale of available data. These models have the advantage that they are typically very easy to simulate from. We use this property to gather insights on the behaviour of an epidemic, given a set of parameters. We derive the most appropriate parameters by making inference on an epidemic data set, under the assumptions of our model.

Fitting epidemic data to models is a difficult task even in the simplest of cases, due to two features that set infectious disease data apart from non-communicable disease data; it is both highly dependent, and often only partially observable, leading to censoring.

The dependence is between the dynamics of the epidemic and state of the population. As an example, there can be no infections, or mechanically no

transitions from the susceptible state to the infectious state, if the number of individuals in the infectious state is 0. The overall rate of transitions at a given time t is dependent on the state of the population. Unlike for instance heart disease, where the risk of disease for individual i is independent of all other members of the population.

The second, and arguably greater, challenge is that we often cannot observe the transmission process of an infectious disease, only the outcome; we can see who recovers or is removed from the population due to death or some other event, but not who infected them or when. We call this a partially observable process. These missing events can be divided into two groups. We know that for each observed removal/recovery event, there must be an associated infection event. We call these events partially observed. We know they must have happened, but we don't know when. On the other hand, it is possible that there are individuals that are infected but have not yet recovered, meaning we have no knowledge of their infection. We call these occult events. These missing information make calculating the likelihood of the epidemic difficult, or more often than not, impossible. In these cases we can turn to advanced statistical methodologies such as Markov Chain Monte Carlo (MCMC). These Bayesian methods allow us to "fill-in" the missing information and obtain posterior distributions for the parameters we are interested in. These methods have revolutionised the analysis of partially observed infectious disease data and have been successfully applied to a myriad of diseases such as COVID-19 (Mbuvha and Marwala, 2020; Taghizadeh, Karimi, and Heitzinger, 2020), Foot-and-mouth (Jewell et al., 2009b; Streftaris and Gibson, 2004), and Influenza (Cauchemez et al., 2004; Huang et al., 2016).

To add to the challenge, non-standard and problem specific algorithms have to be designed in each instance to optimise efficiency and accuracy. When the models become more complex, or the population too large, the cost of computing the likelihood can become very high, and the scale of missing data that needs to be imputed can make the methods highly inefficient. Still they are one of the best

options. Whilst there are likelihood-free alternatives available such as Approximate Bayesian Computation that come with their own advantages (Csilléry et al., 2010), especially potentially when it comes to computational costs and complex big-data problems, these methods only give approximate inference and no guarantees of accuracy. As such, we strongly believe that likelihood-based inference methods, even in the case of complex big-data epidemics, are still worth pursuing.

In this thesis we will be focusing on the challenge of inference for complex big-data epidemics, using the case study of Bovine Tuberculosis in England and Wales. The inference methodology we will be concerned with is the general family of State Transition Models and full likelihood data inference using Markov Chain Monte Carlo (MCMC) methods.

In this chapter we will develop the core principles and methodologies used to make inference on epidemic data using State Transition Models and MCMC inference techniques. This chapter forms the basis for all the work in the thesis that follows. In Section 1.2 we develop the core methodology of State Transitions Models for epidemics. Under these assumptions, in Section 1.3 we describe the framework of Markov Chain Monte Carlo methodologies for making inference on epidemic data. Finally in Section 1.4 we review some of the greatest challenges of modelling epidemics, and highlight the direction of this Thesis for addressing these challenges in order to make full likelihood inference for complex big-data epidemics.

1.2 Continuous-Time State Transition Models

We begin with a natural but simplified model, treating the population members as identical and assuming they mix homogeneously. This is the General Stochastic Epidemic model introduced by Bartlett, 1949. Epidemic models can broadly be divided into two classes, deterministic and stochastic. Whilst both are valuable tools and have their uses, we will be focusing solely on stochastic models in this thesis. Deterministic models can be seen as averages of large population dynamics

of stochastic models, and as such are in continuous time and on a continuous state-space. Stochastic models are better able to quantify the uncertainty associated with epidemic model parameters whilst accounting for complex disease dynamics and heterogeneity in disease spread which are often features of big-data epidemics. In addition the gold standard methods of Markov Chain Monte Carlo for fitting these models allow us to efficiently augment the large amount of missing data often present in epidemic data sets, and we intend to prove these methods are viable for big-data epidemics. Finally, we have modelled our case-study example, Bovine Tuberculosis, as a collection of connected but separate epidemics in small disjoint populations (farms) - a meta-population model. The stochastic fluctuations of epidemics in small populations have a much more significant effect than in larger populations, and as such the deterministic models inability to capture this behaviour can lead to inaccuracies. In this section we will derive the construction of the basic classes of S-I-R model.

We begin with a natural model, treating the population members as individuals, with their own infection and removal times on a continuous scale, who interact with other individuals depending on their covariates such as spatial positioning. Starting with individual level agent-based models and continuing by making simplifying assumptions to addresses potential challenges with fitting the model.

There are a multitude of ways to model the spread of disease through a susceptible population in the state transmission model framework. We can include different states, different transition pathways, different mechanisms of disease spread, varying scale in time, space, and population (Ajelli et al., 2010, Overton et al., 2022, Zhou, Ma, and Brauer, 2004, Kamrujjaman et al., 2022). There are additional complexities we omit at this stage such as household models, meta-population 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.

1.2.1 The core concept of S-I-R models

Consider a population of N individuals. This population is closed, which means N is fixed and there are no births, deaths, emigration, or immigration. We divide this population into three states, each individual can only exist in one state at any given time. The states are;

- S - Susceptible. Individuals in this set are susceptible to infection, and will become infected when they come into contact with an infectious 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 are removed/recover.
- R - Recovered/Removed. Individuals in this set have been removed from the epidemic, either through recovery or death or quarantine or some other mechanism. They have no effect on individuals in either of the other two sets if they come into contact with them, cannot become infected again, and will remain in the removed state indefinitely. Removal grants immunity.

In the simplest case the model is then parameterised by two rates; β/N is the rate at which any given susceptible makes contact with an infectious individual (dividing by N so that the interpretation remains constant regardless of population size), and γ is the rate at which any given infectious individual transitions to the removed state, which can also be thought of as the reciprocal of the duration of an individual's infectious period. The values of these rates will be dependent on the dynamics of the epidemic itself. When these rates are identical for all individuals, we call this a homogeneously mixing model.

The S-I-R process concerns the sequence of transitions of individuals between these states through time. We define an epidemic as the series of infection and removal times in continuous time.

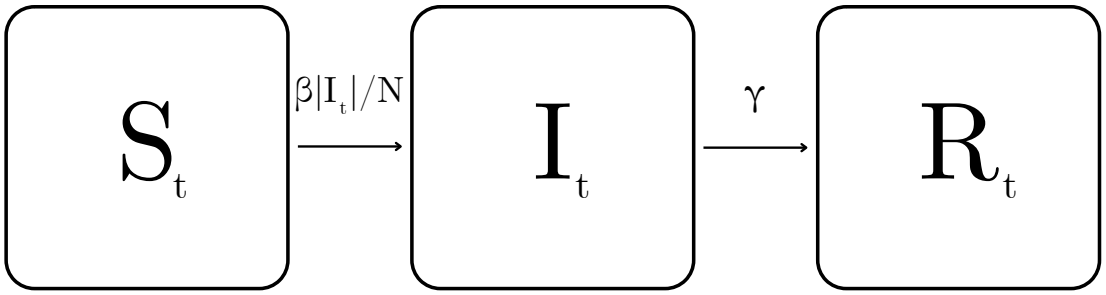


Figure 1.1: A diagram representing the transition of individuals between states and the transition parameters at time t. The population is divided into three disjoint sets; S_t (susceptibles), I_t (infectious), and R_t (removed). The number of individuals in each set (or size) is denoted as $|\cdot|$, for instance $|I_t|$. $\beta|I_t|/N$ is the rate at which any given susceptible transitions to the infectious state, and γ is the rate at which any given infectious individual transitions to the removed state.

1.2.2 The General Stochastic Epidemic S-I-R

The General Stochastic Epidemic (GSE) model is a State Transition Model in a closed homogeneously mixing population. In this section we consider the common example of the S-I-R model. Let us divide the population into the three states S, I, and R. When a susceptible individual comes into contact with an infectious individual they become infected, and at the end of an infected individual's infectious period they become removed. If we make the assumption of exponentially distributed infectious periods, then the memoryless property means that the process will be a Markov Chain (Bartlett, 1949). We define β/N to be the rate of contact between a given susceptible individual and an infected individual, and γ to be the removal rate.

If an individual i becomes infected at time, I_i, then they are infectious for a time of length $Q_i \sim \text{Exp}(\gamma)$, and are removed at time $R_i = I_i + Q_i$. In addition we will define X_t to be the number of susceptibles at time t, and Y_t to be the number of infected at time t.

At a given time t, we multiply the 'force of infection', $\frac{\beta}{N} \cdot Y_t$, by the number of

susceptible individuals to calculate the total 'infectious pressure' in the population, $\lambda_t = \frac{\beta}{N} \cdot X_t \cdot Y_t$, which is the overall population rate of S to I transition events. The overall population rate of I to R transition events is defined as $\gamma \cdot Y_t$. That is to say both depend on the states of individuals in the population. When there are no more infectious individuals there can be no more infections, and the epidemic is over.

As such, the waiting time until the next infection event is distributed Exponentially with rate $\frac{\beta}{N} \cdot X_t \cdot Y_t$, and the waiting time until the next removal event is distributed Exponentially with rate $\gamma \cdot Y_t$. These two processes are competing, and so the next event occurs at the minimum of two Exponential distributions, which is also an exponential distribution with the sum of the two rates. Thus the overall rate of events at time t is thus given by $\frac{\beta}{N} \cdot X_t \cdot Y_t + \gamma \cdot Y_t$ and is Exponentially distributed. Given we draw the waiting time until the next event from this overall rate, the probability that the next event is an infection event is the ratio of the susceptible to infectious transition rate to the overall rate, $\frac{(\beta \cdot X_t \cdot Y_t)/N}{(\beta \cdot X_t \cdot Y_t)/N + \gamma \cdot Y_t}$. Similarly, the probability that the next event is an infectious to removed transition event is the ratio of the infectious to removed transition rate to the overall rate, $\frac{\gamma \cdot Y_t}{(\beta \cdot X_t \cdot Y_t)/N + \gamma \cdot Y_t}$.

Using these rates we can simulate easily from this epidemic using what is known as a Gillespie algorithm (Bailey, 1975). The details of this algorithm can be found in Algorithm 1.1.

Gillespie Algorithm:

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

1. Initialise the states of the individuals at time $t = 0$. Typically with $N - 1$ susceptible individuals, 1 infectious individual, and 0 removed individuals.
2. While the number of infectious individuals is greater than 0,
   (a) Generate the waiting time until the next event as $\Delta t \sim \text{Exponential}(\frac{\beta}{N} \cdot X_t \cdot Y_t + \gamma \cdot Y_t)$.
   (b) The probability that the event is an S to I transition is given by $\frac{(\beta \cdot X_t \cdot Y_t)/N}{(\beta \cdot X_t \cdot Y_t)/N + \gamma \cdot Y_t}$. The probability that the event is an I to R transition is given by $\frac{\gamma \cdot Y_t}{(\beta \cdot X_t \cdot Y_t)/N + \gamma \cdot Y_t}$.
   (c) Generate whether the event was an infection or removal based on these probabilities, and update an individual's state appropriately.
   (d) Update the time $t = t + \Delta t$.

Algorithm 1.1: An algorithm to simulate a General Stochastic S-I-R epidemic in a closed, homogeneous population.

The overall epidemic process is thus the product of two independent but simultaneously occurring processes, the infection process and the removal process. Each of these processes is a Poisson process (Kingman, 1992). A Poisson process is a model of a sequence of discrete events where the average time between events is known, but the exact time until the next event is random. The waiting time until the next event is independent of the event before and the occurrence of one event does not affect the probability another event will occur (the memoryless property). The average rate must be constant (though there are non-homogeneous Poisson processes where the rate can vary through time (Cox and Isham, 1980)), and no two events can occur at the same time.

As such, the likelihood of an epidemic that was generated under the assumptions of the Gillespie algorithm is given by the following definition. In a population of N individuals, we have n_I infected individuals, and n_R removed individuals, resulting in $n_I + n_R$ total events including the initial infection. The infected individuals

belong to the set $\mathcal{I}$, and the removed individuals to the set $\mathcal{R}$. The initial infected individual is indexed by κ. The infection times, I_i for $i \in \mathcal{I}$, are contained in the set $\mathbf{I}$. The removal times, R_i for $i \in \mathcal{R}$, are contained in the set $\mathbf{R}$.

$$f(\mathbf{I}, \mathbf{R}|\beta, \gamma, I_\kappa) \propto \left[\prod_{j \in \{\mathcal{I} \setminus \kappa\}} \frac{\beta}{N} Y_{I_j -} \right] \cdot \exp\left\{ -\frac{\beta}{N} \int_{I_\kappa}^{T} (X_t Y_t)\, dt \right\} \cdot \left[\prod_{j \in \mathcal{R}} \gamma \right] \cdot \exp\left\{ -\gamma \int_{I_\kappa}^{T} (Y_t)\, dt \right\},$$

where,

- β is the infection rate,
- γ is the removal rate,
- I_κ is the initial infection time,
- X_t denotes the number of susceptibles at time t,
- Y_t denotes the number of infectious individuals at time t.

The $Y_{I_{j^-}}$ notation denotes the number of infectious individuals just before the infection time of individual j, I_j. Formally, $Y_{I_{j^-}}$ denotes the left hand limit, $Y_{I_{j^-}} = \lim_{s \uparrow I_j}(Y_s)$.

The Gillespie algorithm is a powerful tool in the arsenal of epidemic modellers, but naturally some diseases will not fit its assumption about a constant homogeneous contact rate between individuals who are considered equally likely to be infected. For instance the spatial locations of individuals may play a key role in their rate of contact (Lloyd and May, 1996). Some diseases require more specific models where each pair of individuals, $\{i, j\}$, has a unique pair-wise infection rate, $\beta_{i,j}$. This is known as a heterogeneously mixing population, where different individuals make infectious contact at different rates.

1.2.3 The Heterogeneous General Stochastic Epidemic S-I-R

An extension to the General Stochastic Epidemic (GSE) allows for the modelling of epidemics in heterogeneous populations by taking into account the contact rate between each pair of individuals. This means that knowledge of who infects whom is a core part of the system.

When a susceptible individual comes into contact with an infectious individual they become infected, and at the end of an infected individual's infectious period they become removed. This is still true as in the previous model, however now each pair of individuals, $\{i, j\}$, has a unique pair-wise infectious contact rate, $\beta_{i,j}/N$.

We are interested in the infection time and removal time for each individual in the population. Let I_i denote the time at which individual i becomes infected, and R_i is the time when individual i is removed. If an individual i becomes infected at time, I_i, then they are infectious for a time of length $Q_i \sim Exp(\gamma)$, and are removed at time $R_i = I_i + Q_i$. If an individual never becomes infected during an epidemic, then we define their infection and removal times to be infinity.

At a given time t, we can define the overall population rate of S to I transition events as $\lambda_t = \sum_{i \in I_t} \sum_{j \in S_t} (\beta_{i,j}/N)$, and the overall population rate of I to R transition events as $\gamma \cdot Y_t$. The overall rate of events at time t is thus given by $\sum_{i \in I_t} \sum_{j \in S_t} (\beta_{i,j}/N) + \gamma \cdot Y_t$. Thus the probability that the next event is an S to I transition is given by $\frac{\left(\sum_{i \in I_t} \sum_{j \in S_t} (\beta_{i,j}/N)\right)}{\left(\sum_{i \in I_t} \sum_{j \in S_t} (\beta_{i,j}/N)\right) + \gamma \cdot Y_t}$, and the probability that it is individual $k \in S_t$ that is infected is given by $\frac{\sum_{i \in I_t} (\beta_{i,k}/N)}{(\sum_{i \in I_t} \sum_{j \in S_t} (\beta_{i,j}/N))}$. The probability that the event is an I to R transition is given by $\frac{\gamma \cdot Y_t}{\left(\sum_{i \in I_t} \sum_{j \in S_t} (\beta_{i,j}/N)\right) + \gamma \cdot Y_t}$.

Algorithm 1.2 presents a method of simulating an S-I-R epidemic in a closed heterogeneous population under the General Stochastic Epidemic construction using these rates.

Heterogeneously Mixing General Stochastic Epidemic Simulation:

Inputs: Population size, N; Infection rates, $\beta_{i,j}$, for all $\{i, j\}$; Removal rate, γ.

1. Generate a set of individuals with covariates and calculate the value of $\beta_{i,j}$ for each pair of individuals $\{i, j\}$ using the chosen model definition.
2. Chose one individual at random, κ, to be the initial infected. Set $I_\kappa = 0$ and generate a new infectious period, Q_κ, from $Q_i \sim Exp(\gamma)$, and calculate R_κ.
3. While the number of infectious individuals is greater than 0,
   (a) Generate the waiting time until the next event as $\Delta t \sim$ Exponential$(\sum_{i\in I_t}\sum_{j\in S_t}(\beta_{i,j}/N) + \gamma \cdot Y_t)$.
   (b) The probability that the event is an S to I transition is given by $\frac{\left(\sum_{i\in I_t}\sum_{j\in S_t}(\beta_{i,j}/N)\right)}{\left(\sum_{i\in I_t}\sum_{j\in S_t}(\beta_{i,j}/N)\right)+\gamma\cdot Y_t}$. The probability that the event is an I to R transition is given by $\frac{\gamma\cdot Y_t}{\left(\sum_{i\in I_t}\sum_{j\in S_t}(\beta_{i,j}/N)\right)+\gamma\cdot Y_t}$.
   (c) Generate whether the event was an infection or removal based on these probabilities.
   (d) If it was an infection event, choose an individual $k \in S_t$ to become infected based on the probabilities $\sum_{i\in I_t}(\beta_{i,k}/N)/(\sum_{i\in I_t}\sum_{j\in S_t}(\beta_{i,j}/N))$.
   (e) If it was a removal event, choose an individual $k \in I_t$ uniformly at random.
   (f) Update the state of the individual k and the variable Y_t.
   (g) Update the time $t = t + \Delta t$.

Algorithm 1.2: An algorithm to simulate a General Stochastic S-I-R epidemic in a closed, homogeneous population.

Assuming that the data we have access to are the number of individuals in the population, and the time when each infected individual transitions from the infectious to removed state, but not when they became infected or who infected them, then the likelihood of a heterogeneously mixing epidemic that was generated under the assumptions of Algorithm 1.2 is given by the following definition. In a population of N individuals, we have n_I infected individuals, and n_R removed individuals, resulting in $n_I + n_R$ total events including the initial infection. The infected individuals belong to the set $\mathcal{I}$, and the removed individuals to the set $\mathcal{R}$.

The initial infected individual is indexed by κ. The infection times, I_i for $i \in \mathcal{I}$, are contained in the set $\mathbf{I}$. The removal times, R_i for $i \in \mathcal{R}$, are contained in the set $\mathbf{R}$.

$$f(\mathbf{I}, \mathbf{R}|\boldsymbol{\beta}, \gamma, I_\kappa) \propto \left[\prod_{j \in \{\mathcal{I}_T \setminus \kappa\}} \left(\sum_{i \in \mathcal{I}_{I_{j-}}} \frac{\beta_{i,j}}{N}\right)\right] \cdot \exp\left\{-\int_{I_\kappa}^{T} \left(\sum_{i \in \mathcal{I}_t} \sum_{j \in \mathcal{S}_t} \frac{\beta_{i,j}}{N}\right) dt\right\} \cdot \left[\prod_{j \in \mathcal{R}_T} \gamma\right] \cdot \exp\left\{-\gamma \int_{I_\kappa}^{T} Y_t \, dt\right\},$$

where,

- $\beta_{i,j}$ is the infectious contact rate between individuals i and j,
- γ is the removal rate,
- I_κ is the initial infection time,
- $\mathcal{S}_t$ is the set of susceptible individuals at time t,
- $\mathcal{I}_t$ is the set of infected individuals at time t,
- $\mathcal{R}_t$ is the set of removed individuals at time t,
- Y_t denotes the number of infectious individuals at time t.

The $\mathcal{I}_{I_{j^-}}$ notation denotes the number of infectious individuals just before the infection time of individual j, I_j. Formally, I_{j^-} denotes the left hand limit, $\mathcal{I}_{I_{j^-}} = \lim_{s \uparrow I_j}(\mathcal{I}_s)$.

1.3 Inference for S-I-R epidemics

We are interested in identifying the posterior distribution of our parameters - that is the distribution of the parameters after considering our current beliefs and the new evidence from the data. This in turn will provide us with the most likely values of the parameters, and a measure of the uncertainty in our estimates. With a fitted

model we can make inference on such things as how long the epidemic will last, how many hospital beds will be needed this winter (Overton et al., 2022), or where in the country the disease is likely to spread to next (Brooks-Pollock, Roberts, and Keeling, 2014). It is the fitting of the model to the data that is the most complex and challenging part of epidemic modelling.

In the models considered in this chapter, the epidemic data that is required to fit these models include the infection times and removal times for each individual, and potentially knowledge of the infectious pathways (who infected whom). In many cases we assume we have the removal times, or a viable proxy such as taking the removal time to be the diagnosis time. Most often, however, we do not have knowledge of the infection times, or who infected whom. These components then need to be treated as missing information, which needs to be inferred or augmented. Data augmentation refers to the introduction of latent variables that depend on the distribution of the existing variables in such a way that the resulting conditional distributions are easier to sample from and/or result in more efficient sampling algorithms. This means that frequentist methods of inference such as maximum likelihood estimation are inherently difficult for epidemics. As such, we prefer to use Bayesian methods of inference. In particular, the most common method used for full likelihood-based inference is Markov Chain Monte Carlo (MCMC). There are a selection of non-likelihood-based methods that will not be covered here such as Approximate Bayesian Computation (Csilléry et al., 2010).

1.3.1 Bayesian Methods

Bayesian methods utilise Bayes' Theorem, which is used to calculate the conditional probabilities of events, to make inference on the posterior distributions of parameters. Bayes' Theorem states that for parameters θ and data X,

$$\pi(\theta|X) = \frac{f(X|\theta)\pi(\theta)}{f(X)},$$

where, $\pi(\theta|X)$ is the posterior distribution of θ, $f(X|\theta)$ is the likelihood of X given θ, and $\pi(\theta)$ represents our prior beliefs about θ. The denominator, $f(X)$, is called the evidence, or the normalising constant, and is a constant that ensures that the posterior distribution integrates to 1. The normalising constant can also be written as

$$f(X) = \int_{-\infty}^{\infty} f(X|\theta)\pi(\theta)d\theta.$$

For the scenarios we are interested in, it is often not possible to calculate the value of the normalising constant. However, as it is constant, Bayes' Theorem then also states that the posterior distribution of θ is proportional to the likelihood multiplied by the prior,

$$\pi(\theta|X) \propto f(X|\theta)\pi(\theta).$$

This opens up the possibility of methods such as Markov Chain Monte Carlo, which allow us to generate samples from a target distribution without needing to explicitly compute the normalising constant.

1.3.2 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) methods are conceptually simple, highly customisable and extendable, and have a rich body of research and tools behind them. In simple terms, they are methods that allow us to draw dependent samples from the joint posterior distribution of our parameters of interest. We can do this by drawing samples from a known friendly distribution and accepting or rejecting that sample based on some probability derived using the prior distribution and likelihood of our parameters. This in turn allows us to construct an empirical distribution of the parameters, giving us valuable insights.

There are many benefits to using MCMC methodology to make inference on epidemics. Firstly, we can deal with our missing-data problem using a method known as 'data augmentation'. This involves treating each of our missing data points as a parameter of the model, we refer to them as 'nuisance parameters', and

then making inference on them in the same way we do for our parameters of interest. Unlike with imputing missing data, data augmentation is not intended to produce point estimates of the most likely value for the missing data, but instead to explore the posterior distribution of all possible values of the truth. Exploring the space of possibilities will thus allows us to better quantify our confidence in our estimates of the parameters of interest and future findings.

Next, as stated above MCMC methods produce samples from the joint posterior distribution of our parameters. When the conditional distributions of the parameters are of a known form (such as a Gaussian or Gamma distribution) it is easy to produce samples given a current set of parameter values, and we can use a method known as a Gibbs sampler to get draws directly from them. However, in the case of epidemics, these distributions often do not correspond to a common distributional form, which would usually make drawing samples from them very challenging. MCMC, however, has methods such as the Metropolis-Hastings step that allow us to make dependent draws from any posterior distribution, as long as we can calculate the likelihood. The conditional posterior distributions - the posterior distributions for some of the parameters given values of the others - we are usually interested in for epidemics are often low in dimension and typically uni-modal, so these methods have the potential to work well.

Next, the rich body of research supporting MCMC methods, and the vast array of tools for a multitude of problems available, equip the method well for combating the unique and varied challenges of emerging epidemics. There are methods to support unique distributions, improved efficiency, rapid speed of inference for ongoing epidemics, and countless more (O'Neill, 2002, Sherlock, Fearnhead, and Roberts, 2010).

Finally, apart from the inherent computational costs, MCMC methods can be reasonably simple to implement, and there are many papers, packages, and books to support the creation and evaluation of these methods.

There are however some drawbacks. The previously mentioned computational

costs of implementing an MCMC algorithm can be prohibitively expensive, and methods to alleviate these burdens can become very complex. In addition, the samples of the posterior distributions will usually be dependent on each other. In intuitive terms this means that the amount of information in our sample set is not equal to the amount of samples we have. We can quantify this using auto-correlation and effective sample size. Auto-correlation is the correlation between two chains offset by t positions, i.e. the chain start at sample n and the chain starting at sample $n + t$. The greater the auto-correlation within the chains, the greater the uncertainty in our estimates, which we can measure using effective sample size. Effective sample size is the number of independent samples with the same estimation power as N auto-correlated samples, and can be calculated as the ratio of the number of dependent samples, N, to the sum of the auto-correlations for all lags (t) from $-\infty$ to ∞ (Gelman, Carlin, et al. (2013)). In extremely inefficient implementations this could mean that a million dependent samples has an effective sample size in the single digits. Finally, whilst MCMC is asymptotically guaranteed (given the correct conditions are satisfied) to provide draws from the correct posterior distribution, we typically expect the algorithm to take time to converge to this 'stationary' distribution, and it can be difficult to confirm if it has happened during a run. The method may encounter issues such as finding different 'stationary' distributions based on how it is initialised, or getting stuck in local minima.

Even with these considerable drawbacks, MCMC is still one of the best tools for making inference on epidemic data. Next we will lay out a general framework for how to perform MCMC for epidemics, which will be utilised in the remainder of the Thesis.

1.3.3 Components of MCMC

Let our data be denoted by $\mathbf{I}$ and $\mathbf{R}$, the infection and removal times respectively, and our model parameters by θ. During this thesis we will consider models that do not assume or require knowledge of the infectious pathways (who infected whom),

though having observations of infectious pathways can improve the inference. To perform MCMC inference on this data we need two core components:

- The likelihood function of the epidemic, $f(\mathbf{I}, \mathbf{R}|\theta)$,
- Prior distributions for each parameter, $\pi(\theta)$,

The likelihood is constructed based on the model we have chosen for the epidemic, which should reflect our beliefs about the process that generated the data. The prior distributions on the parameters we can choose freely to reflect our prior knowledge about the parameters, or choose a form that complements the form of our likelihood. The joint posterior distribution can then be derived using these two components.

The joint posterior distribution of the parameters and any missing data is proportional to the likelihood multiplied by the prior. In some cases it will be possible to construct conditional posterior distributions for some of the parameters and/or missing data analytically, given values of other parameters/missing data, such that they have common distributional forms that we are familiar working with, like Gamma or Gaussian distributions. In these instances we can sample directly from these conditional distributions using standard statistical techniques. Often, however, it is not possible or is undesirable to construct these conditional distributions with common forms. In these cases methods such as a Metropolis-Hastings step will be necessary to acquire samples from the posterior distribution, and for those we will also need:

- Proposal distributions to draw samples from, $q(\theta'|\theta)$,
- Metropolis-Hastings acceptance probabilities, $\alpha(\theta, \theta')$, to help decided whether to accept the proposed sample.

1.3.3.1 Posterior Distribution

The posterior is the distribution of a parameter vector, including nuisance parameters, after we have updated our prior knowledge or assumptions with the evidence

from our new data. Mathematically it is proportional to the product of the prior distribution of the parameters and the likelihood of the data. If we know very little about the parameters, we can use an 'uninformative' prior, such as a uniform distribution, that gives equal weight to all possibilities. If we know more then we can use a distribution that can reflect the information and uncertainty we believe or assume to be true.

To derive the posterior distribution of a parameter, let $f(X|\theta)$ be the likelihood of some data X that is generated from a model that is dependent on parameter θ. We first put a prior distribution on θ, $\pi(\theta)$, which represents what we currently know about the parameter. If we think θ has a mean of 3, we can reflect this in the prior for instance, and it will contribute to our inference. By Bayes' Theorem, the posterior is proportional to the likelihood multiplied by the prior;

$$\pi(\theta|X) \propto f(X|\theta)\pi(\theta)$$

When appropriate it can be of benefit to choose the prior on θ such that the prior and the likelihood of the data will be "conjugate", which will result in the posterior of θ taking the form of a known, 'friendly' distribution. If we have conjugacy for the full joint posterior of all the parameters and nuisance parameters, then we don't need to sample at all. If we have component-wise conjugacy for the joint conditional posterior of some of the parameters/missing data given the others, then we can sample directly from this joint conditional posterior for these elements. For those elements that do not have conjugacy, special methodology is needed to make our inference; the Metropolis-Hastings step.

1.3.3.2 The Metropolis-Hastings step

The Metropolis-Hastings (MH) step is a form of rejection sampling, that is used to generate samples from a posterior distribution. Typically it is used when it is not possible to generate samples from the distribution because we cannot calculate the

normalising constant of the posterior. Even if we are not able to sample from the posterior directly, we may be able to evaluate it's density for a given set of inputs. This will tell us how likely that value of the parameter is given the data and the prior, and possibly the current value of other parameters if there is dependency.

The Metropolis-Hastings step is made up of a proposal distribution and an acceptance probability function which combined generate a Markov chain of dependent samples from the stationary distribution of the chain, which once converged is set up to be the posterior distribution of interest.

Lets say we are interested in some parameter, θ. We first choose a proposal distribution for the parameter, q, and set $Q(\theta'|\theta)$ be the cdf of q. We can propose a value for $\theta' \sim Q(\cdot|\theta)$. Then we calculate the value of the posterior for our current sample, θ^*, and our proposed sample, θ'. Lets denote these $\pi(\theta^*|X)$ and $\pi(\theta'|X)$ respectively. Then, we will accept the newly proposed value with some probability α, which is known as the Metropolis-Hastings acceptance probability.

A common choice for the proposal distribution is a Normal distribution centred around the current value of the parameter, θ^*, with variance σ^2, where σ^2 is a tuning parameter. Choosing this proposal distribution makes this algorithm into the Random Walk Metropolis (RWM). Sampling from this proposal distribution gets us a proposal draw, θ', where $\theta' \sim N(\theta^*, \sigma^2)$.

The MH acceptance probability, α, is dependent on the posterior likelihood of the proposed parameter. It is calculated,

$$\alpha = \min\left\{\frac{\pi(\theta'|X)}{\pi(\theta^*|X)}\frac{q(\theta^*|\theta')}{q(\theta'|\theta^*)}, \quad 1\right\}, \tag{1.1}$$

where $q(\theta^*|\theta')$ is the probability of being at θ' and "moving to" θ^* on the proposal distribution, and vice versa for $q(\theta'|\theta^*)$. We can note here that as we are taking the ratio of the posterior calculated at two different values, the normalising constant would cancel, and so we have no need of calculating it.

In the case where we have chosen the proposal distribution to be Normally distributed, these proposals will be symmetric. This means that the probability

density of going from $\theta' \to \theta^*$ is the same as going from $\theta^* \to \theta'$, and so the components cancel;

$$\begin{aligned}\alpha &= \min\left\{\frac{\pi(\theta'|X)}{\pi(\theta^*|X)}, \quad 1\right\} \\ &= \min\left\{\frac{f(X|\theta')\pi(\theta')}{f(X|\theta^*)\pi(\theta^*)}, \quad 1\right\}.\end{aligned}$$

As we can see from what remains, if the new parameter is more likely than the current one then we just accept it, otherwise we accept it with probability the ratio of the two posteriors. If we accept it then we record the draw and update our current value, if we reject it then we just discard the draw and record our current value again.

We control this acceptance rate through our proposal variance, σ^2, however, our goal is not 100% acceptance, as this would just return the proposal distribution. Instead we are balancing two goals. If σ^2 is too large, then we will propose large "jumps" (the relative distance between our current and new draws is more likely to be large). When this happens the posterior density is likely to be very different and we are likely to reject these changes as a result (especially since the MHRW prefers to stay in areas of high posterior probability), however, when these changes are accepted the two draws will be a lot less dependent. On the other hand, if we have a small σ^2, then we are going to propose small "jumps" (the two draws are likely to be relatively similar). When this happens, we are likely to accept the new parameter as the posterior densities will be fairly similar, but there will be a high level of dependence or 'auto-correlation' between the two samples. We want to balance the dependence/auto-correlation with the number of samples accepted.

Under theoretical conditions, the optimal acceptance rate for a univariate Gaussian conditional posterior distribution when using a Gaussian Random Walk Metropolis proposal is 44% (Gelman, Roberts, and Gilks, 1996). This is taken as a rule of thumb for most single parameter inference. If we increase the number of parameters in each MH step, and use multivariate proposal distributions with covariance matrices, that ideal theoretical acceptance rate tends to 23% as the number

of parameters increases to ∞ (Roberts, Gelman, and Gilks, 1997).

There are many proposal distributions we could use, even a uniform distribution is a valid choice. Changing the proposal distribution is what leads to the many named MCMC algorithms such as MALA (Roberts and Tweedie, 1996) and Hamiltonian Monte Carlo (Duane et al., 1987), which can improve the efficiency of the inference in different circumstances. Many of the advanced ones depend on gradient information. These methods could be used on the parameter space, however they are more problematic for the data augmentation space because the likelihood can be discontinuous when you propose changes (ie. you don't just move along the likelihood curve, but shift the curve entirely).

1.3.4 MCMC framework

Here we will present a framework algorithm for how to perform MCMC inference for epidemic data, using the components described above, to obtain samples from $\pi(\theta|\mathbf{R})$.

The most commonly encountered scenario is that we know the removal times, $\mathbf{R}$, as we can define them ourselves (such as when someone gets diagnosed at the doctors), but the infection times, $\mathbf{I}$, are usually unknown, and so we assume that the infection times will be treated as nuisance parameters which will need to be inferred.

Assume we have an epidemic model $\mathcal{M}$ with parameters $\theta_1, \ldots, \theta_p$. Let us say that for all parameters $\theta_1, \ldots, \theta_i$, $i < p$, we have conditional posterior distributions, conditional on the values of the other parameters, the missing infection times data, $\mathbf{I}$, and the removal times data, $\mathbf{R}$, that we can easily draw samples from (Gamma distributions for instance), and the rest of the parameters will require a Metropolis-Hastings step.

In a population of N individuals, we have n_I infected individuals, and n_R removed individuals, resulting in $n_I + n_R$ total events including the initial infection. In this case we will assume the epidemic is complete and $n_I = n_R$. The infected individuals

belong to the set $\mathcal{I}$, and the removed individuals to the set $\mathcal{R}$. The initial infected individual is indexed by κ. The infection times, I_i for $i \in \mathcal{I}$, are contained in the set $\mathbf{I}$. The removal times, R_i for $i \in \mathcal{R}$, are contained in the set $\mathbf{R}$.

In this scenario a framework for an MCMC algorithm is given below in Algorithm 1.3.

MCMC framework for epidemics:

Inputs: Data, $\mathbf{R}$; Likelihood, $f(\mathbf{I}, \mathbf{R}|\theta)$; Prior for each $\theta_j, \pi(\theta_j), j = 1, \ldots, d$; Prior on $\mathbf{I}$, $\pi(\mathbf{I}|\theta)$; Exact conditional posterior distributions for each $\theta_{1:i}$; Proposal distributions for each $\theta_{(i+1):p}$, $q(\theta'_j|\theta_j), j = (i+1), \ldots, p$; Proposal distribution for $\mathbf{I}$, $q(\mathbf{I}'_k|\mathbf{I}_k), k = 1, \ldots, n_I$.

1. Initialise the algorithm by choosing values for the parameters and the nuisance parameters.

2. For $j = 1, \ldots, i$: Generate a new realisation of θ_j, θ'_j, from its conditional posterior distribution, conditional on the current value of all the other parameters, $\theta_{1:p\setminus j}$, the infection times, $\mathbf{I}$, and the removal times, $\mathbf{R}$. This is known as a Gibbs sampler. Update the θ vector: $\theta_j \leftarrow \theta'_j$.

3. For $j = (i+1), \ldots, p$: Generate a new realisation of θ_j, θ'_j, from its appropriate proposal distribution, conditional on the current value of all the other parameters, the infection times, and the removal times. Accept this sample with probability calculated using the Metropolis-Hastings acceptance probability α. If the new realisation is accepted, then update the θ vector: $\theta_j \leftarrow \theta'_j$, if it is rejected then discard the draw and update the θ vector: $\theta_j \leftarrow \theta_j$. This is known as a Metropolis-Hastings step.

$$\alpha = \min\left\{\frac{\pi(\theta'_j|\theta_{1:p\setminus j}, \mathbf{I}, \mathbf{R})}{\pi(\theta|\theta_{1:p\setminus j}, \mathbf{I}, \mathbf{R})}\frac{q(\theta_j|\theta'_j)}{q(\theta'_j|\theta_j)}, \quad 1\right\}$$

4. Choose a random infected individual $k \in \mathcal{I}$. Update the infection time of individual k, I_k, using again a MH step. If the new time is invalid for some reason (*e.g.* it occurs after the last removal say) then it should automatically be rejected as the likelihood should be 0. If the new realisations are accepted then update the infection times vector: $I_k \leftarrow I'_k$, if not then discard the new draw and update the infection times vector: $I_k \leftarrow I_k$.

$$\alpha = \min\left\{\frac{\pi(I'_k|\mathbf{I}_{\mathcal{I}\setminus k}, \mathbf{R}, \theta)}{\pi(I_k|\mathbf{I}_{\mathcal{I}\setminus k}, \mathbf{R}, \theta)}\frac{q(I_k|I'_k)}{q(I'_k|I_k)}, \quad 1\right\}$$

5. Repeat steps 2-4 T times then discard the first B draws as "burn-in" leaving $T - B$ draws (approximately) from the posterior distribution.

Algorithm 1.3: A general framework for performing MCMC inference for epidemic data.

Our output from this algorithm should be $(T - B)$ dependent draws from the

joint posterior distribution of our parameters. We discard the B burn-in samples to remove the period the algorithm spent 'finding' the stationary distribution of the parameters. From this we can estimate features of the posterior distributions that are of interest, such as means, medians, and variances. In addition we can feed the joint posterior samples back into the model to estimate features of the epidemic via simulation such as expected length, expected cases, probability of dying out naturally via averaging the simulation results. We can also use simulations to do projection and retrospective analyses.

Note that the framework presented here is for an MCMC method known as Metropolis-within-Gibbs, which combines the direct conditional sampling steps of a Gibbs sampler, with the proposed and accepted/rejected conditional sample steps of a Metropolis-Hastings algorithms. There are many possible MCMC methods that exist that may use significantly different tools to increase efficiency, but the core concept and goal will remain the same.

1.4 The Challenges of Epidemic Inference Addressed in this Thesis

We now have a way of building an empirical distribution of our parameters of interest based on likelihood methods that also account for missing data. There are however still challenges in making these methods feasible in the modern data age, and on the scale of epidemic and pandemic data.

1.4.1 The computational costs of MCMC

Markov-Chain Monte Carlo (MCMC) methods are typically very computationally expensive, even with optimised code implementations. For every proposed sample of the potentially thousands of parameters drawn, when including the nuisance parameters, typically a full recalculation of the likelihood is required. Of these

proposals, in line with the optimal rejection probabilities corresponding to the earlier optimal acceptance probabilities, 56% to 77% of samples are rejected and discarded in an optimally tuned algorithm. Of the accepted parameter draws, an indeterminable amount is discarded as burn-in before the chain locates the stationary distribution. These steps are necessary to ensure accurate inference and minimise the dependence between samples, as the greater the dependence between samples, the more samples need to be generated. For this reason, in complex modelling scenarios such as epidemics, where the likelihoods are large and expensive to compute, we may look for ways to reduce the computational burden of the algorithms at the cost of accuracy.

1.4.2 The missing data

Computational costs can, to some degree, be mitigated through additional compute power and intelligently designed algorithms, however, this does not get around the issue of missing data. As pandemics break out in ever growing populations, and epidemic data becomes easier to collect, store, and share, the scale of epidemic data becomes unwieldy. With data available at the individual level across whole countries, the level of missing data rises to tens or hundreds of millions of records which need to be imputed. Finding efficient ways of exploring the state space of all these missing data becomes a significant challenge, that more powerful hardware simply cannot address alone, as this corresponds to the mixing and efficiency of the inference algorithms. For this reason too we may look at ways of aggregating the data to different resolutions, in order to reduce the number of data points that need to be imputed, at the cost of accuracy.

If we could perform inference for an epidemic under the assumptions of the fully heterogeneous individual-level model presented in Section 1.2.3, it would be the gold-standard. This, however, is rarely possible, especially for anything larger than a small epidemic. In the following chapter, we propose a potential model to address some of the challenges presented here. We consider a model that takes

into account additional spatial covariate data in order to discretise the data and reduce the computational burden of computing the likelihood. We also show that this model is still incapable of overcoming the challenges of the largest epidemic and pandemic data sets available in the modern world, setting the scene for the rest of the thesis, which explores methods for making full likelihood inference on big-data epidemics with hundreds of millions of data points.

This thesis is concerned with novel methods for performing full likelihood inference using gold-standard MCMC methods for big-data epidemics. The first half of the thesis builds up the readers knowledge of making inference on epidemic data, whilst justifying the need for new methodology, and highlighting potential avenues of research. In Chapter 2 we introduce an individual-level continuous-time spatial epidemic model, that attempts to deal with large quantities of data through a simplified spatial kernel we call a "Near vs. Far" model. In Chapter 3 we explore discrete approximations for State Transition Models via a discrete-time population-level epidemic model, and how this approximation can vastly reduce the computational burden of the inference whilst maintaining high levels of accuracy, under the right conditions. The latter half of the thesis builds upon the previous chapters to build a model and an efficient full-likelihood inference scheme for a case-study example of a big-data epidemic, Bovine Tuberculosis. In Chapter 4 we introduce our case study example, giving an overview of the literature and previous models, and an exploratory analysis of data provided by the Animal and Plant Health Agency (APHA). Following this in Chapter 5 we develop our own model for bTB and an inference scheme based on a partially simulated epidemic, which uses some geographical, cattle, and movement data from APHA, and augments it with simulated badger, cattle testing, and epidemic data. With this inference method validated on simulated data, in Chapter 6 we remove the simulated data and adapt the model and inference method for only observed data. Finally in Chapter 7 we conclude the thesis with a review of the works completed and suggestions for future avenues of research.

Chapter 2

Near Vs. Far

2.1 Introduction

The gold standard of epidemic modelling is to consider every individual in continuous time, with unique time-varying pair-wise contact rates, possibly even taking into account each individual's time-varying covariates. However, for even small to moderately sized epidemics this becomes infeasible. There are identifiability issues due to the potentially large numbers of dependent parameters and potentially large amounts of missing data common to epidemic data sets. At the same time the MCMC methodologies become extremely inefficient and the computational burden of calculating the likelihood becomes unwieldy.

A simplification to the model that captures the majority of the behaviour and yet vastly reduces the computational burden can allow the methodology to scale to real-world challenges.

In this chapter we introduce one such example of a simplified spatial S-I-R model with heterogeneous mixing. Instead of having a unique infectious contact rate between every pair of individuals, the pair-wise infectious contact rate will take one of two fixed values, depending on the distance between the individuals on a plane. We call this model the 'Near vs Far General Stochastic Epidemic'. We also explore alternative parameterisations for the implementation of the Near vs

Far kernel to investigate the effect on the efficiency of inference. Deardon et al., 2010 investigated a similar simplification for the UK 2001 FMD epidemic they call linearising the distance kernel by taking a Taylor series expansion.

In this Chapter we apply the frameworks laid out in Chapter 1 for constructing the model and applying MCMC methodologies to make inference for a simulated epidemic. In Section 2.2 we explain the details of the model and give an algorithm for simulating an epidemic in this style. Following this in Section 2.3 we provide the form of the likelihood which is used in Section 2.4 to derive the posterior distributions. In Section 2.5 we present three additional forms of the proposal distribution, and apply these to the parameters and latent variables in Section 2.6. We use these components in Section 2.7 where we explain the MCMC schema applied to make inference for our example, and in Section 2.8 we present an alternative parameterisation of the epidemic model and subsequent MCMC algorithm. Finally we present and compare the results in 2.9.

2.2 Simplified Heterogeneity

The general stochastic epidemic (GSE) makes the assumption of a unique rate of infectious contact between each pair of individuals, $\beta_{i,j}$. However, fitting a model with a unique parameter for each pairing is often impossible. As such we tend to make $\beta_{i,j}$ a function of the covariates of the individuals to reduce the size of the parameter space.

There are many models we could consider to investigate heterogeneity in mixing. Depending on the disease in question and the modelling assumptions we have made, we could consider heterogeneity in individuals susceptibility or infectivity, or both, based on covariates such as spatial location, age, species, occupation, social structure, behaviour, vaccination status, and many more. Examples include,

1. An age-stratified model such as Balabdaoui and Mohr, 2020. For instance, let c_i be the age group covariate of individual i, then $\beta_{i,j} = \beta \cdot f(c_i, c_j)$ is dependent

on the contact rate between different age groups, ie. school children contact a lot of other school children but less commonly contact young adults.

2. A household model such as Neal and Roberts, 2004. The natural grouping of individuals into small units is modelled, as an example $\beta_{i,j} = \beta + \beta_H \cdot \mathbf{1}_{i \sim j}$ where all individuals contact at a base rate β and individuals in the same household have additional contact rate β_H, where $\mathbf{1}_{i \sim j}$ is equal to 1 if individuals i and j share a household, and 0 otherwise.

3. A spatial kernel that depends on the distance between individuals in space such as Keeling et al., 2001. An example could be a Gaussian spatial kernel where $\beta_{i,j} = \beta \cdot f(i,j)$ where $f(i,j)$ is a Gaussian density centred on the spatial location of individual i with standard deviations σ_x, σ_y as tuning parameters, which represent the scale of the Gaussian density along the x and y axes, respectively. Depending on the shape of the kernel, typically the further individual j is from individual i, the lower the contact rate.

There are countless more possibilities. In this section we present a simplified spatial model.

2.2.1 Near vs Far GSE S-I-R

We assume that each individual has a fixed location on a plane. When considering a pair of individuals, i, j, an infectious individual is more likely to make infectious contact and infect a susceptible individual if they are closer together. There are many ways that we could incorporate this kind of spatial heterogeneity, and we have chosen a conceptually simple one. If an infectious individual is within a Euclidean distance d of a susceptible individual, then they make infectious contact at rate β_1, otherwise, they make infectious contact at rate β_2. This has reduced the size of the parameter space from $N(N-1)$ to 2. We are considering cases where $\beta_1 > \beta_2$. Once an individual is infected they recover at rate γ. Figure 2.1 presents a visual representation of the model.

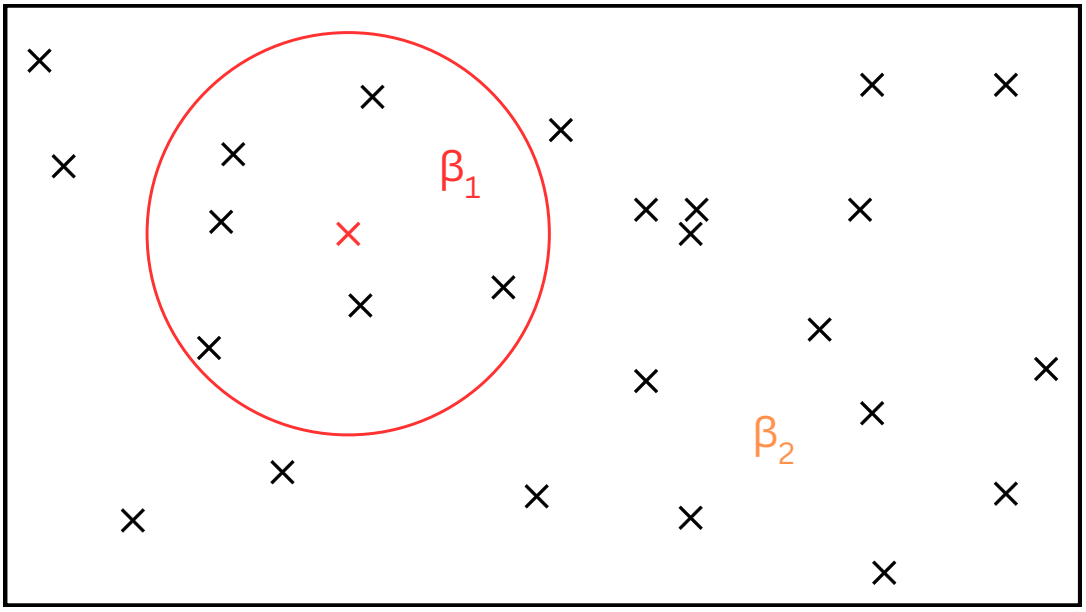


Figure 2.1: A diagram representing the Near Vs Far GSE. The "x" represent the positions of individuals on the 2-d plane. The red x represents an infected individual. The individuals within distance d of the infected individual are contained in the red circle, and make infectious contact with the infected individual at rate β_1. All individuals outside the red circle make infectious contact with the infected individual at rate β_2.

The General Stochastic Epidemic presented in Chapter 1 can be adapted to this kind of heterogeneity with the appropriate setting of the $\beta_{i,j}$. With its hard boundary and fixed spatial positioning it will not always be appropriate, but is viable for instance when modelling crop diseases or disease spread by cattle herds on different farms that share pastures. In cases where it is appropriate we can vastly reduce the computational costs involved in simulating and making inference on such data. Algorithm 2.1 presents the steps for simulating an epidemic in this framework, based on the Gillespie algorithm (Gillespie, 1977), where

$$\beta_{(i,j)} = \begin{cases} \beta_1, & \text{if} \quad \left\{d_{(i,j)} < d\right\}, \\ \beta_2, & \text{otherwise.} \end{cases} \tag{2.1}$$

Near Vs. Far General Stochastic Epidemic Simulation:

Inputs: Population size, N; Infection rates, β_1 and β_2; Removal rate, γ; Distance cut-off, d.

1. Generate the position of all individuals i on the plane. Calculate the distances $d_{i,j}$ between each pair of individuals $\{i, j\}$. Calculate the value of $\beta_{i,j}$ for each pair of individuals $\{i, j\}$.
2. Choose one individual at random, κ, to be the initial infected. Set $I_\kappa = 0$ and generate a new infectious period, Q_κ, from $Q_\kappa \sim \text{Exp}(\gamma)$, and calculate $R_\kappa = I_\kappa + Q_\kappa$.
3. For each $\{i, \kappa\}$ pair, $i \in$ Susceptibles, generate a time until contact, $W_{i,\kappa}$, from $W_{i,j} \sim \mathsf{Exp}(\beta_{i,\kappa})$.
4. Then,
   (a) For $i \in$ Susceptibles, $j \in$ Infectious, find the pair of individuals $\{i, j\}$ which has the minimum $\{I_j + W_{i,j}\}$ subject to $\{I_j + W_{i,j} < I_j + Q_j\}$. Set $I_i = I_j + W_{i,j}$. Generate Q_i, from $Q_i \sim \mathsf{Exp}(\gamma)$, and calculate $R_i = I_i + Q_i$.
   (b) Update the sets of Susceptible and Infectious individuals.
   (c) For the new infectious individual, i, from step (a), and each $m \in$ Susceptibles, generate a new time until contact, $W_{m,i}$, from $W_{m,i} \sim \mathsf{Exp}(\beta_{m,i})$.
   (d) Repeat steps (a) and (b) until the susceptible or infectious population reaches size 0.

Algorithm 2.1: An algorithm to simulate a General Stochastic S-I-R epidemic in a closed, homogeneous population.

2.3 Likelihood

The likelihood of the Near vs Far epidemic takes the form of the heterogeneous General Stochastic Epidemic presented in Chapter 1, with the pairwise infection rate defined as a function of the model parameters and the individuals covariates.

The likelihood is defined by its infection times and removal times, conditional on the transition parameters and the initial infection time, with no requirement to

know who infected whom. In a population of N individuals, we have n_I infected individuals, and n_R removed individuals, resulting in $n_I + n_R$ total events including the initial infection. The infected individuals belong to the set $\mathcal{I}$, and the removed individuals to the set $\mathcal{R}$. The initial infected individual is indexed by κ. The infection times, I_i for $i \in \mathcal{I}$, are contained in the set $\mathbf{I}$. The removal times, R_i for $i \in \mathcal{R}$, are contained in the set $\mathbf{R}$.

The model simplifies the pair-wise infectious contact rate, $\beta_{(i,j)}$, to two distinct possibilities, β_1 or β_2 of the form given in Equation (2.1).

The value of $\beta_{i,j}$ depends on d, the distance at which the infection rate changes, but we are only interested in the total susceptible pressure exerted on each individual. We introduce the notation D_t^1 to represent the number of susceptible-infectious pairs within distance d of each other at time t, and D_t^2 to represent the number of susceptible-infectious pairs further than distance d from each other at time t.

In this parameterisation the heterogeneous GSE likelihood simplifies to,

$$f(\mathbf{I}, \mathbf{R}|\boldsymbol{\beta}, \gamma, I_\kappa) \propto \left[\prod_{j \in \{\mathcal{I}_T \setminus \kappa\}} \left(D^1_{I_{j-}}\beta_1 + D^2_{I_{j-}}\beta_2\right)\right] \cdot \exp\left\{-\int_{I_\kappa}^{T} \left(D_t^1\beta_1 + D_t^2\beta_2\right) dt\right\} \cdot \left[\prod_{j \in \mathcal{R}_T} \gamma\right] \cdot \exp\left\{-\gamma \int_{I_\kappa}^{T} Y_t \, dt\right\}, \tag{2.2}$$

where,

- γ is the removal rate,
- κ indexes the initial infective,
- $\mathcal{S}_t$ is the set of susceptible individuals at time t,
- $\mathcal{I}_t$ is the set of infected individuals at time t,
- $\mathcal{R}_t$ is the set of removed individuals at time t,

- Y_t denotes the number of infectious individuals at time t.

The $\mathcal{I}_{I_{j^-}}$ notation denotes the number of infectious individuals just before the infection time of individual j, I_j. Formally, I_{j^-} denotes the left hand limit, $\mathcal{I}_{I_{j^-}} = \lim_{s \uparrow I_j}(\mathcal{I}_s)$.

2.4 Posterior

In this section we present the posterior distributions of the parameters in the Near vs. Far model, using the methodology explored in Chapter 1.

Individuals are arranged at fixed points on a plane, and the infection rate between two individuals i and j is dependent on the Euclidean distance between them. The infection rate takes the form given in Equation (2.1).

We have chosen the prior distribution on β_1 to be a Gamma$(\nu_{\beta_1}, \lambda_{\beta_1})$ because of its flexibility, non-negative support, and interpretability. The prior has the form

$$\pi(\beta_1|\nu_{\beta_1}, \lambda_{\beta_1}) = \frac{(\lambda_{\beta_1})^{\nu_{\beta_1}}}{\Gamma(\nu_{\beta_1})}\beta^{(\nu_{\beta_1}-1)}e^{-\lambda_{\beta_1}\beta_1}.$$

The form of the Gamma distribution we are choosing to use has $\nu > 0$ as the shape parameter, and $\lambda > 0$ as the rate parameter.

The prior distributions for β_2 and γ are also chosen to be Gamma distributions with unique hyper-parameters. In the case of γ this is also because it is conjugate and results in a conditional posterior for γ that is of a known form. We will assume a uniform prior on d bounded by the minimum and maximum distances between all pairs of individuals in the population.

The conditional posterior distributions are:

$$\pi\left(\beta_1|\beta_2, \gamma, d, \mathbf{R}, \mathbf{I}, I_\kappa, \nu_{\beta_1}, \lambda_{\beta_1}\right) \propto \left[\prod_{j \neq \kappa}^{n_I}\left(D^1_{I_{j^-}}\beta_1 + D^2_{I_{j^-}}\beta_2\right)\right] \cdot \left[\beta_1^{(\nu_{\beta_1}-1)}\right] \cdot \left[\exp\left\{-\int_{I_\kappa}^{T}\left(D^1_t\beta_1\right)dt - \lambda_{\beta_1}\beta_1\right\}\right].$$

$$\pi\left(\beta_2|\beta_1,\gamma,d,\mathbf{R},\mathbf{I},I_\kappa,\nu_{\beta_2},\lambda_{\beta_2}\right)\propto\left[\prod_{j\neq\kappa}^{n_I}\left(D^1_{I_{j-}}\beta_1+D^2_{I_{j-}}\beta_2\right)\right]\cdot\left[\beta_2^{(\nu_{\beta_2}-1)}\right]$$
$$\cdot\left[\exp\left\{-\int_{I_\kappa}^{T}\left(D^2_t\beta_2\right)dt-\lambda_{\beta_2}\beta_2\right\}\right].$$

$$\pi\left(d|\beta_1,\beta_2,\gamma,d,\mathbf{R},\mathbf{I},I_\kappa,\nu_{\beta_2},\lambda_{\beta_2}\right)\propto\left[\prod_{j\neq\kappa}^{n_I}\left(D^1_{I_{j-}}\beta_1+D^2_{I_{j-}}\beta_2\right)\right]\left[\exp\left\{-\int_{I_\kappa}^{T}\left(D^1_t\beta_1+D^2_t\beta_2\right)dt\right\}\right].$$

$$\pi\left(\gamma|\beta_1,\beta_2,d,\mathbf{R},\mathbf{I},I_\kappa,\nu_\gamma,\lambda_\gamma\right)\sim\mathsf{Gamma}\left(\left[\lambda_\gamma+\int_{I_\kappa}^{T}\left(Y_t\right)dt\right],\left[n+\nu_\gamma\right]\right).$$

$$\pi(\mathbf{I}|\mathbf{R},I_\kappa,\beta_1,\beta_2,\gamma,d,\nu_{\beta_1},\lambda_{\beta_1},\nu_{\beta_2},\lambda_{\beta_2},\nu_\gamma,\lambda_\gamma)$$
$$\propto\left[\prod_{j\neq\kappa}^{n_I}\left(D^1_{I_{j-}}\beta_1+D^2_{I_{j-}}\beta_2\right)\right]\cdot\exp\left\{-\int_{I_\kappa}^{T}\left(D^1_t\beta_1+D^2_t\beta_2\right)dt\right\}\cdot\exp\left\{-\gamma\int_{I_\kappa}^{T}\left(Y_t\right)dt\right\}.$$

2.5 Metropolis-Hastings Steps

Metropolis-Hastings steps are a useful tool for sampling from most of the posterior distributions presented in Section 2.4. They will form part of our MCMC inference schema and produce dependent samples of our parameters of interest, whilst accounting for our missing data.

In this section, we present the proposal distributions and subsequent Metropolis-Hastings acceptance probabilities for the parameters and unknown data from Section 2.4.

2.5.1 The Random Walk Metropolis-Hastings step

We will be using a common Metropolis-Hastings algorithm, the Random Walk Metropolis. The proposal distribution is a Normal distribution centred around the current value of the parameter, θ^*, with variance σ^2, where σ^2 is a tuning parameter. Our parameter draw is θ', where $\theta' \sim N(\theta^*, \sigma^2)$. The normal distribution is symmetric so the probability of going from $\theta' \to \theta^*$ is the same as going from $\theta^* \to \theta'$. Thus, the proposal distribution will cancel in the acceptance probability and will again result in Equation (1.1).

2.5.1.1 The Multiplicative Random Walk Metropolis

In some cases we found the algorithms to be more efficient when proposing on the log scale, with the added benefits of the proposed draws can only be positive like the parameters. Transforming to the log scale to propose is called a Multiplicative Random Walk Metropolis (Dellaportas and Roberts, 2003). The proposal distribution is now skewed due to the logarithmic transform, and so is no longer symmetric. We need to consider the Jacobian of the transformation, which accounts for the change in density due to the transformation, and adjust the acceptance probability accordingly.

To do this we make use of the result that for two random variables θ and ϕ, where $\phi = \log(\theta)$, the posterior $\pi(\phi|X) = \theta \cdot \pi(\theta|X)$. So the acceptance probability is given by

$$\alpha = \min\left\{\frac{\theta' \cdot \pi(\theta'|X)}{\theta^* \cdot \pi(\theta^*|X)}, \quad 1\right\}$$

2.5.1.2 The Folded Random Walk Metropolis

By taking advantage of the symmetric nature of the Random Walk proposal, we can bound the proposal distribution whilst still maintaining a symmetric distribution. This is especially useful for parameters that are strictly positive but are inefficiently

explored on the log scale, or parameters that exist within clear bounds.

First consider a Normal proposal centred on the current value of the parameter, θ^*, with variance σ^2, where σ is a tuning parameter. We introduce a lower bound, l, and an upper bound, u. We propose as normal from the proposal distribution, however, if we propose a value below l or above u, then we 'fold' the distribution back on itself using the following logic:

When a value $\phi \sim N(\theta^*, \sigma^2)$ is drawn, the following repeated fold operations are performed:

Fold operations:

Inputs: Current value of θ, θ^*; Proposed value, ϕ; Lower bound, l; Upper bound, u

```
while φ ∉ [l, u] do
    if φ < l then
        φ ← 2l − φ
    end
    else if φ > u then
        φ ← 2u − φ
    end
end
```

Algorithm 2.2: Fold operations for the Folded Random Walk.

the proposal θ' is then set to ϕ.

Thus θ' only has support in $[l, u]$ and it's density is the sum of an infinite sequence of Gaussian densities:

Denoting $g(\theta)$ as the density of a $N(\theta^*, \sigma^2)$ random variable at θ, then

$$q(\theta'|\theta) = g(\theta') + \sum_{i=1}^{\infty} \Big[g\left(2(i)l - 2(i-1)u - \theta'\right) + g\left(2(i)u - 2(i)l + \theta'\right) + g\left(2(i)u - 2(i-1)l - \theta'\right) + g\left(2(i)l - 2(i)u + \theta'\right) \Big] \quad (2.3)$$

Even if the bounds are not equidistant from the mean, the symmetry of the proposal distribution is still preserved. The reason for this is non-obvious unless we

consider the probability density function of the Gaussian distribution. We use the form below, and consider $\sigma = 1$ for simplicity:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
$$\propto e^{-\frac{(x-\mu)^2}{2}}.$$

Thus we can see that comparing two distributions with $\sigma = 1$ we need only concern ourselves with the $(x-\mu)^2$ term, and the evaluation of the two distributions will have the same value if $(x_1 - \mu_1)^2 = (x_2 - \mu_2)^2$. Now consider an example where $x = \theta$ and $\mu = m$, and the reverse move distribution where $x = m$ and $\mu = \theta$. Then $(\theta - m)^2 = (\theta^2 - m\theta + m^2) = (m - \theta)^2$, thus we've shown that the standard random walk is reversible. If we then substitute θ for any term in the infinite sum, such as $\phi = 2l - \theta$, then we can see that $(\phi - m)^2 = (2l - \theta - m)^2 = ((2l - m) - \theta)^2$. For $g\left(2(i)u - 2(i)l + \theta\right)$ the reverse term will be $g\left(2(i)l - 2(i)u + m\right)$. Thus every term in the infinite sum when $x = \theta$ and $\mu = m$ has a matching term of equal value in the infinite sum when $x = m$ and $\mu = \theta$, so $\pi(\theta|m) = \pi(m|\theta)$ and the proposal is reversible. Thus the acceptance probability is still given by Equation (1.1).

2.6 Proposal Distributions

We use a Multiplicative Random Walk for β_1 and β_2. For d we will use a Folded Random Walk. Since γ has a Gamma posterior we can sample directly from the posterior using a Gibbs sampler. It is worth noting that as the product in the posterior for β_1 (and for β_2) can be expanded to be expressed as a sum of polynomials in β_1, it is proportional to a mixture of Gamma distributions from which we could sample directly utilising a Gibbs sampler. We have chosen to explore the Multiplicative Random Walk here as it is more applicable in the later chapters.

2.6.0.1 The Infection rate parameters

The infection rate parameters, β_1 and β_2, are not independent, but their posterior distributions have identical forms.

Define the random variable $B_1 = \log(\beta_1)$. The proposal distribution for B_1 is given by $B_1' \sim N(B_1^*, \sigma_{\beta_1})$. The acceptance probability is given by

$$\alpha_{B_1} = \min\left\{\frac{\beta_1' \cdot \pi\left(\beta_1'|\beta_2, d, \mathbf{R}, \mathbf{I}, I_\kappa, \nu_{\beta_1}, \lambda_{\beta_1}\right)}{\beta_1 \cdot \pi\left(\beta_1|\beta_2, d, \mathbf{R}, \mathbf{I}, {}_\kappa, \nu_{\beta_1}, \lambda_{\beta_1}\right)}, \quad 1\right\},$$

and we can derive a similar result for β_2.

2.6.0.2 The distance

We bound d between the smallest distance between two individuals and the largest distance between two individuals. We will use a Folded Normal Random Walk to propose samples for d, using Algorithm 2.2 which gives the density in Equation (2.3) but with $l = d_{\min}$, $u = d_{\max}$, and θ, θ' being d and d'.

2.6.0.3 The infection times

The missing data, the infection times, are initialised at valid values by using the known removal times and assumptions about the underlying data generating process, and can be updated using a Metropolis-Hastings step.

We choose an infected individual uniformly on the set of infected/removed individuals, and then replace its infection time with a new one. The new infection time is generated by randomly drawing a new infectious period for that individual and subtracting it from their observed removal time.

We let Q_i be the random variable denoting the infectious period of individual i. Then the infection time of individual i, $I_i = R_i - Q_i$, so $Q_i = R_i - I_i$. In the case where individual s is chosen, the proposal distribution is given by,

$$
\begin{aligned}
q(\mathbf{I}'|\mathbf{I}) &= \frac{1}{n_I - 1} \times \gamma \exp\left\{-\gamma Q'_s\right\} \\
&= \frac{1}{n_I - 1} \times \gamma \exp\left\{-\gamma (R_s - I'_s)\right\},
\end{aligned}
$$

as we do not propose changes to the initial infection time. Then, using the fact that $\exp\left\{-\gamma \int_{I_\kappa}^{T} (Y_t)\, dt\right\}$ can be written as $\exp\left\{-\gamma \sum_{i=1}^{n_I} (R_i - I_i)\right\} = \prod_{i=1}^{n_I} \exp\left\{-\gamma (R_i - I_i)\right\}$, the acceptance probability of the Metropolis-Hastings step is given by;

$$
\begin{aligned}
\alpha_{\mathbf{I}} &= \min\left\{ \frac{\pi(\mathbf{I}'|I_\kappa, \beta_1, \beta_2, \gamma, d)}{\pi(\mathbf{I}|I_\kappa, \beta_1, \beta_2, \gamma, d)} \frac{q(\mathbf{I}|\mathbf{I}')}{q(\mathbf{I}'|\mathbf{I})}, \quad 1 \right\} \\
&= \min\left\{ \frac{\left[\prod_{j\neq\kappa}^{n_I} \left(D^1_{I'_{j-}}\beta_1 + D^2_{I'_{j-}}\beta_2\right)\right] \cdot \exp\left\{-\int_{I_\kappa}^{T} (D^1_t\beta_1 + D^2_t\beta_2)\, dt\right\} \cdot \exp\left\{-\gamma \int_{I_\kappa}^{T} (Y_t)\, dt\right\}}{\left[\prod_{j\neq\kappa}^{n_I} \left(D^1_{I_{j-}}\beta_1 + D^2_{I_{j-}}\beta_2\right)\right] \cdot \exp\left\{-\int_{I_\kappa}^{T} (D^1_t\beta_1 + D^2_t\beta_2)\, dt\right\} \cdot \exp\left\{-\gamma \int_{I_\kappa}^{T} (Y_t)\, dt\right\}} \right. \\
&\qquad \left. \frac{\cdot \frac{1}{n_I - 1} \times \gamma \exp\left\{-\gamma (R_s - I'_s)\right\}}{\cdot \frac{1}{n_I - 1} \times \gamma \exp\left\{-\gamma (R_s - I_s)\right\}}, \quad 1 \right\} \\
&= \min\left\{ \frac{\left[\prod_{j\neq\kappa}^{n_I} \left(D^1_{I'_{j-}}\beta_1 + D^2_{I'_{j-}}\beta_2\right)\right] \cdot \exp\left\{-\int_{I_\kappa}^{T} (D^1_t\beta_1 + D^2_t\beta_2)\, dt\right\} \cdot \prod_{j\neq\kappa}^{n_I} \exp\left\{-\gamma (R_i - I'_i)\right\}}{\left[\prod_{j\neq\kappa}^{n_I} \left(D^1_{I_{j-}}\beta_1 + D^2_{I_{j-}}\beta_2\right)\right] \cdot \exp\left\{-\int_{I_\kappa}^{T} (D^1_t\beta_1 + D^2_t\beta_2)\, dt\right\} \cdot \prod_{j\neq\kappa}^{n_I} \exp\left\{-\gamma (R_i - I_i)\right\}} \right. \\
&\qquad \left. \frac{\cdot \gamma \exp\left\{-\gamma (R_s - I_s)\right\}}{\cdot \gamma \exp\left\{-\gamma (R_s - I'_s)\right\}}, \quad 1 \right\}
\end{aligned}
$$

Since it is only I_s and I'_s that differ many of the terms cancel,

$$
= \min\left\{ \frac{\left[\prod_{j\neq\kappa}^{n_I} \left(D^1_{I'_{j-}}\beta_1 + D^2_{I'_{j-}}\beta_2\right)\right] \cdot \exp\left\{-\int_{I_\kappa}^{T} (D^1_t\beta_1 + D^2_t\beta_2)\, dt\right\}}{\left[\prod_{j\neq\kappa}^{n_I} \left(D^1_{I_{j-}}\beta_1 + D^2_{I_{j-}}\beta_2\right)\right] \cdot \exp\left\{-\int_{I_\kappa}^{T} (D^1_t\beta_1 + D^2_t\beta_2)\, dt\right\}}, \quad 1 \right\}.
$$

This acceptance rate is valid regardless of the number of infection times updated in one move, as the form of the proposal, q, expands with the additional changed infection times, and all the same terms cancel. If any of the proposed infection times are invalid, then the conditional posterior will be equal to 0 and the move will be rejected automatically.

2.7 MCMC Algorithm

Using the details of Sections 2.4 and 2.6 we now present an algorithm to perform MCMC inference for epidemic data under the assumptions of this Near vs Far model. We are assuming the epidemic is completed such that the total number of removed individuals is equal to the total number of infected individuals.

MCMC for the Near vs. Far GSE:

Inputs: Population size, N; Removal times, $\mathbf{R}$; Distance matrix, $\mathrm{M_d}$; Lower and upper bounds for d, $d_{\min}$ and $d_{\max}$; Number of iterations, N_{its}; Tuning parameters, $U_I, \sigma_{\beta_1}, \sigma_{\beta_2}, \sigma_d$.

1. Initialise the process by generating values for the near infection rate, β_1, the far infection rate, β_2, the removal rate, γ, the distance, d, and a valid set of infection times, $\mathbf{I}$, from their respective priors.
2. Draw a sample directly from the conditional posterior distribution for γ using a Gibbs sampler, and record the new value.
3. Use a Metropolis-Hastings step to update the infection times by;
   (a) Randomly select U_I of the infected individuals to have their infection times/periods updated.
   (b) Draw new values for the infectious periods, Q_i, of those infected individuals from the prior distribution for I, and calculate their new infection times using $I_i = R_i - Q_i$.
   (c) Calculate α_I, using the current and proposed sets of infection times, and the current values of the other parameters.
   (d) With probability α_I, accept the proposed infection times update and record the new infection times, otherwise reject and record the current infection times.
4. Use a Metropolis-Hastings step to update the near infection rate, β_1, by;
   (a) Draw a value from a $\mathrm{N}(\log(\beta_1), \sigma_{\beta_1})$ distribution and take it's exponential, this is the proposed value β_1^*.
   (b) Calculate the MH acceptance probability, α_{β_1}, using the current and proposed β_1 values, and the current values of the other parameters.
   (c) With probability α_{β_1}, accept the proposed β_1 update and record the new β_1, otherwise reject and record the current β_1.
5. Use a Metropolis-Hastings step to update the far infection rate, β_2, by;
   (a) Draw a value from a $\mathrm{N}(\log(\beta_2), \sigma_{\beta_2})$ distribution and take it's exponential, this is the proposed value β_2^*.
   (b) Calculate the MH acceptance probability, α_{β_2}, using the current and proposed β_2 values, and the current values of the other parameters.
   (c) With probability α_{β_2}, accept the proposed β_2 update and record the new β_2, otherwise reject and record the current β_2.
6. ...

Algorithm 2.3: The MCMC algorithm used to make inference for the Near vs. Far GSE.

MCMC for the Near vs. Far GSE (continued):

Inputs: Population size, N; Removal times, R; Distance matrix, $\mathrm{M_d}$; Lower and upper bounds for d, d_l and d_u; Number of iterations, N_{its}; Tuning parameters, $U_I, \sigma_{\beta_1}, \sigma_{\beta_2}, \sigma_d$.

5. ...

6. Use a Metropolis-Hastings step to update the distance, d, by;

   (a) Use a Folded Normal Random Walk to propose a sample for d, using Algorithm 2.2 which gives the density in Equation (2.3) but with $l = d_{\min}$, $u = d_{\max}$, and θ, θ' being d and d'.

   (b) Calculate the MH acceptance probability, α_d, using the current and proposed d values, and the current values of the other parameters.

   (c) With probability α_d, accept the proposed d update and record the new d, otherwise reject and record the current d.

7. Repeat steps 2-6 for N_{its} iterations, and then discard the first B samples as burn-in.

Algorithm 2.4: Continued: The MCMC algorithm used to make inference for the Near vs. Far GSE.

2.8 An Alternative Parameterisation

We wish to investigate whether alternative parameterisations have an effect on the efficiency of our inference. To do this we introduce a new parameterisation that modifies the model to a single global infection rate, β, and a scalar, $p \in [0, 1]$, of that global infection rate if the individuals are greater than distance d from each other. We will refer to this new parameterisation as parameterisation 2.

In this section we provide the details of what needs to change in the likelihood, posteriors, proposal distributions, and Metropolis-Hastings steps to make inference on this new parameterisation.

It is important to note the two parameterisations are not identical, as the priors placed on β_1 and β_2 for parameterisation 1 as it will now be called allow for $\beta_2 > \beta_1$,

however by restricting p to $[0, 1]$ we ensure that $\beta_1 = \beta \geqslant p\beta = \beta_2$.

2.8.1 Likelihood

Let $F_{i,j} = 1$ if individual $i \in \mathcal{I}_t$ is within distance d of individual $j \in \mathcal{S}_t$, and 0 otherwise. Then,

$$\beta_{i,j} = \left[F_{i,j}\beta + (1 - F_{i,j})p\beta\right].$$

If $p = 0$ then beyond distance d an individual cannot infect another, if $p = 1$ then distance has no effect on the infectious contact rate. In this parameterisation the GSE likelihood simplifies to,

$$\begin{aligned} f(\mathbf{I}, \mathbf{R}|\boldsymbol{\beta}, \gamma, I_\kappa) \propto & \left[\prod_{j \neq \kappa}^{n_I} \left(\beta \sum_{i \in \mathcal{I}_{I_{j-}}} \left[F_{i,j} + (1 - F_{i,j})p\right]\right)\right] \\ & \cdot \exp\left\{-\int_{I_\kappa}^{T} \left(\beta \sum_{i \in \mathcal{I}_t} \sum_{j \in \mathcal{S}_t} \left[F_{i,j} + (1 - F_{i,j})p\right]\right) dt\right\} \\ & \cdot \left[\prod_{i=1}^{n_R} \gamma\right] \cdot \exp\left\{-\gamma \int_{I_\kappa}^{T} Y_t \, dt\right\}, \end{aligned}$$

where,

- $\mathcal{S}_t$ is the set of susceptible individuals at time t,
- $\mathcal{I}_t$ is the set of infected individuals at time t.

2.8.2 Posterior

Since β is now a common term to both infectious contact rates, we can bring it to the front of any product, sum, or integral that it is involved in, as a common factor. This has the potential to improve the efficiency of our algorithm because, whilst we still need to use a Metropolis-Hastings step for p, we can now use a Gibbs sampler for $\beta = \beta_1$, and β_2 is easily recoverable.

The prior distribution for β is a $\mathsf{Gamma}(\nu_\beta, \lambda_\beta)$ with the form

$$\pi(\beta|\nu_{\beta_1}, \lambda_\beta) = \frac{(\lambda_\beta)^{\nu_\beta}}{\Gamma(\nu_\beta)}\beta^{(\nu_\beta-1)}e^{-\lambda_\beta\beta}.$$

We will assume a Uniform prior for p between the values of [0,1].

The conditional posterior distributions are:

$$\pi\left(\beta|p,\gamma,d,\mathbf{R},\mathbf{I},I_\kappa,\nu_\beta,\lambda_\beta\right) \sim \mathsf{Gamma}\left(\left[\lambda_\beta + \int_{I_\kappa}^{T}\left(\sum_{j\in\mathcal{S}_t}\sum_{i\in\mathcal{I}_t}\left[F_{i,j}+(1-F_{i,j})p\right]\right)dt\right], \left[n_I+\nu_\beta-1\right]\right)$$

$$\begin{aligned}\pi\left(p|\beta,d,\gamma,\mathbf{R},\mathbf{I},I_\kappa,\nu_\beta,\lambda_\beta\right)\propto & \left[\prod_{j\neq\kappa}^{n_I}\left(\sum_{i\in\mathcal{I}_{I_{j-}}}\left[F_{i,j}+(1-F_{i,j})p\right]\right)\right] \\ & \cdot\left[\exp\left\{-\beta\left[\int_{I_\kappa}^{T}\left(\sum_{j\in\mathcal{S}_t}\sum_{i\in\mathcal{I}_t}\left[F_{i,j}+(1-F_{i,j})p\right]\right)dt\right]\right\}\right]\end{aligned}$$

with the posteriors for γ, d, and the infection times having the same form as in Section 2.4 except with the updated likelihood of the new parameterisation.

2.8.3 Proposal distributions

The MH acceptance probabilities for the infection times and the distance, d, remain the same as in Section 2.6 with the appropriate posterior distributions from Section 2.8.2, and we still have a Gibbs sampler for γ. In addition, the MH step for $\beta_1 = \beta$ is no longer needed, as we can now sample directly from the conditional posterior distribution. Thus, we just need to derive the acceptance probability for the proportion, p, and we can recover β_2 from β and p.

2.8.3.1 The proportion

As we did with the distance, we will use a Folded Normal Random Walk for $p \in [0,1]$, using Algorithm 2.2 which gives the density in Equation (2.3) but with $l = 0$, $u = 1$,

and θ, θ' being p and p'.

2.8.4 MCMC

Here we present the full algorithm for making inference on this model under parameterisation 2, based on the assumptions made in this section.

MCMC for the reparameterised Near vs. Far GSE:

Inputs: Population size, N; Removal times, R; Distance matrix, $\mathrm{M_d}$; Lower and upper bounds for d, d_l and d_u; Number of iterations, N_{its}; Tuning parameters, U_I, σ_p, σ_d.

1. Initialise the process by generating values for the maximum infection rate, β, the proportion, p, the removal rate, γ, the distance, d, and a valid set of infection times, I, from their respective priors.
2. Draw a sample directly from the conditional posterior distribution for γ using a Gibbs sampler, and record the new value.
3. Draw a sample directly from the conditional posterior distribution for β using a Gibbs sampler, and record the new value.
4. Use a Metropolis-Hastings step to update the infection times by;
   (a) Randomly select U_I of the infected individuals to have their infection times/periods updated.
   (b) Draw new values for the infectious periods, Q_i, of those infected individuals from the prior distribution for I, and calculate their new infection times using $I_i = R_i - Q_i$.
   (c) Calculate α_I, using the current and proposed sets of infection times, and the current values of the other parameters.
   (d) With probability α_I, accept the proposed infection times update and record the new infection times, otherwise reject and record the current infection times.
5. Use a Metropolis-Hastings step to update the proportion, p, by;
   (a) Use a Folded Normal Random Walk to propose a sample for p, using Algorithm 2.2 which gives the density in Equation (2.3) but with $l = 0$, $u = 1$, and θ, θ' being p and p'.
   (b) Calculate the MH acceptance probability, α_p, using the current and proposed p values, and the current values of the other parameters.
   (c) With probability α_p, accept the proposed p update and record the new p, otherwise reject and record the current p.
6. ...

Algorithm 2.5: The MCMC algorithm used to make inference for the reparameterised Near vs. Far GSE.

MCMC for the reparameterised Near vs. Far GSE:

Inputs: Population size, N; Removal times, R; Distance matrix, $\mathrm{M_d}$; Lower and upper bounds for d, d_l and d_u; Number of iterations, N_{its}; Tuning parameters, $U_I, \sigma_{\beta_1}, \sigma_{\beta_2}, \sigma_d$.

5. ...

6. Use a Metropolis-Hastings step to update the distance, d, by;

   (a) Use a Folded Normal Random Walk to propose a sample for d, using Algorithm 2.2 which gives the density in Equation (2.3) but with $l = d_{\min}$, $u = d_{\max}$, and θ, θ' being d and d'.

   (b) Calculate the MH acceptance probability, α_d, using the current and proposed d values, and the current values of the other parameters.

   (c) With probability α_d, accept the proposed d update and record the new d, otherwise reject and record the current d.

7. Repeat steps 2-6 for N_{its} iterations, and then discard the first B samples as burn-in.

Algorithm 2.6: Continued: The MCMC algorithm used to make inference for the reparameterised Near vs. Far GSE.

2.9 Results

We performed inference for a simulated population of 100 individuals, with one initial infective and roughly 25% of the population infected in total. We are interested in whether accurate and useful inference of the parameters can still be attained despite the simplified spatial kernel assumptions. We are also interested in whether the parameterisation of the simplified spatial kernel has an effect on the efficiency of our inference.

2.9.1 The Simulated Dataset

We simulated a population of 100 individuals on a 2-D plane with dimensions 20 units wide by 20 units high. Each individual was uniformly generated an x and y

coordinate on the plane. In Figure 2.2 we show the position of the individuals on the plane, those individuals who were eventually infected (red), and the initial infected who was chosen at random (blue).

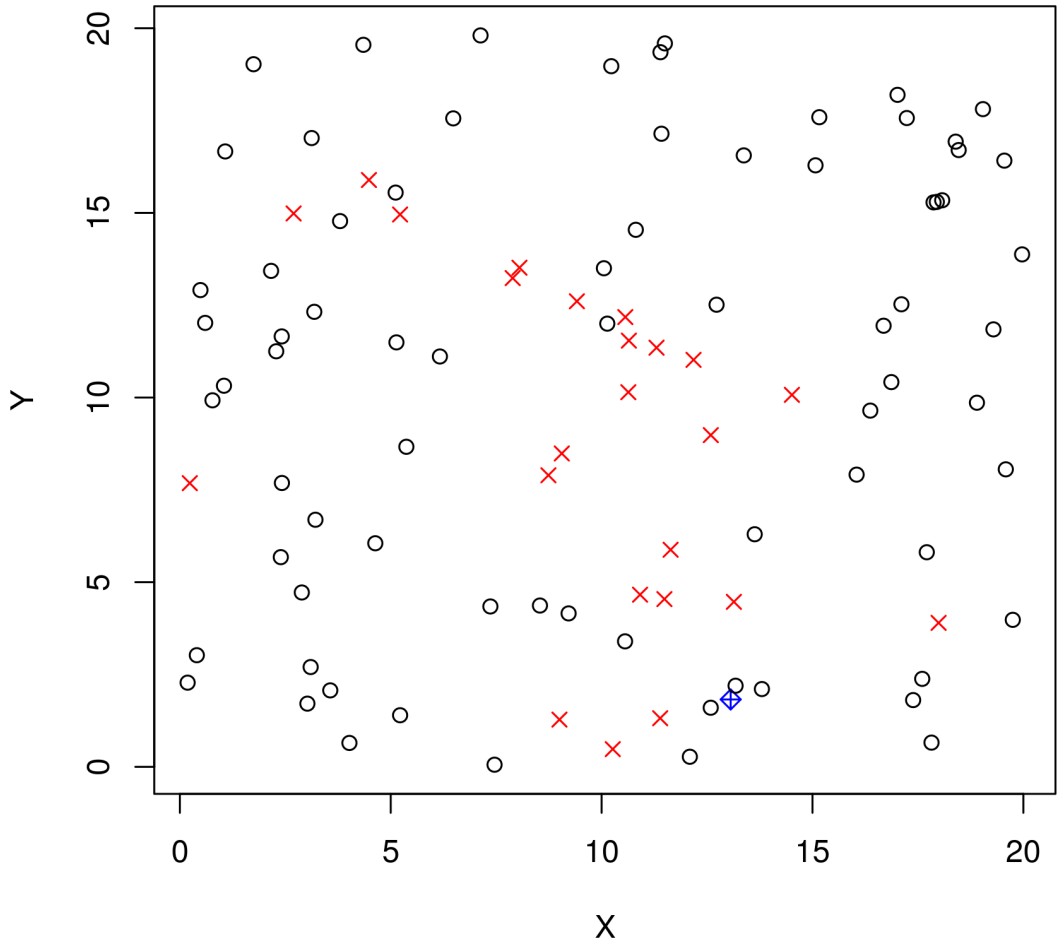


Figure 2.2: Heterogeneous simulated data set: Individuals were uniformly placed on the 20x20 plane. The initial infected (blue) was chosen at random, and the individuals infected in the course of the epidemic are denoted by red crosses.

The minimum distance between two individuals $d_{\min} = 0.078914$, and the maximum distance between two individuals $d_{\max} = 24.42934$. Table 2.1 below shows the proportions of individuals within various distances of each other, and

proportions of infected individuals within various distances of each other.

Distance, d	1	3	5	7	10	12	15	20
Total Proportion (%)	0.67	6.00	14.40	24.80	46.52	59.58	77.94	97.33
Infected Proportion (%)	2.0	13.67	33.67	46.67	71.00	84.67	95.34	100.00

Table 2.1: A table showing the proportion of individuals within distance d of each other, and the proportion of individuals who were eventually infected within distance d of each other.

2.9.2 Results Overview

We ran both algorithms for $100,000$ iterations, with the same prior distributions where possible. For parameterisation 1 we placed a $\mathsf{Gamma}(0.001, 1)$ prior on β_1, β_2, and γ, and a $\mathsf{Uniform}(d_{\min}, d_{\max})$ prior on d. For parameterisation 2 we placed a $\mathsf{Gamma}(0.001, 1)$ prior on β and γ, a $\mathsf{Uniform}(d_{\min}, d_{\max})$ prior on d, and a $\mathsf{Uniform}(0, 1)$ prior on d.

Overall the algorithms were able to recover reasonable and informative posterior distributions for all parameters, and the reparameterised heterogeneous model performed better than the original with improved mixing, higher effective sample sizes for most parameters, and was able to avoid β_1 becoming unbounded as d did not get too large or too small.

It is worth noting at this stage that we are not expecting the posterior mean to equal the true value. As epidemics are stochastic any number of parameter sets and combinations could have generated the epidemic we observed, with different probabilities. This is part of the reason why we consider the posterior distributions. Thus there may be a set of parameters that were more likely to generate our observed epidemic than the parameter set that generated it. So we are just looking for the true parameters to fall within the main mass of the posterior distribution, for the posterior to have a nice shape (as we would expect reasonably unimodal marginal posteriors for most of our parameters in this context) and reasonable variance, and whether the posterior is sufficiently distinct from the prior.

2.9.3 Parameterisation 1: Two infection parameters

This model has two distinct infection parameters, β_1 and β_2, both of which required Metropolis-Hastings steps. The utilisation of each is dependent on the distance parameter d, and they are correlated with the removal rate γ. The more parameters there are in the model, especially when those parameters are strongly correlated, the higher the potential for identifiability issues. However, in cases such as the Near vs Far model, where the extra parameters are related to additional covariate information, there is the potential that the extra information can improve inference.

The following table presents the summaries of the marginal posterior distributions:

	True Value	Mean	95% CI	Std. Dev.	ESS
β_1	0.007	0.007877	(0.00303, 0.02400)	0.006170	140.40
β_2	0.00007	0.000240479	(0.0000272, 0.0007030)	0.000183	478.31
γ	0.11	0.08942	(0.0455, 0.1540)	0.027900	779.44
d	5	4.859	(1.84, 6.20)	1.08	97.23

Table 2.2: The summary of the marginal posterior distributions for the Heterogeneous model.

We can see in Figure 2.3 that all the true values of all four parameters sit comfortably within the posterior mass, with many close to the areas of high posterior mass.

What should be noted is the threshold effect of some parameters. Because there is a limited number of individuals in the population, d can vary within ranges without significantly altering the likelihood. Also when values get to a certain size the effect on the likelihood becomes negligible, which to some extent explains the long tails of β_1 and β_2.

Finally there is a level of correlation between the parameters which means that large values in one can be accounted for by small values in others. Figure 2.4 presents the pair-wise contour plots associated with the parameters. The red dotted lines represent the true value, and the yellow dotted lines represent the pair of values of highest posterior mass in the 2-D plot.

We can see for instance that extremely large values in β_1 are accounted for by d tending to 0, which essentially means that β_1 has no effect on the likelihood and is free to take any value. Equally we can see how long the tail of β_1 extends and how that relates the values of γ and β_2. Also, Also, Figure 2.4(a) shows that even though the parameterisation did not constrain $\beta_2 < \beta_1$, the 2-D plot shows no evidence of this $\beta_2 > \beta_1$, suggesting there is sufficient information in the data to enforce this.

As for the algorithms performance; the average acceptance probability for the infection times was 39.9%. The average acceptance probability for β_1 was 46.6%. The average acceptance probability for β_2 was 46.0%. The average acceptance probability for d was 53.5%. The samples of γ were generated using a Gibbs sampler so all samples were accepted by definition.

The trace plots in Figure 2.5 show the chain of parameter draws that were accepted. The initial burn-in in orange has been discarded, and the remaining samples are assumed to have come from the stationary distribution of the chain, which is the posterior distribution of the parameter. The trace plots demonstrate the correlation of some parameters, with similar behaviour for certain parts of the chain. We can also see that the mixing for d is more sparse as the model struggles to accept new values for d and spends time stuck.

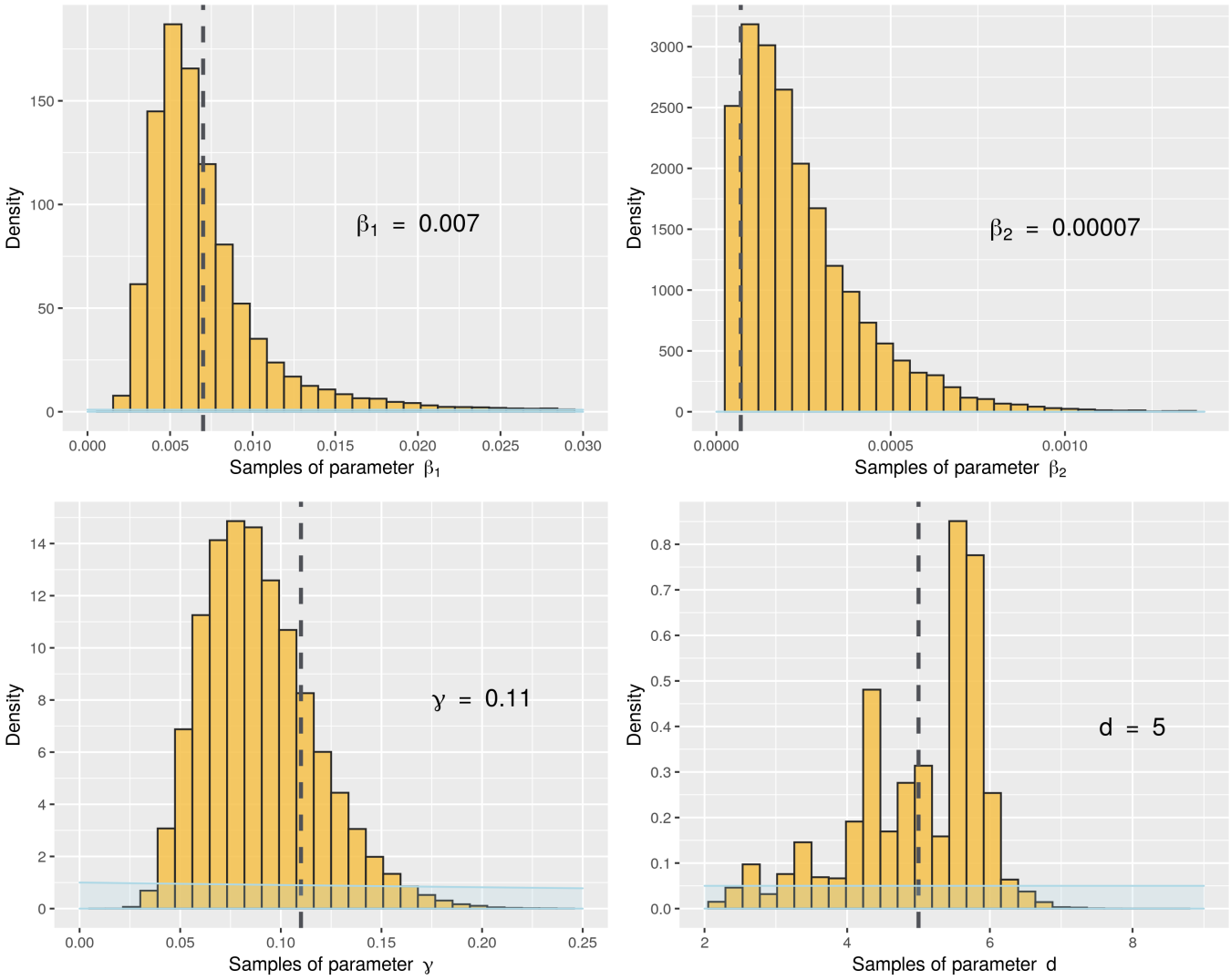
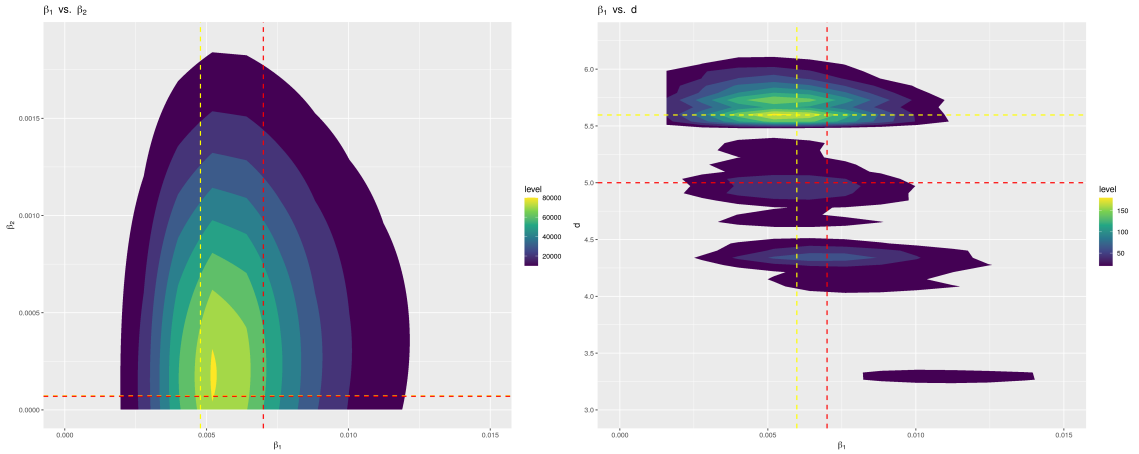
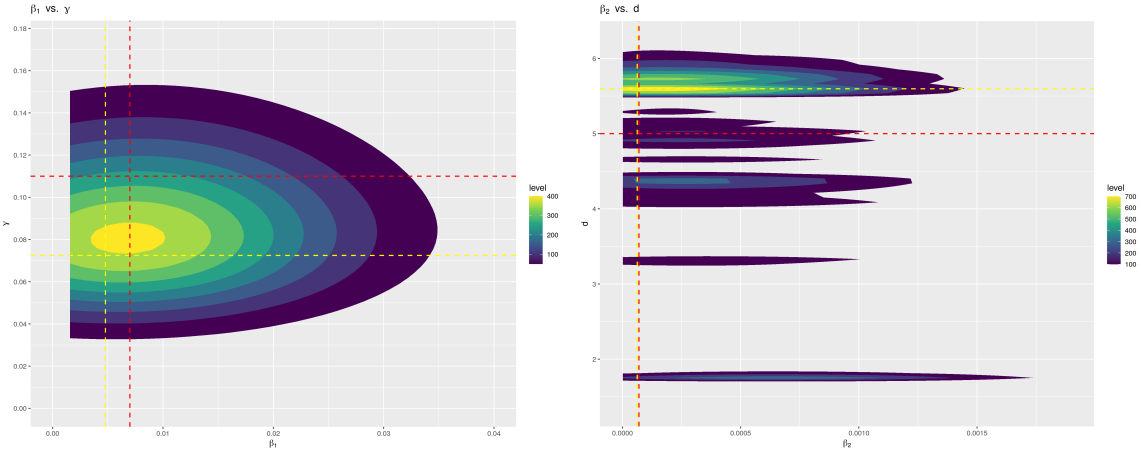


Figure 2.3: Heterogeneous Results: The plots show the marginal posterior histograms for each of the parameters of interest. The value printed on the plot is the true value of the parameter used to generate the simulation, and its location is represented by the dashed line. The prior distribution of the parameter is shown in blue.



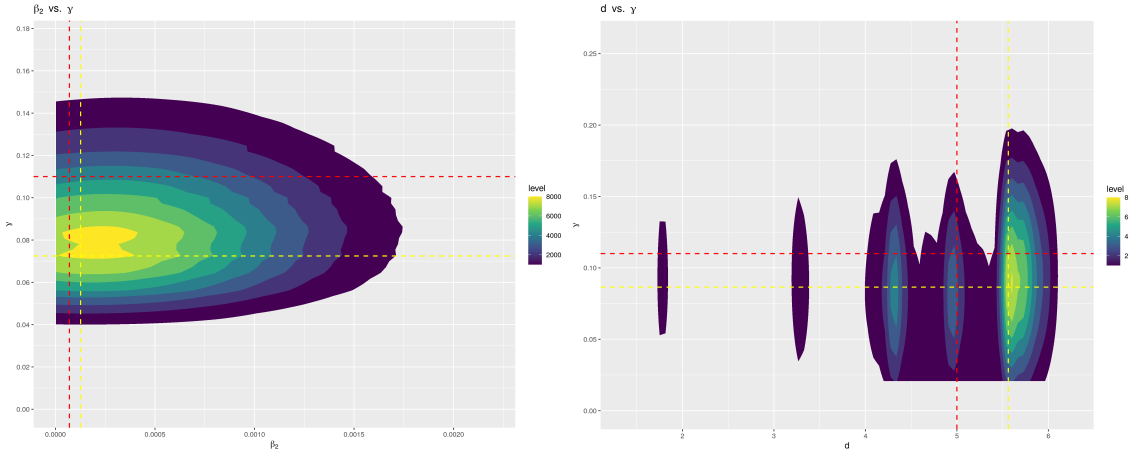
(a) β_1 against β_2.

(b) β_1 against d.



(c) β_1 against γ.

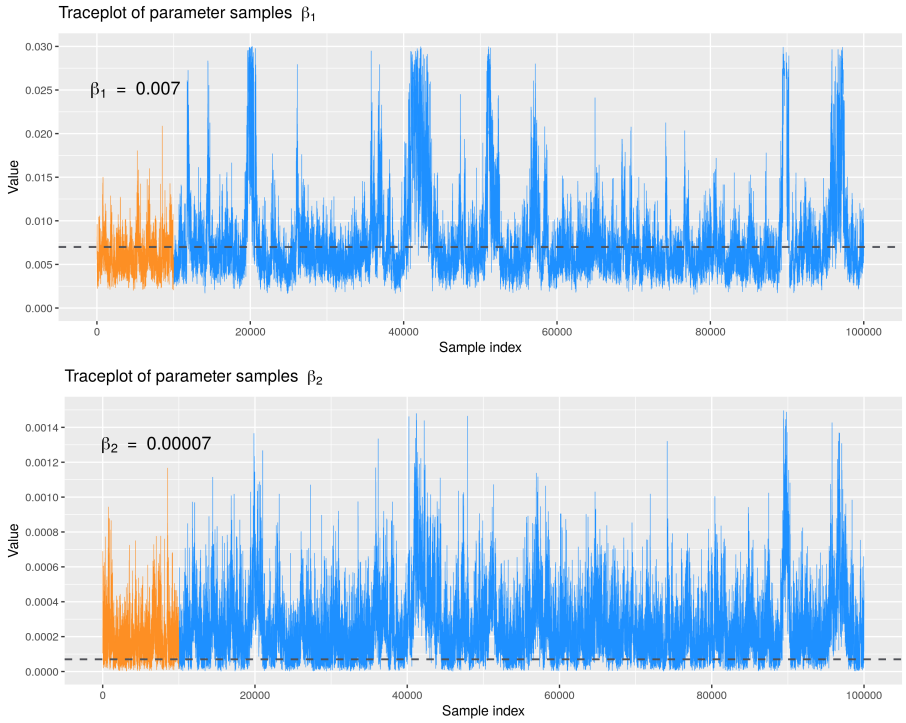
(d) β_2 against d.



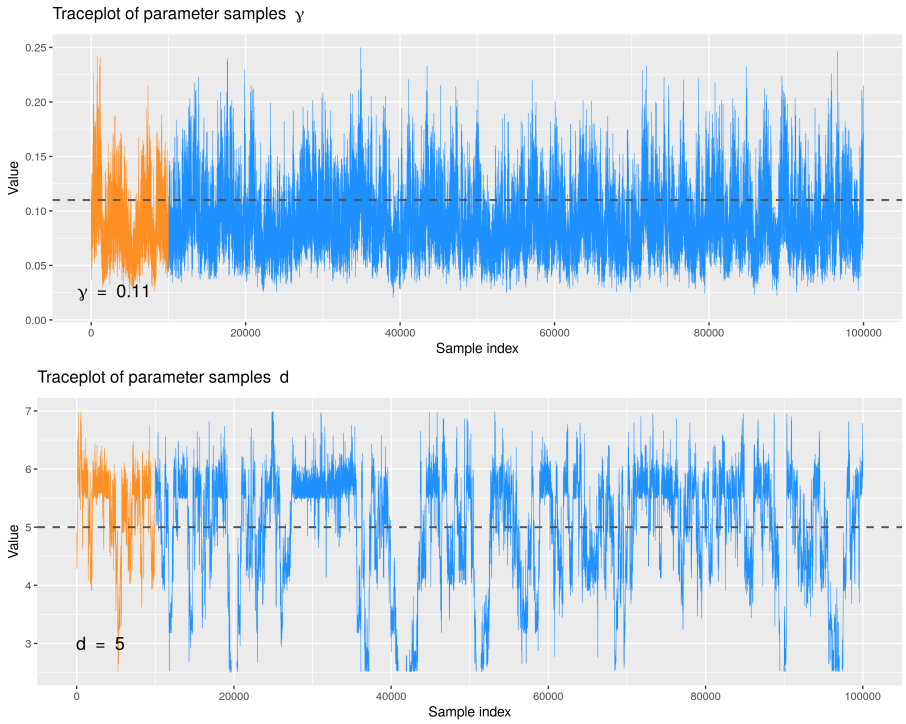
(e) β_2 against γ.

(f) d against γ.

Figure 2.4: Heterogeneous Results: Contour plots of the posterior samples for each pair of the parameters of interest. Brighter contours represent areas of higher density. The yellow dashed lines show the pair-wise point of highest density on the contour plots, and the red dashed lines represent the true values of the parameters that generated the simulation.



(a) Trace plot for β_1 and β_2.



(b) Trace plot for d and γ.

Figure 2.5: Heterogeneous Results: Trace plots of the posterior samples. The initial burn-in is represented in orange, which gets discarded. The remainder of the chain in blue is assumed to represent the stationary distribution of the chain. The true value of the parameter is shown by the dashed line.

2.9.4 Parameterisation 2: Scaled global infection rate

Here we present the inference results for parameterisation 2. Overall the parameterisation mixed better and was more efficient than the first. In the plots below we present the posterior distributions of the parameters of interest, β_1 and β_2, but we made inference on the overall infectious contact rate, β, and a scalar $p \in [0, 1]$, with $\beta_1 = \beta$ and $\beta_2 = p\beta$.

The following table presents the summaries of the marginal posterior distributions:

	True Value	Mean	95% CI	Std. Dev.	ESS
β_1	0.007	0.005823	(0.00267, 0.01140)	0.002340	692.01
β_2	0.00007	0.0001922	(0.0000229, 0.0005470)	0.000139	1496.02
γ	0.11	0.08970	(0.0457, 0.1540)	0.027900	804.44
d	5	5.328	(3.43, 6.50)	0.772	397.22

Table 2.3: The summary of the marginal posterior distributions for the Reparmeterised Heterogeneous model.

The goal of this inference was to investigate whether the reparameterisation would have a positive impact on our ability to make inference. The overall result is that yes, the reparameterisation does help. By comparing Tables 2.2 and 2.3 we can see that, with the exception of γ which remains approximately the same, the effective samples sizes for all the parameters are dramatically improved under this new parameterisation, and the credible intervals for the parameters are tighter.

From Figure 2.6 we can see that the posterior distributions produced are near identical. Comparing the trace plots presented in Figure 2.8 to those previous, we see a notable improvement. For all parameters the size of the jumps are much larger and each explores the posterior much more uniformly. For d in particular it still struggles but there is a clear improvement and it definitely gets stuck less often, with many more periods of good exploration.

Most notably by comparing the contour plots we can notice a marked improvement in the spread of the data. In the β_1 vs β_2 plot for instance, the spread of values for β_1 is clearly confined to areas of higher posterior mass. The mixing improved

such that less time was spent exploring low values of d that lead to unbounded values of β_1, which resulted in lighter tails for β_1. Additionally the 2-D areas of highest posterior mass are much closer to the truth, as demonstrated by the red dotted lines in Figure 2.7 which represent the true values, and the yellow dotted lines that represent the pair of values with the highest posterior mass.

The average acceptance probability for the infection times was 39.691%. The average acceptance probability for p was 50.681%. The average acceptance probability for d was 46.878%. In addition due to the reduced number of MH steps the code for this parameterisation runs up to 25% faster than for parameterisation 1.

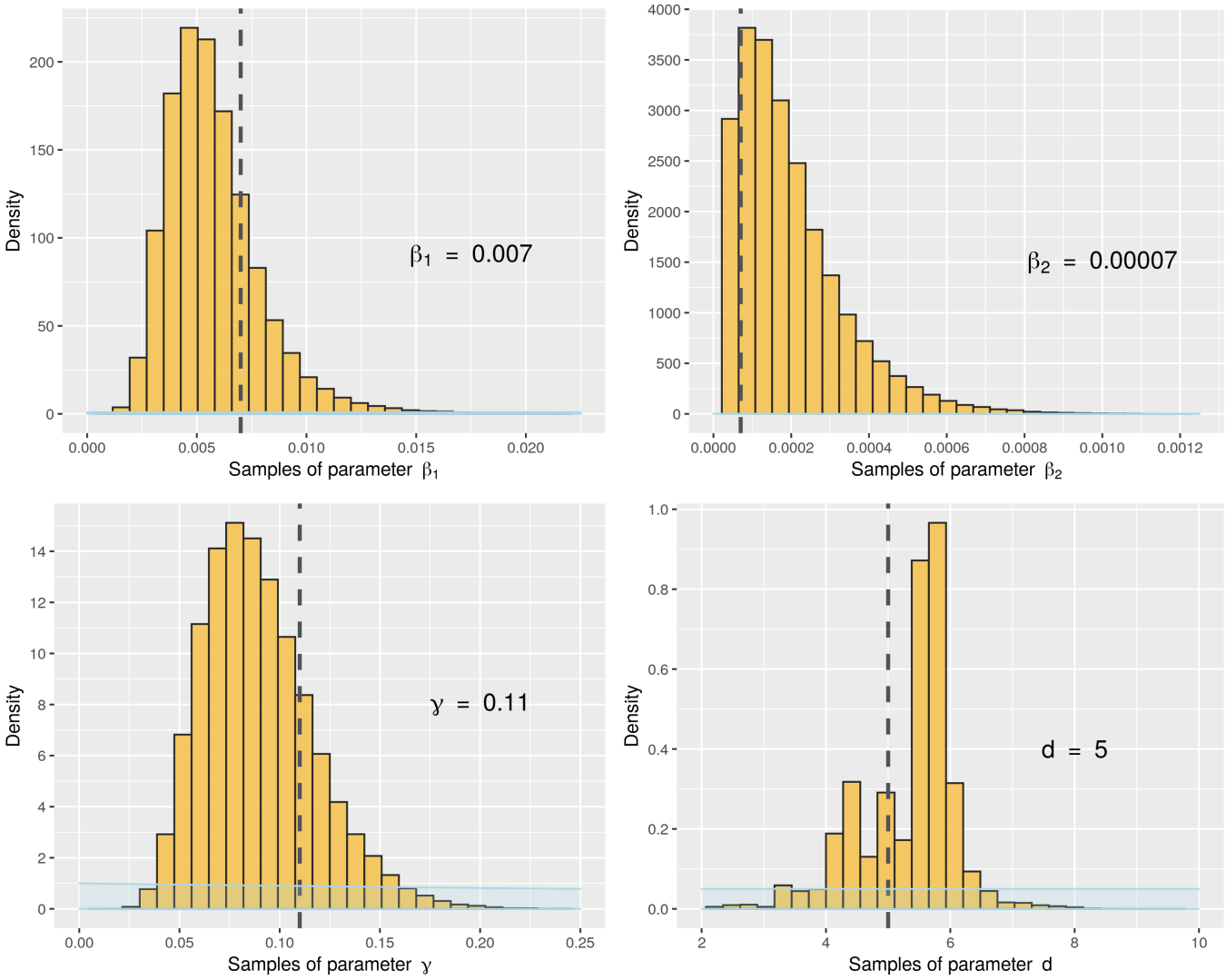
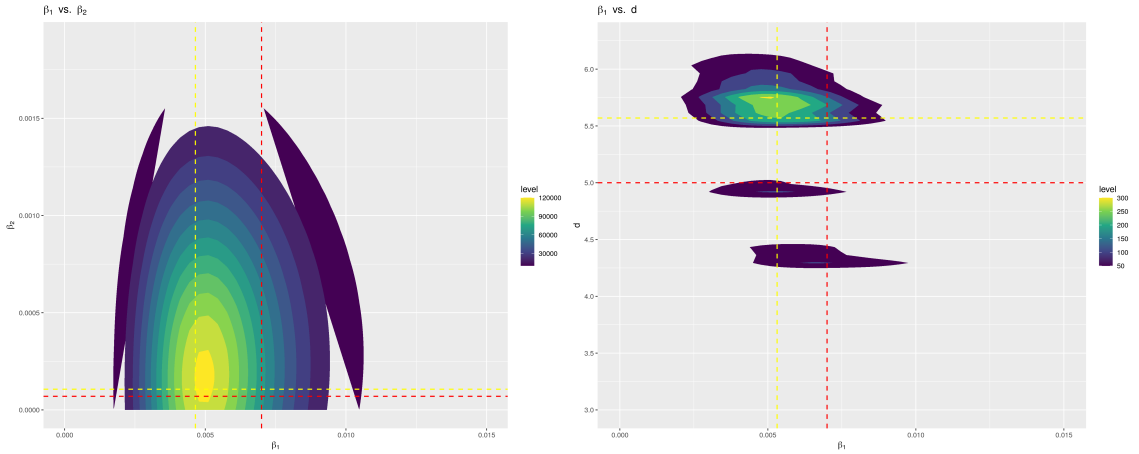
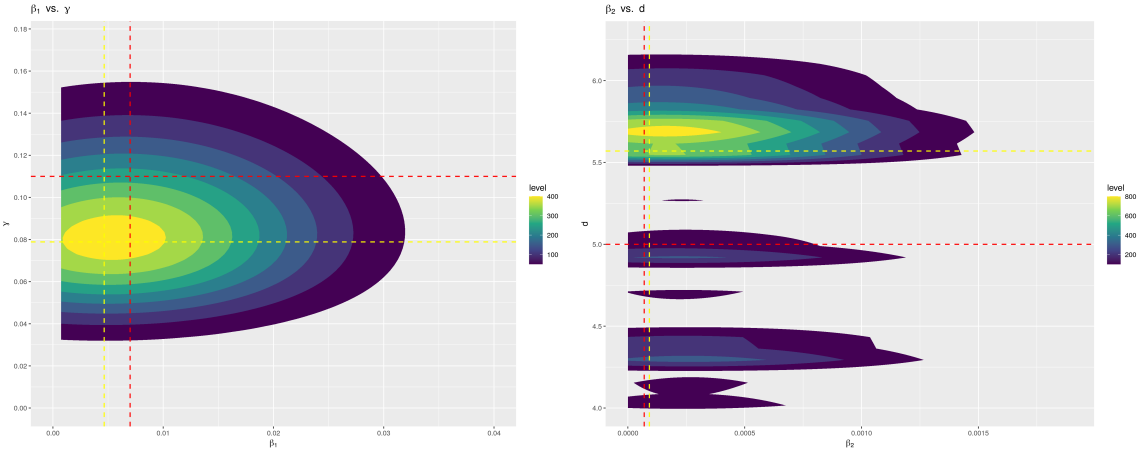


Figure 2.6: Reparameterised Heterogeneous Results: The plots show the marginal posterior histograms for each of the parameters of interest. The value printed on the plot is the true value of the parameter used to generate the simulation, and its location is represented by the dashed line. The prior distribution of the parameter is shown in blue.



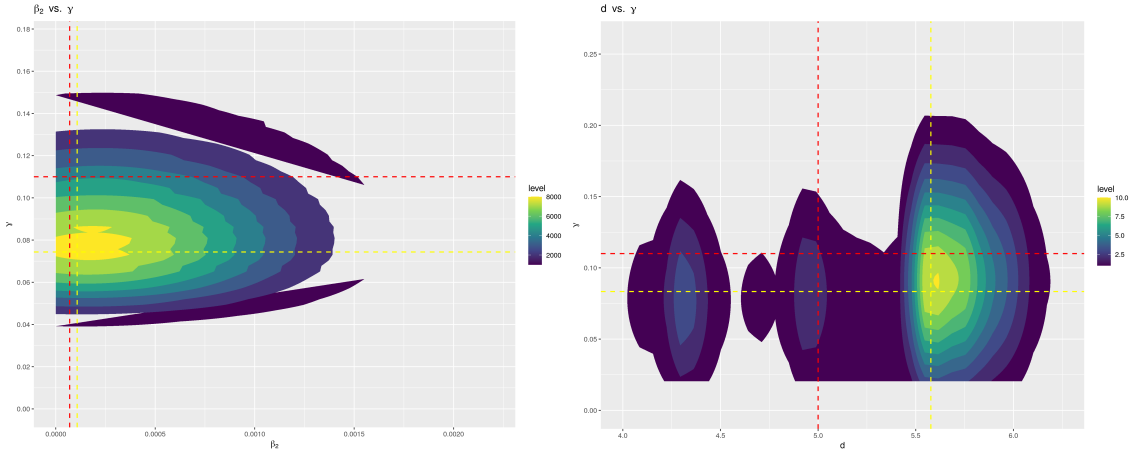
(a) β_1 against β_2.

(b) β_1 against d.



(c) β_1 against γ.

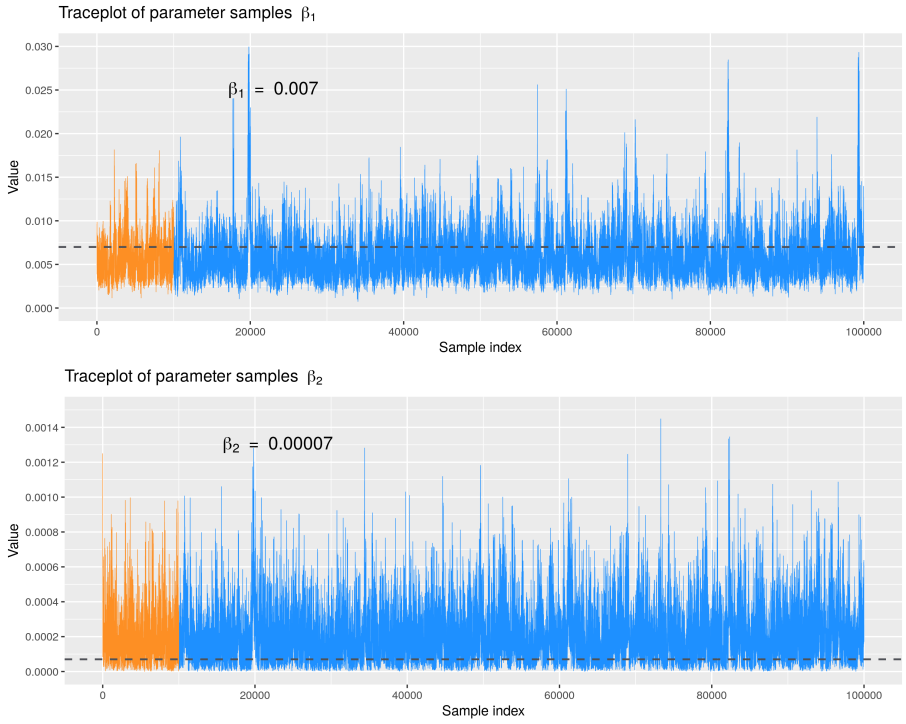
(d) β_2 against d.



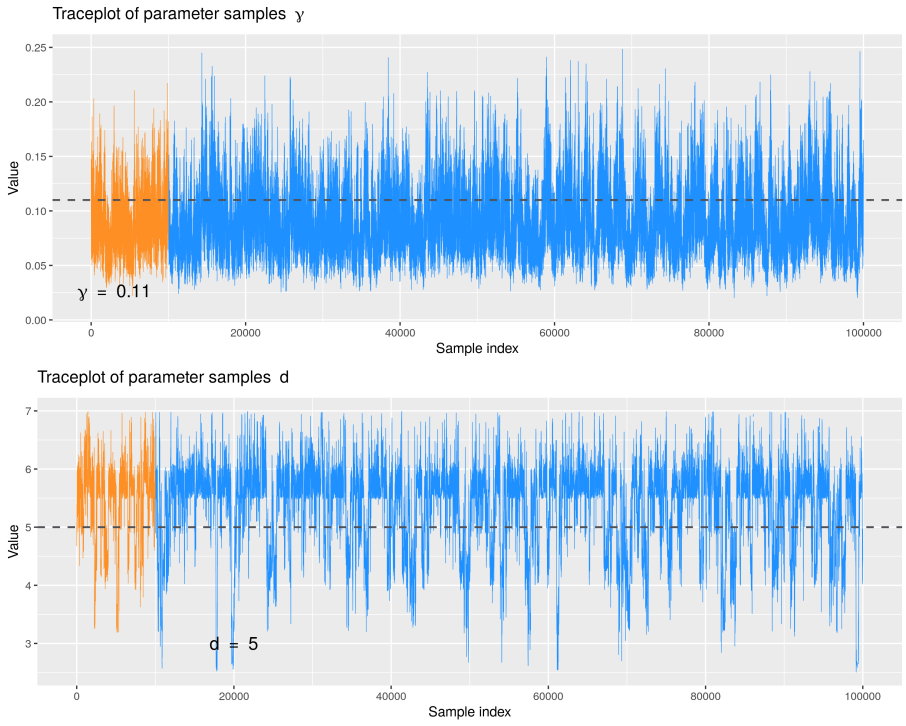
(e) β_2 against γ.

(f) d against γ.

Figure 2.7: Reparameterised Heterogeneous Results: Contour plots of the posterior samples for each pair of the parameters of interest. Brighter contours represent areas of higher density. The yellow dashed lines show the pair-wise point of highest density on the contour plots, and the red dashed lines represent the true values of the parameters that generated the simulation.



(a) Trace plot for β_1 and β_2.



(b) Trace plot for d and γ.

Figure 2.8: Reparameterised Heterogeneous Results: Trace plots of the posterior samples. The initial burn-in is represented in orange, which gets discarded. The remainder of the chain in blue is assumed to represent the stationary distribution of the chain. The true value of the parameter is shown by the dashed line.

2.10 Discussion

In this chapter we explored a simplified heterogeneous General Stochastic Epidemic, named the Near vs Far model, and methods of inference for epidemic data assumed to have been generated under this model. In an ideal scenario we would be able to make epidemic inference at the resolution of unique individuals, however that is often not possible. Models such as the Near vs Far, if they can provide accurate inference, can act as a feasible alternative for some epidemics. Our goal with this chapter was to explain the need for and theory behind the model, and assess using a simulated epidemic whether the simplified assumptions and discretisation of the spatial kernel would allow for accurate and efficient inference with reasonable computational cost. This direct supports our goals of the thesis of identifying methods to deal with the challenges of complex large data epidemics. We were also interested in exploring whether the parameterisation of the model affects its efficiency and accuracy, as this could have implications for future chapters and models.

Overall we found that the simplified Near vs Far model was a reasonable alternative to the gold standard method, returning accurate inference from a reasonably efficient algorithm. We also found that the reparameterisation did indeed improve the efficiency of the algorithm, opening this as a possible route of research for future models and challenges.

The inference did struggle however with identifiability issues of the distance term, d, which defines the threshold at which the we switch from the 'near' infectious contact rate to the 'far' infectious contact rate. This in turn had effects on the inference of the infection rates, though it was improved under the second parameterisation. This could be due to the small population size of 100 individuals with only 25 infected. This may have meant we had insufficient data on the distances between individuals to accurately determine the ideal threshold. Inference on a larger population may be more effective. Equally we could introduce stricter bounds on d, making $d_{\min}$ larger and $d_{\max}$ smaller, such that at least 1 to 3 pairs of infected individuals are affected by each parameter. This is a reasonable assumption given

we have chosen to use a Near Vs. Far model. Alternatively, we could use better proposal distributions that accounted for the fact that when d is likely misspecified by being too small or too large, the value for β_1 or β_2 become unbounded as it has no effect on the likelihood, leading to inflated tails.

However, it may simply be that for a homogeneous distributed population on a plane, this method is insufficient. If we considered a population with a more structured spatial distribution, such as clustering or latices, the model may more effectively be able to identify d.

Overall improvement is needed but we have shown that a simplified model can made accurate and efficient inference on epidemic data. However this model construction would still be too computationally intensive to do for an epidemic on the scale of tens of millions of individuals. The code for this inference was written in R, and on average took roughly 30 minutes to generate 100,000 samples for a population of 100 infected. Moving forward in this thesis we will be switching to a more efficient coding language designed for high computation costs scientific simulation, models, and processes; Julia.

Now that we have verified that discretised models have the potential to produce accurate inference, in the next section we take the discretisation of models further and look at discrete-time population-level models. These models show much more potential for being able to make inference at the scale we are interested in.

Chapter 3

Discrete approximations for State Transition Models

3.1 Introduction

The population of England is roughly 56 million people, and there were 21 million confirmed cases of COVID-19 over a 3 year period (UK Health Security Agency, 2023). In England and Wales there were roughly 22 million cattle between 2012 and 2019, and 70 million tests were performed for Bovine Tuberculosis in that period (see Chapter 4). Whilst the spatial discretisation presented in Chapter 2 may be a plausible solution for moderately sized epidemic, these big-data epidemics and pandemics make the continuous-time individual-level model infeasible, both in terms of the computational complexity of calculating the likelihood and the efficiency of the MCMC with regards to updating the missing data. If we consider the term in the continuous likelihood (Eq.(2.2)) which relates to total infectious pressure over the epidemic,

$$\int_{I_1}^{T} \left(\sum_{i \in \mathcal{I}_t} \sum_{j \in \mathcal{S}_t} \beta_{i,j} \right) dt = \sum_{i=1}^{n_I} \sum_{j=1}^{N} \beta_{i,j} [(R_i \wedge I_j) - (I_i \wedge I_j)]$$

where $\beta_{i,j}$ is the pairwise infection rate, I_i is the infection time of individual i, and R_i is the removal time of individual i, we can clearly see that the scale of these big-data challenges makes this model infeasible.

On top of this we have only considered one of the simplest model formations, the S-I-R epidemic model. One of the limitations of the S-I-R model is the biological implausibility that an individual is infectious as soon as they get infected. In this case many models introduce any number of latent states, and other infectious statuses, to better capture the behaviour of the epidemic. Examples range from Brooks-Pollock, Roberts, and Keeling, 2014 who use an Exposed latent state to represent being infected but not infectious for Bovine TB, to Overton et al., 2022 who introduce a number of different states to represent exposure, symptomatic vs non-symptomatic, hospitalisation status, recovery, and death for COVID-19. To make inference for these more complex models and big-data challenges we may also require more complex and computationally intensive MCMC methodologies to improve efficiency.

In cases such as these we can use a discrete approximation to the models presented in Chapters 1 and 2. First we can aggregate from the individual level to a population-level, and only consider the number of individuals in each state of the model at a given time point, rather than the infectious history of each individual. Then we can also discretise in time, and rather than look at the time of each event, count the number of events that occur in a window of time. These choices change the underlying distributional assumptions of the model, however, the parameters are still interpretable in the same way. As a result we arrive at an approximation for the individual level continuous-time model that has the potential to scale to the largest big-data challenges currently faced.

In this Chapter we will derive a more complex epidemic structure; the S-E-I-R, and present an advanced MCMC schema for making inference on it. We will then investigate the results of our changes using simulated data sets.

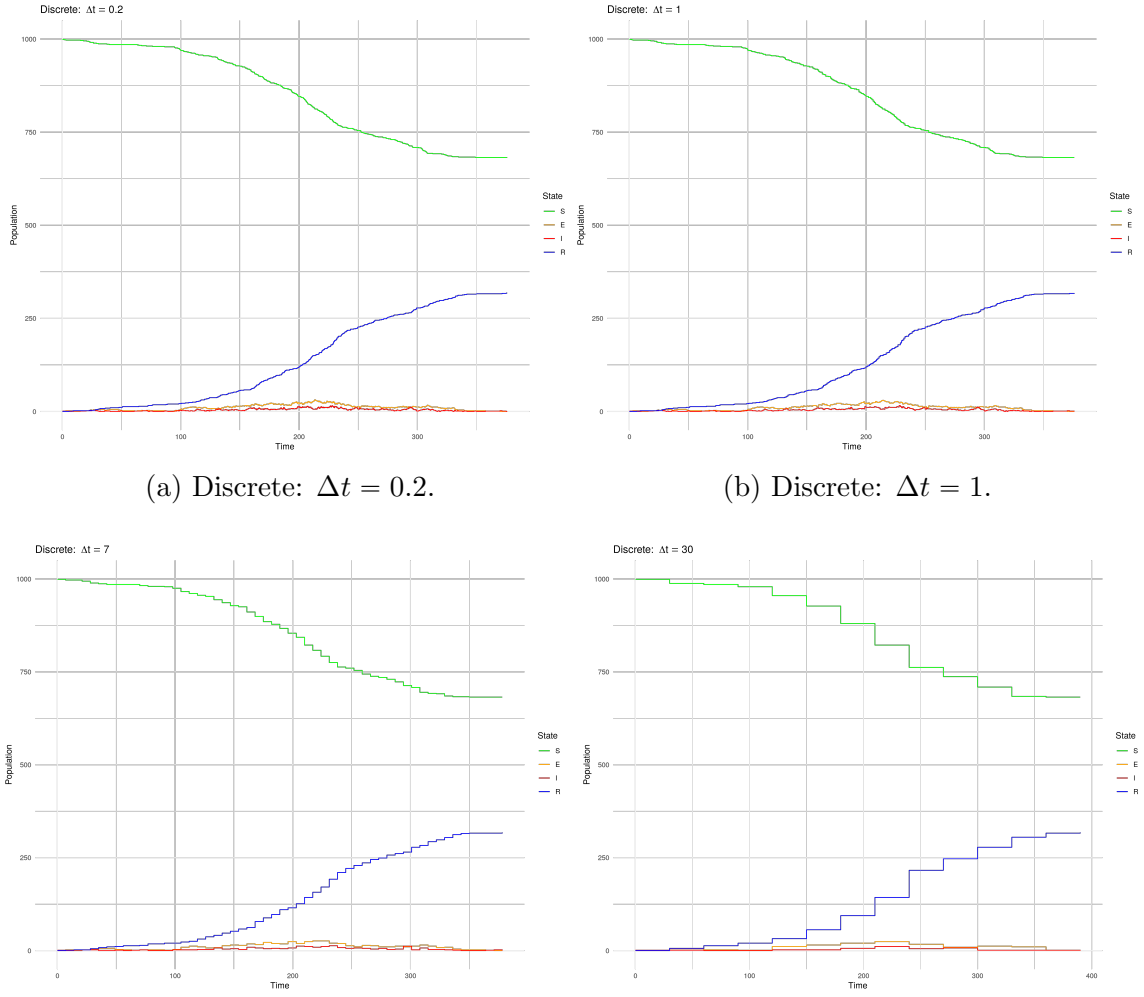
In Section 3.3 we introduce the discrete-time S-E-I-R model. In Section 3.4 we

derive the likelihood for this model, and use it in Section 3.5 to derive the posterior distributions used in the MCMC algorithms presented in Section 3.8. In Section 3.7 we introduce new MCMC proposal functions that update multiple parameters at once by taking advantage of the correlation structure of accepted samples, and also adapts the tuning parameters automatically. Finally in Section 3.9 we present the results of this inference schema on a simulated data set.

3.2 Discretising epidemic data

In contrast to the continuous-time General Stochastic epidemic, a discrete-time model makes observations of the epidemic process at discrete intervals, and counts the number of events that occurred within each interval. We call these intervals time-steps, and the size of the time-steps is chosen based on the dynamics of the disease and population in question.

If events in the epidemic occur at a relatively slow rate, say at the scale of weeks, then discretising the data into daily or possibly weekly blocks should allow us to significantly decrease the computational burden without drastically affecting the accuracy of the inference. However, if we over discretise the data then we risk losing significant information about the features of the epidemic. Figure 3.1 shows the same S-E-I-R epidemic curves under different discretisation schemes. The discrete-time model only allows each individual to make one transition during each timestep. For instance an individual could not transition from E to I and then I to R in the same timestep. This assumption puts a minimum bound on the waiting time within each state equal to the discretised timestep. As the parameters can be interpreted as the waiting time in each state, we can see that if the discretisation is too intense when compared to the dynamics of the data, the inferred parameters can be artificially inflated. For this reason the discretisation is actually an approximation of the epidemic, one that trades computational efficiency for accuracy.



(a) Discrete: $\Delta t = 0.2$.

(b) Discrete: $\Delta t = 1$.

(c) Discrete: $\Delta t = 7$.

(d) Discrete: $\Delta t = 30$.

Figure 3.1: Diagrams demonstrating the effect of different discretisation scales on a continuous-time epidemic in a population of 1000 individuals, 1 initial infected, and parameters $[\beta, \delta, \gamma] = [0.25, 0.08, 0.22]$.

3.3 The Chain-Binomial S-E-I-R

The Chain-Binomial S-E-I-R construction (Bailey, 1975, Lekone and Finkenstädt, 2006, O'Neill and Roberts, 1999) is a stochastic epidemic model for homogeneous populations. It is a discrete approximation to the continuous-time General Stochastic Epidemic model that operates in discrete time and is primarily concerned with the number of individuals in each state at given time points $t \in [1, \ldots, T]$. It

does not require knowledge of the pathways of disease transmission (who infected whom), just as the continuous-time model presented in Chapter 2.

As a result individuals are exchangeable. At each time step, the number of transitions between each valid state pair is assumed to have come from a Binomial distribution with probability defined by the state of the system. This Markov Chain of Binomial draws defines the epidemic, given some starting conditions, hence the name. This model comes with many advantages, including that it is very simple to simulate, and the likelihood is significantly easier to compute.

The S-E-I-R model introduces the latent state E, exposed, between the susceptible and infectious states. Individuals in the exposed state are infected, and will definitely transition to the infectious state after a randomly distributed amount of time, but are currently not infectious and cannot infect others.

This additional latent state addresses the implausible biological assumption of the S-I-R model that individuals are infectious as soon as they are infected. Sometimes we know this is needed because of previous clinical research, in other cases it is evident from the data, and it may be that we conclude that it must be true once attempts at fitting the 3 state S-I-R produce poor or inconsistent results. We are aiming for the simplest model that best explains the observed patterns.

In this section we present the details of the S-E-I-R model and an algorithm for simulating an epidemic in this construction.

3.3.1 S-E-I-R model specification

Consider a closed homogeneous population of N individuals. The population is divided into four independent states:

- S - Susceptibles - These individuals can be infected when they come into contact with an infectious individual.
- E - Exposed - These individuals have been infected but are not yet infectious, and exert no infectious pressure.

- I - Infectious - These individuals are infected and are capable of infecting susceptible individuals.

- R - Removed - These individuals have recovered from being infectious, are no longer able to infect others, cannot become infected again, and will remain in the removed state indefinitely.

Let $\mathcal{S}(t), \mathcal{E}(t), \mathcal{I}(t), \mathcal{R}(t)$ represent the number of individuals in the susceptible, exposed, infectious, and removed state respectively at time t. We are assuming a closed population, meaning at any given time-step, $t \in [1, \ldots, T]$, the total number of individuals in the population is equal to the sum of all the individuals in each state; $N = \mathcal{S}(t) + \mathcal{E}(t) + \mathcal{I}(t) + \mathcal{R}(t)$. There are no immigration, emigration, births, or deaths.

We initialise the epidemic with initial states $\mathcal{S}(0), \mathcal{E}(0), \mathcal{I}(0), \mathcal{R}(0)$. At each time-step we model the transitions of individuals between the states. Individuals can only make one transition per time-step, and once individuals reach the removed state, R, they remain there. Individuals can transition through the states via;

$$S \rightarrow E \rightarrow I \rightarrow R$$

The rates of transition between the states are given by;

- β - The exposure rate - The rate at which one susceptible individual becomes exposed for each infectious individual in the population, or the rate of contact between susceptible and infectious individuals.

- δ - The onset of infectiousness rate - The rate at which each exposed individual transitions to the infectious state. It controls the length of the incubation period.

- γ - The removal rate - The rate at which each infectious individual transitions to the removed state. It controls the length of the infectious period.

The model operates in discrete time and as such uses probabilities of events in a given timestep, rather than the rates of events occurring as in the continuous-time model. We can transform the continuous-time rates to the probabilities of transitioning from one state to the other in a timestep, with the size of the timestep defined as Δt, using the cumulative density function of the exponential distribution. The transition probabilities are calculated using:

- $p_{exp}(t) = 1 - \exp\left\{-\frac{\beta}{N}\mathcal{I}(t-1)\Delta t\right\}$,
- $p_{inf} = 1 - \exp\{-\delta\Delta t\}$,
- $p_{rem} = 1 - \exp\{-\gamma\Delta t\}$.

The model then states that the number of Susceptible to Exposed events during timestep t, $dS(t)$, is distributed $\text{Bin}(S(t),\ p_{exp}(t))$, the number of Exposed to Infectious events during timestep t, $dE(t)$, is distributed $\text{Bin}(E(t),\ p_{inf})$, and the number of Infectious to Removed events during timestep t, $dR(t)$, is distributed $\text{Bin}(I(t),\ p_{rem})$.

3.3.2 Simulation

We wish to simulate an SEIR epidemic in continuous time, such that we can discretise it under the assumptions of the Chain-Binomial model, and investigate the accuracy and efficiency of inference this new resolution of data, given the known true parameters. We use an extension to the homogeneous General Stochastic Epidemic simulator presented in Chapter 1, that includes the additional Exposed state. We choose a population of 1000, with 1 initial infected, and rates of 0.25 for the exposure rate β, 0.08 for the onset of infectiousness rate δ, and 0.22 for the removal rate γ. The continuous-time epidemic is then discretised at 4 resolutions; $\Delta t \in [0.2, 1, 7, 30]$. Exact discretisations of the epidemic are visualised in the epidemic curve plots in Figure 3.1. The plots show the complete epidemic. In Section 3.9 we make inference on this dataset at the four different levels of discretisation, considering an ongoing

epidemic using data from $t = 0$ to the 250^{th} removal, and comparing the accuracy and efficiency of the different discretisation levels for approximating inference of the continuous-time data.

The full data of the continuous GSE SEIR epidemic cannot be exactly discretised and match the chain Binomial formulation, because there is a chance that the same individual undergoes two transitions within one timestep. This is possible regardless of the timestep size, but the probability of it occurring tends to 1 as the size of the timestep increases. In practice to discretise continuous data, or simply change the resolution of the data, we keep the known data (removal times) fixed, and allow the other data to change to create a valid chain Binomial epidemic from which to initialise our inference. However, if desired it is possible to simulate an epidemic directly from the chain Binomial data generating process, and this provides insight into the construction of the likelihood. Algorithm 3.1 presents a method of simulating an S-E-I-R epidemic in a closed homogeneous population under the Chain-Binomial construction.

Chain-Binomial Simulation:

Inputs: Population size, N; Exposure rate, β; Onset of infection rate, δ; Removal rate, γ.

1. Initialise the number of susceptible (S), exposed (E), infected (I), and removed (R) individuals at time $t = 0$, with $\mathcal{S}(t) + \mathcal{E}(t) + \mathcal{I}(t) + \mathcal{R}(t) = N$ at all times $t \in 0, \ldots, T$. As an example, $\mathcal{S}(0) = N - 1$, $\mathcal{E}(0) = 0$, $\mathcal{I}(0) = 1$, and $\mathcal{R}(0) = 0$.

2. Then for each subsequent time-step,

   (a) Draw the number of S to E events in time-step t, $dE(t)$, using

   $$dE(t) \sim \text{Binomial}\left(\mathcal{S}(t-1), p_{exp}(t)\right),$$

   where $\mathcal{S}(t-1)$ is the number of susceptible individuals available at the start of time-step t. The probability of an S to I event is given by $p_{exp}(t) = 1 - \exp\left\{-\beta\frac{\mathcal{I}(t-1)}{N}\right\}$, where $\beta \geqslant 0$ is the exposure rate of the epidemic, and $\mathcal{I}(t-1)$ is the number of infectious individuals available at the start of time-step t.

   (b) Draw the number of E to I events in time-step t, $dI(t)$, using

   $$dI(t) \sim \text{Binomial}\left(E(t-1), p_{inf}\right),$$

   where $\mathcal{E}(t-1)$ is the number of exposed individuals available at the start of time-step t. The probability of an E to I event is given by $p_{inf}(t) = 1 - \exp\{-\delta\}$, where $\delta \geqslant 0$ is the onset of infection rate of the epidemic.

   (c) Draw the number of I to R events in time-step t, $dR(t)$, using

   $$dR(t) \sim \text{Binomial}\left(\mathcal{I}(t-1), p_{rem}\right),$$

   where $\mathcal{I}(t-1)$ is the number of infected individuals available at the start of time-step t. The probability of an I to R event is given by $p_{rem} = 1 - \exp\{-\gamma\}$, where $\gamma \geqslant 0$ is the removal rate of the epidemic.

   (d) Update the states via

   $$\begin{aligned}
   \mathcal{S}(t) &= \mathcal{S}(t-1) - dE(t), \\
   \mathcal{E}(t) &= \mathcal{E}(t-1) + dE(t) - dI(t), \\
   \mathcal{I}(t) &= \mathcal{I}(t-1) + dI(t) - dR(t), \\
   \mathcal{R}(t) &= \mathcal{R}(t-1) + dR(t),
   \end{aligned}$$

   and set $t = t + 1$.

3. Run the process for the T time-steps, or until the exposed and infectious states reaches size zero.

Algorithm 3.1: An algorithm to simulate a Chain-Binomial S-E-I-R epidemic in a closed, homogeneous population.

3.4 The likelihood of an S-E-I-R epidemic

We define our data to be the vectors (**S**, **E**, **I**, **R**) which pertain to the number of individuals in each state at each timestep $t \in [1, \ldots, T]$. We are asking, "How likely is it that we see this many individuals in each state at each timestep?". It is equally valid to consider the vector of events (**dE**, **dI**, **dR**) pertaining to the number of *new* individuals in each state at each timestep, given the initial conditions $\mathcal{S}(0)$, $\mathcal{E}(0)$, $\mathcal{I}(0)$, and $\mathcal{R}(0)$, since the process is a Markov chain.

The conditional likelihood of the number of new Exposed individuals at timestep t, $dE(t)$, is given by;

$$f(dE(t)|\mathcal{S}(t-1), \mathcal{I}(t-1), \beta) = \binom{\mathcal{S}(t-1)}{dE(t)} (p_{exp}(t))^{dE(t)} (1 - p_{exp}(t))^{\mathcal{S}(t-1)-dE(t)},$$

where $p_{exp}(t) = 1 - \exp\left\{-\beta \frac{\mathcal{I}(t-1)}{N}\right\}$ as in the simulation.

Similarly, the conditional likelihood of $dI(t)$ is given by

$$f(d\mathcal{I}(t)|\mathcal{E}(t-1), \delta) = \binom{\mathcal{E}(t-1)}{dI(t)} (p_{inf})^{dI(t)} (1 - p_{inf})^{\mathcal{E}(t-1)-dI(t)},$$

where $p_{inf} = 1 - \exp\{-\delta\}$.

And the conditional likelihood of $dR(t)$ is given by

$$f(dR(t)|\mathcal{I}(t-1), \gamma) = \binom{\mathcal{I}(t-1)}{dR(t)} (p_{rem})^{dR(t)} (1 - p_{rem})^{\mathcal{I}(t-1)-dR(t)},$$

where $p_{rem} = 1 - \exp\{-\gamma\}$.

So the joint conditional likelihood for the events at time t can be given by;

$$f(dE(t), dI(t), dR(t)|\mathcal{S}(t-1),\mathcal{E}(t-1),\mathcal{I}(t-1),\mathcal{R}(t-1),\beta,\delta,\gamma) =$$
$$\binom{\mathcal{S}(t-1)}{dE(t)}(p_{exp}(t))^{dE(t)}(1-p_{exp}(t))^{\mathcal{S}(t-1)-dE(t)}$$
$$\times \binom{\mathcal{E}(t-1)}{dI(t)}(p_{inf})^{dI(t)}(1-p_{inf})^{\mathcal{E}(t-1)-dI(t)}$$
$$\times \binom{\mathcal{I}(t-1)}{dR(t)}(p_{rem})^{dR(t)}(1-p_{rem})^{\mathcal{I}(t-1)-dR(t)}$$

Now, given this is a Markov Chain, the joint conditional likelihood of all the time-steps is just the product of the likelihoods for each time-step, given the initial conditions;

$$f(\mathbf{dE}, \mathbf{dI}, \mathbf{dR}|\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma) =$$
$$\prod_{t=1}^{T}\Bigg[\binom{\mathcal{S}(t-1)}{dE(t)}(p_{exp}(t))^{dE(t)}(1-p_{exp}(t))^{\mathcal{S}(t-1)-dE(t)}$$
$$\times \binom{\mathcal{E}(t-1)}{dI(t)}(p_{inf})^{dI(t)}(1-p_{inf})^{\mathcal{E}(t-1)-dI(t)}$$
$$\times \binom{\mathcal{I}(t-1)}{dR(t)}(p_{rem})^{dR(t)}(1-p_{rem})^{\mathcal{I}(t-1)-dR(t)}\Bigg] \qquad (3.1)$$

3.5 The posterior distributions of an S-E-I-R epidemic

The posterior distribution of a parameter is proportional to the likelihood of the data multiplied by the prior distribution of the parameter. We are interested in making inference on our three model parameters, β, δ, and γ. We assume the removal events, **dR**, to be observed, and the exposure and infection events to be unobserved. We will treat these unknown parameters as 'nuisance' parameters and augment the data with estimated plausible values via the MCMC methodology. As such we also wish

to derive posterior distributions for them as well. The only additional components we need to define in order to derive the posteriors is the prior distributions on the transmission parameters.

For β we will use a Gamma$(\phi_\beta, \sigma_\beta)$ distribution, where ϕ_β is the shape parameter and σ_β is the scale parameter. For δ we will use a Gamma$(\phi_\delta, \sigma_\delta)$ distribution, where ϕ_δ is the shape parameter and σ_δ is the scale parameter. For γ we will use a Gamma$(\phi_\gamma, \sigma_\gamma)$ distribution, where ϕ_γ is the shape parameter and σ_γ is the scale parameter. As all three parameters are positive real numbers, these priors have support for all possible values. In addition, the versatility of the Gamma distribution allows us to choose hyper-parameters that can accurately represent our prior beliefs about the parameters, whether we have strong or weak beliefs.

The joint conditional posterior of all of the parameters is thus given by:

$$
\begin{aligned}
&\pi(\beta, \delta, \gamma | (\mathcal{S},\mathcal{E},\mathcal{I},\mathcal{R}), \phi_\beta, \sigma_\beta, \phi_\delta, \sigma_\delta, \phi_\gamma, \sigma_\gamma) \propto \\
&\qquad f((\mathcal{S},\mathcal{E},\mathcal{I},\mathcal{R}) | \beta, \delta, \gamma) \times \pi(\beta) \times \pi(\delta) \times \pi(\gamma) \propto \\
&\qquad \prod_{t=1}^{T} \Bigg[\binom{\mathcal{S}(t-1)}{dE(t)} (p_{exp}(t))^{dE(t)} (1 - p_{exp}(t))^{\mathcal{S}(t-1) - dE(t)} \\
&\qquad\quad \times \binom{\mathcal{E}(t-1)}{dI(t)} (p_{inf})^{dI(t)} (1 - p_{inf})^{\mathcal{E}(t-1) - dI(t)} \\
&\qquad\quad \times \binom{\mathcal{I}(t-1)}{dR(t)} (p_{rem})^{dR(t)} (1 - p_{rem})^{\mathcal{I}(t-1) - dR(t)} \Bigg] \\
&\qquad\quad \times (\beta)^{\phi_\beta - 1} e^{-(\beta/\sigma_\beta)} \times (\delta)^{\phi_\delta - 1} e^{-(\delta/\sigma_\delta)} \times (\gamma)^{\phi_\gamma - 1} e^{-(\gamma/\sigma_\gamma)}
\end{aligned}
$$

In the proposal functions presented in Section 3.7 we explain the use of a block updater that updates all three transition parameters at once, as such we do not require the marginal distributions in this case. We explored single site updates and using reparameterisations to take advantage of conjugate priors, however in the test examples the block updater was more effective.

Similarly, since the events of time step t depend on the states at timestep $t-1$ and thus the events at $t-1$, the posterior for the number of new exposure events at each timestep, **dE** and the posterior for the number of new infectious individuals at each timestep, **dI**, is also the same as above. Depending on the form of the proposal every time step may not be required to be calculated, as we will explore in Section 3.7.2.

3.6 Adaptive Random Walk with Transformed Parameters

In the previous chapter we presented a simple Metropolis Random Walk MCMC algorithm for generating samples from the posteriors of the parameters. This can be effective, but we can improve the efficiency, and finding the optimal tuning parameters can take multiple test runs. In the case of epidemics we have shown in Chapter 2 that the parameters can be highly correlated. For instance if we increase the infection rate we can make a complimentary decrease in the removal rate and maintain a similar likelihood value. As such jumping in three random directions when exploring the parameter space is not the most efficient proposal scheme. Through the choice of more intelligent proposal distributions, we can both improve the efficiency of our MCMC algorithms, and even automate some of the tuning process which up until this point has been manual.

In the case of our Chain-Binomial S-E-I-R construction, none of the posteriors of the parameters have the form of a known distribution using the parameterisations we have chosen, so as before we will need to use a Metropolis-Hastings step to sample conditional draws from the posterior of interest. It is possible to have conjugacy by setting $p = \exp\{-\delta\}$ and $q = \exp\{-\gamma\}$, and placing Beta priors on p and q, however in our exploration we found it more efficient to propose all 3 parameters as a set, taking advantage of the correlations. From the results in Chapter 2 we saw that the parameters in the S-I-R model are highly dependent, and this holds true

for the S-E-I-R model as well (Jewell et al., 2009a). With this in mind, we have chosen a proposal distribution that allows us to use this dependence to improve the efficiency of the algorithm, by using a multi-site sampler and taking consideration of the correlation between the previously accepted samples. In addition we will be making proposals on the log-scale using a Multiplicative random walk Metropolis-Hastings, as in Chapter 2. This ensures that the proposals for the parameters will be positive, and also has the potential to improve the efficiency of the algorithm. For the examples we tested and present the results for in Section 3.9, we found that proposals on the log-scale did improve efficiency. As a result the Metropolis-Hastings acceptance probability needs to take into account this transformation. Finally, we are also able to incorporate automatic tuning of the hyperparameters of the proposal distributions, which will aim to optimise the acceptance rate by adapting the tuning parameters during the MCMC in response to its performance.

3.6.1 Adaptive MCMC

Adaptive MCMC algorithms address the challenge of finding optimal tuning parameters for a proposal distribution without the need to rerun chains (Haario, Saksman, and Tamminen, 2001). They achieve this by updating the hyperparameters of the proposal distribution during the run, based on the history of the chain so far. An issue arises in that if this process is allowed to run indefinitely, then the algorithm may become optimised for exploring a minor part of the distribution, such as a tail or minor mode, and as such become more inefficient than a manually tuned algorithm, or may even result in the chain having a different stationary distribution (Sherlock, Fearnhead, and Roberts, 2010). To address this issue the adaption either needs to either take place over a finite amount of time (e.g., only for the first 5000 iterations) or the rate of adaption needs to tend to 0, a concept called diminishing adaption (Sherlock, Fearnhead, and Roberts, 2010). Under these conditions the algorithm is still guaranteed to converge to the correct stationary distribution, though consideration should still be made for the starting conditions (Roberts and

Rosenthal, 2009).

The adaptive schema used in this thesis is based on the Adaptive Metropolis-Within-Gibbs algorithm adapted from Roberts and Rosenthal, 2009 and the Block Adaptive Multiplicative Random Walk adapted from Sherlock, Fearnhead, and Roberts, 2010. Each iteration we propose a new set of parameters from one of two possible multivariate-normal distributions centred on the log of the current parameters. The first proposal distribution does not take into account the correlation structure of the previously accepted samples, and the second does, both however are tuned automatically to optimise the acceptance rate. The tuning schema for each differs. For the first finite adaption is used, and for the second diminishing adaption is used (the adaptation rate tends to zero). During the chain, the correlated proposal is used to make the majority of the proposals, and the uncorrelated proposal is included in an attempt to improve the chains ability to explore.

3.6.1.1 The proposal function for the parameters

First let $\boldsymbol{\theta} = (\beta, \delta, \gamma)$, the array of the parameters, and let $\mathbf{A}_h$ be the array of accepted samples of all of the parameters so far. We then define Σ_h as the empirical posterior covariance matrix of the accepted samples $\mathbf{A}_h$, which we will update after each iteration, h.

We require the posterior covariance matrix to be positive definite which, due to finite sample effects, is not guaranteed until a reasonable number of sufficiently distinct samples have been accepted. In addition, we also want to avoid the covariance matrix being overly sensitive to the accepted samples before we begin fine tuning the tuning parameters.

Let the total number of samples $N_{its} = N_1 + N_2$. We have chosen to set $N_1 = 5000$ assuming the total number of iterations is sufficiently large. For the first N_1 samples, use the proposal distribution:

$$\log(\boldsymbol{\theta}') \sim \text{Multivariate-Normal}\left(\log(\boldsymbol{\theta}^*), \frac{1}{d}\lambda^2\mathbf{I}\right),$$

where $\boldsymbol{\theta}^*$ are the current values of the parameters, d is the dimension of the parameter space (in this case $d = 3$), and λ is a tuning parameter, and $\mathbf{I}$ is the $d \times d$ identity matrix, based on Sherlock, Fearnhead, and Roberts, 2010. In this case we can see that we are not taking into account the correlation between the parameters, just something akin to the average variance of the parameters.

Once we have a sufficient number of accepted samples, we can propose samples from the joint log-proposal distribution of the parameters:

$$\log(\boldsymbol{\theta}') \sim \text{Multivariate-Normal}(\log(\boldsymbol{\theta}^*), m^2\Sigma_h),$$

where $\boldsymbol{\theta}^*$ are the current values of the parameters, and m is a tuning parameter, based on Sherlock, Fearnhead, and Roberts, 2010. This proposal takes into account the correlation between previous accepted samples.

After the first N_1 samples, we can propose from either of the two distributions, as based on Sherlock, Fearnhead, and Roberts, 2010. We propose samples using

$$\log(\boldsymbol{\theta}') \sim \text{Multivariate-Normal}\left(\boldsymbol{\theta}^*, \frac{1}{d}\lambda^2\mathbf{I}\right)$$

(we call this 'Mixture 1') with probability 0.05, and propose samples using

$$\log(\boldsymbol{\theta}') \sim \text{Multivariate-Normal}(\boldsymbol{\theta}^*, m^2\Sigma_h)$$

('Mixture 2') with probability 0.95.

The idea is that we will spend the majority of our time making efficient proposals by taking into account the covariance of the samples, and will try to avoid 'getting stuck' by sometimes proposing jumps to different areas of the posterior by not taking into account the correlations. The hyperparameters λ and m are automatically updated using the process explained in Section 3.6.1.3. In theory we can initialise $\lambda = 2.38^2/d$ and $m = 2.38/d^{0.5}$, which follows directly from the optimal scaling limit results reviewed in Sherlock, Fearnhead, and Roberts, 2010. In practice we used

these are starting conditions and run a short chain to identify reasonable values from which to initialise the final algorithm.

3.6.1.2 Metropolis-Hastings acceptance probability

For this multiplicative random walk the MH acceptance probability for the parameters is defined as,

$$\alpha = \min\left\{\frac{(\beta' \cdot \delta' \cdot \gamma')}{(\beta \cdot \delta \cdot \gamma)} \frac{\pi(\beta', \delta', \gamma' | (\mathcal{S}, \mathcal{E}, \mathcal{I}, \mathcal{R}), \phi_\beta, \sigma_\beta, \phi_\delta, \sigma_\delta, \phi_\gamma, \sigma_\gamma)}{\pi(\beta, \delta, \gamma | (\mathcal{S}, \mathcal{E}, \mathcal{I}, \mathcal{R}), \phi_\beta, \sigma_\beta, \phi_\delta, \sigma_\delta, \phi_\gamma, \sigma_\gamma)}, \quad 1\right\}.$$

3.6.1.3 Adaptive tuning

In this new proposal we defined two new tuning parameters, λ and m. With each algorithm we run, we could spend time finding the optimal values of these parameters to attain our desired acceptance rate, and this would lead to the most efficient algorithm. However, epidemic modelling sometimes requires rapid solutions and spending time manually tuning algorithms isn't ideal. In these cases one possibility is to use adaptive algorithms that automatically tune the parameters, and are still very efficient given reasonable starting conditions. We present here one such adaptive tuning methodology:

We begin with Mixture 1. We use a finite adaption schema laid out in Roberts and Rosenthal, 2009. Let us define ν_k as the log-rate of adaptation of the tuning parameter λ. We let each 'batch' of 25 samples be denoted by the subscript k such that batch $k = 1$ is iterations $1, \ldots, 25$, $k = 2$ is iterations $26, \ldots, 50$, and so on. Then let ψ_k be the proportion of Metropolis-Hastings accepted samples in batch k. Then, at the start of each new batch, update λ using the formula

$$\log(\lambda) = \log(\lambda) + \nu_k$$

where,

$$\nu_k = \begin{cases} -\min\left(0.05, \frac{1}{\sqrt{k}}\right), & \text{if } \psi_k < 0.33, \\ +\min\left(0.05, \frac{1}{\sqrt{k}}\right), & \text{if } \psi_k \geqslant 0.33. \end{cases}$$

For detailed balance to be satisfied and the MCMC algorithm to target the correct posterior distribution, the tuning parameters eventually have to be fixed. In this case we choose to stop tuning λ after the first N_1 iterations and fix it at its final value. The 0.33 acceptance rate target was chosen in line with the optimal scaling for batch updaters presented in Sherlock, Fearnhead, and Roberts, 2010.

For Mixture 2, we define the rate of adaptation to be $\Delta_m = \frac{m_0}{100}$. We begin to tune m after the first N_1 iterations, and do so every iteration. The iterations are indexed by it. Each iteration, it, if the proposal came from Mixture 1, m does not get updated. Otherwise, if the proposed parameters are Metropolis-Hastings rejected, then set

$$m = m - \left(\frac{\Delta_m}{\sqrt{it}}\right)$$

and if the proposed parameters are Metropolis-Hastings accepted, then set

$$m = m + 2.3\left(\frac{\Delta_m}{\sqrt{it}}\right).$$

The forms of these update functions are chosen for acceptance rate of 30%, inline with the optimal scaling results for batch updaters with Gaussian proposals reviewed in Sherlock, Fearnhead, and Roberts, 2010.

3.7 Data Augmentation

In this section we present the posterior distributions, proposal distributions, and subsequent Metropolis-Hastings acceptance probabilities we have chosen to make inference on the discrete approximation to the S-E-I-R epidemic. For the parameters we will use the adaptive MCMC methodology as laid out in Section 3.6. For the data augmentation steps we will introduce a number of new proposal functions for

the discrete-time population-level epidemic, assuming the epidemic is still on going. Finally we explore the effect of updating the missing data on the likelihood, in order to reduce the computational burden of calculating the posterior.

3.7.1 Two Kinds of Data Augmentation

We assume that we have data on the removal events, **dR**, but that we are missing the data for the exposure events, **dE**, and the onset of infection events, **dI**. We introduce the generalised notation $d\kappa(t)$ for $\kappa \in \{E, I\}$ to represent the number of events of type κ at time t. We know that at any given time, s, that $\sum_1^s dE(s) \geqslant \sum_1^s dI(s) \geqslant \sum_1^s dR(s) - 1$, accounting for the initial infecteds recovery. That is, there can only be as many removal events by time t as there have been infection events, and there can only be as many infection events as there has been exposed events, otherwise the epidemic is invalid.

For our data augmentation, we now consider the type of update we are going to propose, instead of the event. We will consider two kinds of update. The first is what we will denote "moving an event in time"; as we know there are $\sum_1^T dR(t)$ removal events, we know that for a completed epidemic there are also that many exposure and infection events, and for an ongoing epidemic, at least that many. As such, we can 'move events around in time' by choosing an event that occurred at time t and 'moving' it forwards or backwards in time, whilst ensuring we maintain a valid epidemic.

The second type of update to the events we propose is most appropriate for epidemics which are not complete, which is "adding or removing an event". Since as we mentioned, if an epidemic is ongoing then we do not know how many exposure and infection events have actually occurred, just the minimum of each. For our update, we can propose choosing a timestep t and increasing or decreasing the number of events of each type that occur there.

We then also have the possibility of improving the efficiency of the proposals by introducing qualifiers for which timesteps can be updated each iteration, or even the

probability of choosing each timestep to be updated.

3.7.2 Posterior Distributions

We have two scenarios in which to consider updating events - the first is when the epidemic is "complete" and we know the final number of removed individuals (and as such the total number exposed and infected), and the second is when the epidemic is "ongoing" and there are still active exposed and infectious individuals in the population, and the number of individuals in these two states is unknown.

In the complete case we have knowledge of the removal events, $dR(t)$ for all t, and there are no exposed or infectious individuals left in the population at time T. As we know there are $\sum_1^T dR(t)$ removal events, we know that for a completed epidemic there are also that many exposure and infection events. We call these exposure and infection events partially observed, as we know they must have happened, we just don't know when. This means that once we have a valid epidemic, we can't augment it by adding new events or removing existing events, we can only move the events around in time.

In the ongoing case, we still have knowledge of the removal events, $dR(t)$ for all t, but we assume there are still exposed or infectious individuals left in the population at time T. For each removal event, we know there must also be a partially observed exposed and infection event (except for the initial infective), but also additional occult (unobserved) exposure and infection events, the only limit on which is the total population size. As such we can augment the data by adding or removing exposed and infectious events, as long as there are enough removal events at each time-step.

The number of individuals in each state at a given time-step t, and how these relate to the events, are expressed as:

$$\begin{aligned}
\mathcal{S}(t) &= \mathcal{S}(t-1) - dE(t), \\
\implies dE(t) &= \mathcal{S}(t-1) - \mathcal{S}(t),
\end{aligned}$$

$$\begin{aligned}
\mathcal{E}(t) &= \mathcal{E}(t-1) + dE(t) - dI(t), \\
&= \mathcal{E}(t-1) + \mathcal{S}(t-1) - \mathcal{S}(t) - dI(t), \\
\implies dI(t) &= \mathcal{E}(t-1) - \mathcal{E}(t) + \mathcal{S}(t-1) - \mathcal{S}(t),
\end{aligned}$$

$$\begin{aligned}
\mathcal{I}(t) &= \mathcal{I}(t-1) + dI(t) - dR(t), \\
&= \mathcal{I}(t-1) + \mathcal{E}(t-1) - \mathcal{E}(t) + \mathcal{S}(t-1) - \mathcal{S}(t) - dR(t), \\
\implies dR(t) &= \mathcal{I}(t-1) - \mathcal{I}(t) + \mathcal{E}(t-1) - \mathcal{E}(t) + \mathcal{S}(t-1) - \mathcal{S}(t),
\end{aligned}$$

$$\begin{aligned}
\mathcal{R}(t) &= \mathcal{R}(t-1) + dR(t), \\
&= \mathcal{R}(t-1) + \mathcal{I}(t-1) - \mathcal{I}(t) + \mathcal{E}(t-1) - \mathcal{E}(t) + \mathcal{S}(t-1) - \mathcal{S}(t),
\end{aligned}$$

and recall that $dE(t) \sim \text{Binomial}(S(t-1), p_{\text{exp}}(t))$, $dI(t) \sim \text{Binomial}(E(t-1), p_{\text{inf}})$, and $dR(t) \sim \text{Binomial}(I(t-1), p_{\text{rem}})$.

To find the marginal conditional posterior likelihood of a set of events we absorb those elements that the events do not depend on into the proportion sign. The question then is which states and other events does each event depend on. In the following sections we present a series of worked examples that showcase which states and timesteps are affected when events are moved, added, or removed, and as such which likelihood terms need to be computed. Following this we present the details of the proposal distributions we utilise in our MCMC algorithm.

3.7.2.1 The complete case - moving events in time

In this case, we are moving events in time. This means that the total number of events remains the same, but we take an event that occurs at time t and "move" it to time $t+k$. Assuming we move one event at a time, this can also be thought of as adding or subtracting that event as appropriate from all the states from t to $t+k$.

Lets consider the following farm A around time t.

A	Start of timestep				End of timestep			
	$\mathcal{S}$	$\mathcal{E}$	$\mathcal{I}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{E}$	$\mathcal{I}$	$\mathcal{R}$
$t-1$	5	4	2	1	5	4	2	1
t	5	4	2	1	5	3	3	1
$t+1$	5	3	3	1	4	3	4	1
$t+2$	4	3	4	1	3	4	4	1

The events as they currently stand are;

- $(t-1)$: Nothing
- (t) : One $E \rightarrow I$ transition
- $(t+1)$: One $S \rightarrow E$ transition and One $E \rightarrow I$ transition
- $(t+2)$: One $S \rightarrow E$ transition

Beginning with the S to E transition events, the following table shows what happens in this example when we move an S to E transition event back in time from $u = t+2$ to time $r = t$.

A	Start of timestep				End of timestep			
	$\mathcal{S}$	$\mathcal{E}$	$\mathcal{I}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{E}$	$\mathcal{I}$	$\mathcal{R}$
$t-1$	5	4	2	1	5	4	2	1
t	5	4	2	1	4	4	3	1
$t+1$	4	4	3	1	3	4	4	1
$t+2$	3	4	4	1	3	4	4	1

We can clearly see here that when moving an S to E transition event from time $t+2$ to t the only states that are affected are $\mathcal{S}(t), \mathcal{S}(t+1), \mathcal{E}(t)$, and $\mathcal{E}(t+1)$, and

the events dependent on those are the S to E and E to I transition events for times $t+1$ and $t+2$, so the joint conditional posterior likelihood for the S to E transition events is given by,

$$\pi(\mathbf{dE}|\mathbf{dI},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\propto \prod_{t=(r+1)}^{u}\left[\binom{\mathcal{S}(t-1)}{dE(t)}(p_{exp}(t))^{dE(t)}(1-p_{exp}(t))^{\mathcal{S}(t-1)-dE(t)} \times \binom{\mathcal{E}(t-1)}{dI(t)}(p_{inf})^{dI(t)}(1-p_{inf})^{\mathcal{E}(t-1)-dI(t)}\right]$$

Similarly for the E to I transition events, moving the E to I transition event from time $r=t+1$ to time $u=t+2$ (ie. $k=1$), we get

A	Start of timestep				End of timestep			
	$\mathcal{S}$	$\mathcal{E}$	$\mathcal{I}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{E}$	$\mathcal{I}$	$\mathcal{R}$
$t-1$	5	4	2	1	5	4	2	1
t	5	4	2	1	5	3	3	1
$t+1$	5	3	3	1	4	4	3	1
$t+2$	4	4	3	1	3	4	4	1

We can clearly see here that when moving an E to I transition event from time $t+1$ to $t+2$ the only states that are affected are $\mathcal{E}(t+1)$ and $\mathcal{I}(t+1)$, and the events dependent on those are the S to E transition and E to I transition events for time $t+2$. Recall however that the S to E transition events at time $t+2$, $dE(t+2)$, are also dependent on $\mathcal{I}(t+1)$ through $p_{\text{inf}}(t+2)=1-\exp\left\{-\beta\frac{\mathcal{I}(t+1)}{N}\right\}$, so the joint conditional posterior likelihood for the E to I transition events is given by,

$$\pi(\mathbf{dI}|\mathbf{dE},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\propto$$
$$\prod_{t=(r+1)}^{u}\left[\binom{\mathcal{S}(t-1)}{dE(t)}(p_{exp}(t))^{dE(t)}(1-p_{exp}(t))^{\mathcal{S}(t-1)-dE(t)}\right.$$
$$\times\binom{\mathcal{E}(t-1)}{dI(t)}(p_{inf})^{dI(t)}(1-p_{inf})^{\mathcal{E}(t-1)-dI(t)}$$
$$\left.\times\binom{\mathcal{I}(t-1)}{dR(t)}(p_{rem})^{dR(t)}(1-p_{rem})^{\mathcal{I}(t-1)-dR(t)}\right]$$

3.7.2.2 The ongoing case - adding and removing events

From the sequence of relationships above we can see that increasing or decreasing $dE(t+1)$ (to increase or decrease the total number of exposed events) will change $\mathcal{S}(t+1)$, and a change to $\mathcal{S}(t+1)$ leads to a change in $\mathcal{E}(t+1)$. A change in $\mathcal{E}(t+1)$ will lead to a change in the p.m.f of $\mathcal{I}(t+1)$ (as the new infection events will be generated from a different number of exposed individuals), which would lead to a change in the p.m.f. of $\mathcal{R}(t+1)$. As we can see from Eq. 3.1, all of the events after $t+1$ depend on the states at $t+1$ and after, and so the joint conditional posterior likelihood of the S to E transition events, assuming an event is added or removed at time k, will be given by:

$$\pi(\mathbf{dE}|\mathbf{dI},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\propto$$
$$\prod_{t=k}^{T}\left[\binom{\mathcal{S}(t-1)}{dE(t)}(p_{exp}(t))^{dE(t)}(1-p_{exp}(t))^{\mathcal{S}(t-1)-dE(t)}\right.$$
$$\times\binom{\mathcal{E}(t-1)}{dI(t)}(p_{inf})^{dI(t)}(1-p_{inf})^{\mathcal{E}(t-1)-dI(t)}$$
$$\left.\times\binom{\mathcal{I}(t-1)}{dR(t)}(p_{rem})^{dR(t)}(1-p_{rem})^{\mathcal{I}(t-1)-dR(t)}\right]$$

Also, from the sequence of relationships above we can see that increasing or

decreasing $dI(t+1)$ (to increase or decrease the total number of infectious events) will change $\mathcal{E}(t+1)$ and $\mathcal{I}(t+1)$, and a change to $\mathcal{I}(t+1)$ leads to a change in the p.m.f of $\mathcal{R}(t+1)$. All E to I and I to R transition events after $t+1$ depend on the $\mathcal{E}, \mathcal{I}, \mathcal{R}$ states at $t+1$ and after. Note also however that $dE(t+1) \sim \text{Binomial}(S(t+1), p_{\text{inf}}(t+1))$ where $p_{\text{inf}}(t+1) = 1-\exp\left\{-\beta\frac{\mathcal{I}(t)}{N}\right\}$, so all the S to E transition events after $t+1$ also depend on the $\mathcal{I}$ state after $t+1$. So again the joint conditional posterior likelihood of the E to I transition events, assuming an event is added or removed at time k, will be given by:

$$\begin{aligned}\pi(\mathbf{dI}|\mathbf{dE}, \mathbf{dR}, \mathcal{S}(0), &\mathcal{E}(0), \mathcal{I}(0), \mathcal{R}(0), \beta, \delta, \gamma) \propto \\ &\prod_{t=k}^{T}\Bigg[\binom{\mathcal{S}(t-1)}{dE(t)}(p_{exp}(t))^{dE(t)}(1-p_{exp}(t))^{\mathcal{S}(t-1)-dE(t)} \\ &\quad\times\binom{\mathcal{E}(t-1)}{dI(t)}(p_{inf})^{dI(t)}(1-p_{inf})^{\mathcal{E}(t-1)-dI(t)} \\ &\quad\times\binom{\mathcal{I}(t-1)}{dR(t)}(p_{rem})^{dR(t)}(1-p_{rem})^{\mathcal{I}(t-1)-dR(t)}\Bigg]\end{aligned}$$

3.7.3 Proposal Functions and Metropolis-Hastings Acceptance Probabilities

In this section we formalise the ideas of moving, adding, and removing events, and present proposal distributions for augmenting the epidemic data in this fashion.

3.7.3.1 "Moving an event in time" update

Begin by considering one event type, κ, from the set $\{E, I\}$. Let Δ be the magnitude and direction of the move in t, and draw Δ with equal probability from the set {-1, 1} to represent a movement backwards or forwards in time.

We could then choose t arbitrarily, but this is likely to waste compute time, for instance when we choose a timestep that does not contain any events. This is more

likely when the magnitude of the discretisation timestep is smaller. For this reason we can introduce qualifiers that increase the likelihood of proposing a new valid state of the epidemic, such as ensuring there are events to move at a given timepoint. In addition, when the number of events at a timepoint is small, it is more likely that altering these events will result in an invalid epidemic. These are typically towards the beginning and end of a completed epidemic. For this reason, we can also weight each timestep by the proportion of events of type κ it contains. Finally, we do not allow movements to before timestep 1 or to after timestep T.

If $\Delta > 0$, choose a t such that $d\kappa(t) > 0$, for $t \in [1, (T-1)]$, and if $\Delta < 0$, choose a t such that $d\kappa(t) > 0$, for $t \in [2, T]$. Weight the probability that an event at t is chosen to be moved by the number of events, $d\kappa(t)$. For now, we will move only one event at a time. So, for instance, if $d\kappa(t) = 5$, and $\Delta = -1$, then $d\kappa'(t) = 4$, and $d\kappa'(t-1) = d\kappa(t-1) + 1$. The vector of event counts at each timestep can be represented as $\mathbf{d}\kappa$ for the current set and $\mathbf{d}\kappa'$ for the proposed set. The values Δ can take, and the number of events moved, are both tuning parameters. The proposal distribution when $\Delta = 1$ is thus given by;

$$q(\mathbf{d}\kappa'|\mathbf{d}\kappa, \Delta = 1) = \frac{1}{2} \cdot \frac{d\kappa(t)}{\sum_{s\in[1,\dots,T-1]} \{d\kappa(s)\}},$$

and

$$q(\mathbf{d}\kappa|\mathbf{d}\kappa', \Delta = -1) = \frac{1}{2} \cdot \frac{d\kappa'(t+1)}{\sum_{s\in[2,\dots,T]} \{d\kappa'(s)\}},$$

where the $1/2$ represents having chosen to move events forward, and $\sum_{s\in[2,\dots,T]} \{d\kappa'(s)\}$ is the total number of events of type κ after proposing the move, and does not equal $\sum_{s\in[1,\dots,T-1]} \{d\kappa(s)\}$.

The proposal distribution when $\Delta = -1$ is thus given by;

$$q(\mathbf{d}\kappa'|\mathbf{d}\kappa, \Delta = -1) = \frac{1}{2} \cdot \frac{d\kappa(t)}{\sum_{s\in[2,\dots,T]} \{d\kappa(s)\}},$$

and

$$q(\mathbf{d}\kappa|\mathbf{d}\kappa', \Delta = 1) = \frac{1}{2} \cdot \frac{d\kappa'(t-1)}{\sum_{s\in[1,\ldots,T-1]}\{d\kappa'(s)\}},$$

where the $1/2$ represents having chosen to move events forward, and $\sum_{s\in[1,\ldots,T-1]}\{d\kappa'(s)\}$ is the total number of events of type κ after proposing the move, and does not equal $\sum_{s\in[2,\ldots,T]}\{d\kappa(s)\}$.

3.7.3.2 "Moving an event in time" Metropolis-Hastings acceptance probabilities

For the exposure events, the MH acceptance probabilities will be given by

$$\begin{aligned}\alpha &= \min\left\{\frac{\pi(\mathbf{dE}'|\mathbf{dI},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)}{\pi(\mathbf{dE}|\mathbf{dI},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)}\frac{q_u(\mathbf{dE}\mid\mathbf{dE}')}{q_u(\mathbf{dE}'\mid\mathbf{dE})}, \quad 1\right\} \\ &= \min\left\{\frac{\prod_{t=r}^{u}\left[\pi(dE_t', dI_t|dR_t,\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\right]}{\prod_{t=r}^{u}\left[\pi(dE_t, dI_t|dR_t,\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\right]}\frac{q(\mathbf{dE}\mid\mathbf{dE}')}{q(\mathbf{dE}'\mid\mathbf{dE})}, \quad 1\right\},\end{aligned}$$

where the product is only over those timesteps that are affected by the move, between timesteps r and u, and for those parts of the likelihood that are affected as detailed in Section 3.7.2.1. The $q(\cdot)$ represent the proposal distributions for moving an exposure event in time.

For the infection events, the MH acceptance probabilities will be given by

$$\begin{aligned}\alpha &= \min\left\{\frac{\pi(\mathbf{dI}'|\mathbf{dE},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)}{\pi(\mathbf{dI}|\mathbf{dE},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)}\frac{q_u(\mathbf{dI}\mid\mathbf{dI}')}{q_u(\mathbf{dI}'\mid\mathbf{dI})}, \quad 1\right\} \\ &= \min\left\{\frac{\prod_{t=r}^{u}\left[\pi(dE_t, dI_t', dR_t|\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\right]}{\prod_{t=r}^{u}\left[\pi(dE_t, dI_t, dR_t|\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\right]}\frac{q(\mathbf{dI}\mid\mathbf{dI}')}{q(\mathbf{dI}'\mid\mathbf{dI})}, \quad 1\right\},\end{aligned}$$

where the product is only over those timesteps that are affected by the move, and for those parts of the likelihood that are affected as detailed in Section 3.7.2.1. The

$q(\cdot)$ represent the proposal distributions for moving an infection event in time.

3.7.3.3 "Adding or Removing an event" update

In each iteration we only choose to either add or remove an event. Choose an event type κ, from the set $\{E, I\}$. Let Δ here be the change in the number of events, and sample Δ from the set $\{-1, 1\}$ to represent a removal or an addition respectively. Let $p_\kappa(t)$ represent the probability of a κ event at time t. For $\kappa = E$, $p_\kappa(t) = p_{exp}(t)$, and for $\kappa = I$, $p_\kappa(t) = p_{inf}$ for all t. The reverse move of an addition update is a removal update, and vice versa.

Again, we could then choose t arbitrarily, but this is likely to waste compute, for instance adding an event that has 0 probability of occurring. We can introduce qualifiers to ensure that at time t there are individuals that can be affected by the event, and that the probability of the event is non-zero. In addition, it's possible that altering timesteps with higher probability of events (typically in the middle of the epidemic) are less likely to invalidate the epidemic, so we can weight each t by its probability of event κ.

Adding an event

If $\Delta > 0$ (an addition), choose a t such that

$$\begin{cases} \{\mathcal{S}(t-1) > 0 \quad \textbf{and} \quad p_{exp}(t) > 0\}, & \text{if } \kappa = E, \\ \{\mathcal{E}(t-1) > 0 \quad \textbf{and} \quad p_{inf} > 0\}, & \text{if } \kappa = I. \end{cases}$$

weighting each t by $p_\kappa(t)$ so that timesteps that are more likely to have events get events proposed more often. Since p_{inf} is fixed in time, this is not strictly necessary for the infection events.

As such, the proposal density is given by;

$$q_{\Delta=1}(\mathbf{d}\kappa' \mid \mathbf{d}\kappa) = \frac{1}{2} \cdot \frac{1}{\sum_s \mathbb{1}\left\{\mathcal{X}(s-1) > 0 \ \textbf{and} \ p_\kappa(s) > 0\right\}} \cdot \frac{p_\kappa(t)}{\sum_s p_\kappa(s)},$$

where $\mathcal{X} \in \{\mathcal{S}, \mathcal{E}\}$ relates to the event type as appropriate, and $\mathbb{1}\{\} = 1$ if there are individuals that can change state and the probability of such an event is non-zero, and 0 otherwise. This is the probability of choosing the given timestep out of all timesteps that could have events, weighted by the probability of events occurring at that timestep. The half represents the probability of choosing an addition event, and the proposal density for the reverse move is given by,

$$q_{\Delta=1}(\mathbf{d}\kappa' \mid \mathbf{d}\kappa) = \frac{1}{2} \cdot \frac{1}{\sum_s \mathbb{1}\{d'_\kappa(s) > 0\}} \cdot \frac{p'_\kappa(t)}{\sum_s p'_\kappa(s)},$$

which is the probability of choosing this timestep to remove events from out of all events that have timesteps, where $\sum_s \{d\kappa'(s) > 0\}$ is the number of timesteps with one or more events of type κ after proposing the move, and does not equal $\sum_s \{d\kappa(s) > 0\}$.

Removing an event

If $\Delta < 0$ (a removal), choose a t such that $d_\kappa(t) > 0$ for $t \in [1, (T-1)]$.

As such, the proposal density is given by;

$$q_{\Delta=-1}(\mathbf{d}\kappa' \mid \mathbf{d}\kappa) = \frac{1}{2} \cdot \frac{1}{\sum_s \mathbb{1}\{d\kappa(s) > 0\}} \cdot \frac{p_\kappa(t)}{\sum_s p_\kappa(s)},$$

and

$$q_{\Delta=-1}(\mathbf{d}\kappa' \mid \mathbf{d}\kappa) = \frac{1}{2} \cdot \frac{1}{\sum_s \mathbb{1}\left\{\mathcal{X}'(s-1) > 0 \ \textbf{and} \ p_\kappa(s) > 0\right\}} \cdot \frac{p'_\kappa(t)}{\sum_s p'_\kappa(s)},$$

where $\mathcal{X} \in \{\mathcal{S}, \mathcal{E}\}$ relates to the event type as appropriate and the logic is the same as in the addition case.

3.7.3.4 "Adding or Removing an event" Metropolis-Hastings acceptance probabilities

For the exposure events, the MH acceptance probabilities will be given by

$$\begin{aligned}\alpha &= \min\left\{\frac{\pi(\mathbf{dE}'|\mathbf{dI},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)}{\pi(\mathbf{dE}|\mathbf{dI},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)}\frac{q_u(\mathbf{dE}\mid\mathbf{dE}')}{q_u(\mathbf{dE}'\mid\mathbf{dE})},\quad 1\right\}\\ &= \min\left\{\frac{\prod_{t=k}^{T}\left[\pi(dE_t',dI_t,dR_t|\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\right]}{\prod_{t=k}^{T}\left[\pi(dE_t,dI_t,dR_t|\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\right]}\frac{q_u(\mathbf{dE}\mid\mathbf{dE}')}{q_u(\mathbf{dE}'\mid\mathbf{dE})},\quad 1\right\},\end{aligned}$$

where the product is over the timestep where the addition/removal occurred onward, and for those parts of the likelihood that are affected as detailed in Section 3.7.2.2. The $q(\cdot)$ represent the proposal distributions for adding/removing an exposure event.

For the infection events, the MH acceptance probabilities will be given by

$$\begin{aligned}\alpha &= \min\left\{\frac{\pi(\mathbf{dI}'|\mathbf{dE},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)}{\pi(\mathbf{dI}|\mathbf{dE},\mathbf{dR},\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)}\frac{q_u(\mathbf{dI}\mid\mathbf{dI}')}{q_u(\mathbf{dI}'\mid\mathbf{dI})},\quad 1\right\}\\ &= \min\left\{\frac{\prod_{t=k}^{T}\left[\pi(dE_t,dI_t',dR_t|\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\right]}{\prod_{t=k}^{T}\left[\pi(dE_t,dI_t,dR_t|\mathcal{S}(0),\mathcal{E}(0),\mathcal{I}(0),\mathcal{R}(0),\beta,\delta,\gamma)\right]}\frac{q_u(\mathbf{dI}\mid\mathbf{dI}')}{q_u(\mathbf{dI}'\mid\mathbf{dI})},\quad 1\right\},\end{aligned}$$

where the product is over the timestep where the addition/removal occurred onward, and for those parts of the likelihood that are affected as detailed in Section 3.7.2.2. The $q(\cdot)$ represent the proposal distributions for adding/removing an infection event.

3.7.3.5 Augmenting the initial conditions

The greater the level of discretisation, the larger the proportion of removal events that will occur during the first two timesteps. Due to the fact that events cannot occur to the same individual in the same timestep, this means that the majority of the exposure and infection events that preceded these removal events must have

occurred before the first timestep - i.e. in the initial conditions. The number of events that occurred in the initial conditions, however, is unknown. At the extreme, if every event occurred in the initial conditions, the epidemic would be valid, but the S to E and E to I rates would have to be extremely small relative the removal rate, that none occurring 'during' the epidemic. For this reason we need to be able to augment and explore the initial conditions, moving events between them and the main data. In conjunction with the 'move' events data augmentation step, it is sufficient and simple to just move events between the initial conditions and the first timestep.

When the magnitude of the timestep is small, the resolution is likely sufficient to allow most if not all events to occur during the epidemic, and so minimal compute is used for this step. When the magnitude is large, it is likely that the likelihood will need to be calculated on every iteration.

Choose an event type from the set $\kappa \in \{E, I\}$. Let Δ here be the direction of movement, and sample Δ from the set $\{-1, 1\}$ to represent a move from $t = 1$ to the initial conditions, or a move from the initial conditions to $t = 1$ respectively.

As such, the proposal density is simply given by;

$$q(\mathbf{X}_0', \mathbf{d}\kappa' | \mathbf{X}_0, \mathbf{d}\kappa) = \frac{1}{2},$$

and the Metropolis-Hastings acceptance probability for $S \to E$ events is given by

$$\alpha = \min \left\{ \frac{\pi(dE_1', dI_1, dR_1 | \mathcal{S}(0)', \mathcal{E}(0)', \mathcal{I}(0), \mathcal{R}(0), \beta, \delta, \gamma)}{\pi(dE_1, dI_1, dR_1 | \mathcal{S}(0), \mathcal{E}(0), \mathcal{I}(0), \mathcal{R}(0), \beta, \delta, \gamma)}, \quad 1 \right\},$$

and the Metropolis-Hastings acceptance probability for $E \to I$ events is given by

$$\alpha = \min \left\{ \frac{\pi(dE_1, dI_1', dR_1 | \mathcal{S}(0), \mathcal{E}(0)', \mathcal{I}(0)', \mathcal{R}(0), \beta, \delta, \gamma)}{\pi(dE_1, dI_1, dR_1 | \mathcal{S}(0), \mathcal{E}(0), \mathcal{I}(0), \mathcal{R}(0), \beta, \delta, \gamma)}, \quad 1 \right\}.$$

3.8 Block Adaptive MCMC for S-E-I-R

Having derived all of its components, we now lay out a more sophisticated MCMC algorithm for making inference on epidemic data in an S-E-I-R framework. We call this algorithm the "Adaptive Block MCMC"; "Block" refers to the idea of sampling the parameters of interest as a set, using a multi-site sampler, taking into account the correlation-covariance relationships, and "Adaptive" refers the process of automatically adapting the tuning parameters to optimise the algorithm. We are considering the case of an incomplete epidemic, so we will include the details of both data augmentation steps.

In this section we present the algorithm and a series of subroutines for making inference on an SEIR chain binomial epidemic. We begin with an overview of the process, and then lay out the details of each element.

3.8.1 The Algorithms

Algorithm 1 is an overview of the MCMC schema which shows the steps taken in each iteration. The subroutines used within it are presented in the sections that follow. During each iteration a new set of parameters are proposed, and then the initial conditions, S to E, and E to I transition events are augmented using "move event" and "add/remove event" data augmentation steps. Finally the adaptive tuning parameters are updated and results recorded.

Algorithm 1: Block Adaptive MCMC Algorithm

Input : $\mathsf{N_{its}}$ = Total desired number of iterations eg. 10^6,
Data = All state and event data,
$(\lambda,\ m,\ \Delta_m)$ = Initial values of the tuning parameters

Output : $\mathsf{Results}$ = Parameter values for each iteration

Elements: $\boldsymbol{\theta}$ = Epidemic parameters, $\mathsf{n_{tune}}$ = Tuning block counter

Set up
  `Initialise` $\boldsymbol{\theta}_{\mathsf{cur}}$
  `Set` it = 1, $\mathsf{n_{tune}} = 1$
Process
  while $it \leqslant \mathsf{N_{its}}$ **do**
    `Blk-Adpt-Metropolis-Hastings-Step()`'s **for** Parameters (Subroutine 3.1)**:**
      $[\beta, \delta, \gamma] \in \boldsymbol{\theta}$
    `Metropolis-Hastings-Step()`'s **for** Data Augmentation (Subroutine 3.3)**:**
      Augment SE initial conditions (Subroutine 3.4)
      Augment EI initial conditions (Subroutine 3.5)
      Move S→E events in time (Subroutine 3.6)
      Move E→I events in time (Subroutine 3.7)
      Add/Remove S→E events (Subroutine 3.8)
      Add/Remove E→I events (Subroutine 3.9)
    Record the $\mathsf{Results}$
    if $it = 25 \cdot (\mathsf{n_{tune}})$ **then**
      $\mathsf{n_{tune}} = \mathsf{n_{tune}} + 1$
    end
    it = it+1
  end

Subroutine 3.1 presents the subroutine for proposing and updating the transition parameters using the Block Adaptive schema. The parameters are drawn on the log scale from one of two Gaussian proposal distributions, one with a scaled identity covariance matrix, and one with a covariance matrix based on the previous accepted samples. The proposed values are then accepted or rejected using a Metropolis-Hastings step, and the hyper-parameters of the proposal distributions are then automatically tuned. The subroutine of the tuning is given in the algorithms following.

Subroutine 3.1: Block Adaptive Metropolis-Hastings Step for Parameters

Input : $(\lambda, \mathsf{m}, \Delta_{\mathsf{m}})$ = Current values of the tuning parameters, and $\mathsf{N_{its}}$, Data, $\mathsf{n_{tune}}$

Output : $\boldsymbol{\theta}_{\mathsf{cur}}$ = Updated epidemic parameters, $(\lambda, \mathsf{m}, \Delta_{\mathsf{m}})$ = Updated tuning parameters

Elements: $\pi(\boldsymbol{\phi}|X)$ = Joint conditional posterior of parameters ϕ given Data X, $q(\boldsymbol{\psi}|\boldsymbol{\phi})$ = Prob. of proposing parameters ψ given the current parameters ϕ, d = Dimension of $\boldsymbol{\theta}_{\mathsf{cur}}$, Σ = Proposal posterior co-variance matrix

Propose Update

- **if** $it \leqslant \min(5000, \mathsf{N_{its}}/10)$ **then**
  - **if** $it = 25 \cdot \mathsf{n_{tune}}$ **then**
    - *Update* λ *using* `Tune_λ()` *(Subroutine 3.2)*
  - **end**
  - `Draw` $log(\boldsymbol{\theta}_{\mathsf{prime}}) \sim N(log(\boldsymbol{\theta}_{\mathsf{cur}}), \frac{\lambda^2}{d} I_d)$
- **else**
  - **with** *5% chance* **then**
    - `Set` $\Sigma = \frac{\lambda^2}{d} I_d$ *(1)*
  - **else**
    - `Set` $\Sigma = m^2 \times$ *[Current empirical Posterior Co-Variance Matrix] (2)*
  - **end**
  - `Draw` $log(\boldsymbol{\theta}_{\mathsf{prime}}) \sim N(log(\boldsymbol{\theta}_{\mathsf{cur}}), \Sigma)$
- **end**

Accept/Reject

- `Calculate` $\pi(\boldsymbol{\theta}_{\mathsf{cur}}|X)$, $\pi(\boldsymbol{\theta}_{\mathsf{prime}}|X)$, $q(\boldsymbol{\theta}_{\mathsf{cur}}|\boldsymbol{\theta}_{\mathsf{prime}})$, $q(\boldsymbol{\theta}_{\mathsf{prime}}|\boldsymbol{\theta}_{\mathsf{cur}})$ using `Posterior_fn()`
- `Calculate` the Metroplis-Hastings acceptance probability as $\alpha = \min\left(1, \frac{\prod_d[\boldsymbol{\theta}_{\mathsf{prime}}] \cdot \pi(\boldsymbol{\theta}_{\mathsf{prime}}|X)}{\prod_d[\boldsymbol{\theta}_{\mathsf{cur}}] \cdot \pi(\boldsymbol{\theta}_{\mathsf{cur}}|X)}\right)$
- Accept or reject the proposal
- **if** $\Sigma =$ *(2)* **then**
  - **if** *update accepted* **then**
    - `Set` m = m + $2.3(\frac{\Delta_m}{\sqrt{it}})$
  - **else**
    - `Set` m = m - $(\frac{\Delta_m}{\sqrt{it}})$
  - **end**
- **end**

Subroutine 3.2 presents the subroutine for tuning the scaling factor of the uncorrelated proposal distribution for the parameters. If the acceptance rate is above the desired value then the tuning parameter is made larger, otherwise it is made smaller. The tuning parameter is updated every 25 iterations up to the 5000^{th} iteration, then it is fixed.

Subroutine 3.2: Function: Tune_λ()

Input : λ_{cur} = The current value of λ for the parameter block of interest,
n_{tune} = The number of tuning blocks so far,
Results = The acceptance (0/1) of the update steps so far
Output : $\lambda_{updated}$ = The updated value of λ
Elements: it = iterations,
acc_prop = Acceptance proportion for the 25 iterations in the n_{tune} th block,
ν = Change in the λ

Function Tune$_\lambda$()
  Calculate acc_prop
  if acc_prop < 0.33 **then**
    Set $\nu = -\min(0.05, \frac{1}{\sqrt{n_{tune}}})$
  else
    Set $\nu = \min(0.05, \frac{1}{\sqrt{n_{tune}}})$
  end
  $\log(\lambda_{updated}) = \log(\lambda_{cur}) + \nu$
  Return($\lambda_{updated}$)
end

Subroutine 3.3 presents the framework for data augmentation of the partially observed and occult events. Depending on the data, event, and proposal function, different subroutines presented in the algorithms that follow are used. First new data is proposed and the posteriors calculated, then the updates are accepted or rejected based on the Metropolis-Hastings acceptance probability.

Subroutine 3.3: Metropolis-Hastings Step for Data Augmentation

Input : `Proposal_fn` = A function to generate the proposal,

$\boldsymbol{\theta}$ = Current values of the parameters,

and $\mathsf{N}_{\mathsf{its}}$, Data

Output : Data = Updated epidemic data

Elements: $\pi(\boldsymbol{X}|\boldsymbol{\theta})$ = Likelihood of the epidemic given parameters $\boldsymbol{\theta}$,

$q(\boldsymbol{X}|\boldsymbol{Y})$ = prob. of proposing data $\boldsymbol{X}$ given the current data $\boldsymbol{Y}$

Propose Update

Propose an update to Data, $\boldsymbol{X}_{\mathsf{cur}}$, using `Proposal_fn()`

`Calculate` $q(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{X}_{\mathsf{prime}})$, $q(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{X}_{\mathsf{cur}})$ using `Proposal_fn()`

Accept/Reject

`Calculate` $\pi(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{\theta})$, $\pi(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{\theta})$ using `Posterior_fn()`

`Calculate` the Metropolis-Hastings acceptance probability as

$$\alpha = \min\left(1, \frac{\pi(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{\theta}) \cdot q(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{X}_{\mathsf{prime}})}{\pi(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{\theta}) \cdot q(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{X}_{\mathsf{cur}})}\right)$$

Accept or reject the proposal

Subroutine 3.4 presents the algorithm for augmenting the S and E states by moving events between the initial conditions and the first timestep, and accepting/rejecting using a Metropolis-Hastings step and recording the results.

Subroutine 3.4: Function: Propose to augment the S and E initial conditions

Input : Data = The states and events of the epidemic at all timesteps

Output : Data′ = The states and events of the epidemic at all timesteps after the update

Elements: Δ = The direction of the move.

Function `Prop_Augment_Init_SE()`

  Generate $\Delta \in \{-1, 1\}$

  if $\Delta > 0$ **then**

    Change an initial E to an initial S

    Add an S to E event at $t = 1$

  else

    Change an initial S to an initial E

    Remove an S to E event at $t = 1$

  end

  `Calculate` the proposal probabilities using

  $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2}$

  $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2}$

  `Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

Subroutine 3.5 presents the algorithm for augmenting the E and I states by moving events between the initial conditions and the first timestep, and

accepting/rejecting using a Metropolis-Hastings step and recording the results.

Subroutine 3.5: Function: Propose to augment the E and I initial conditions

Input : Data = The states and events of the epidemic at all timesteps

Output : Data′ = The states and events of the epidemic at all timesteps after the update

Elements: Δ = The direction of the move.

Function `Prop_Augment_Init_EI()`

  Generate $\Delta \in \{-1, 1\}$

  if $\Delta > 0$ **then**

    Change an initial I to an initial E

    Add an E to I event at $t = 1$

  else

    Change an initial E to an initial I

    Remove an E to I event at $t = 1$

  end

  `Calculate` the proposal probabilities using

  $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2}$

  $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2}$

  `Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

Subroutine 3.6 presents the algorithm for augmenting the S to E transition events by moving one through time, and accepting/rejecting using a Metropolis-Hastings step and recording the results.

Subroutine 3.6: Function: Propose to move an S to E event through time

Input : Data = The states and events of the epidemic at all timesteps

Output : Data′ = The states and events of the epidemic at all timesteps after the update

Elements: t = A timestep in the data,

Δ = The magnitude and direction the event is moved in time

Function `Prop_Move_dE()`

  Generate $\Delta \in \{-1, 1\}$

  if $\Delta > 0$ **then**

    Choose a timestep, $t \in 1 : (T-1)$, weighted by $d_E(t)$

  else

    Choose a timestep, $t \in 2 : T$, weighted by $d_E(t)$

  end

  Update the Data to create Data′

  `Calculate` the proposal probabilities using

  $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{d_E(t)}{\sum_s \{d_E(s)\}}$

  $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{d'_E(t+\Delta)}{\sum_s \{d'_E(s)\}}$

  `Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

Subroutine 3.7 presents the algorithm for augmenting the E to I transition events by moving one through time, and accepting/rejecting using a Metropolis-Hastings step and recording the results.

Subroutine 3.7: Function: Propose to move an E to I event through time

Input : Data = The states and events of the epidemic at all timesteps

Output : Data′ = The states and events of the epidemic at all timesteps after the update

Elements: t = A timestep in the data,

Δ = The magnitude and direction the event is moved in time

Function `Prop_Move_dI()`

  Generate $\Delta \in [-1, 1]$

  if $\Delta > 0$ **then**

    Choose a timestep, $t \in 1 : (T-1)$, such that $d_I(t) > 0$

  else

    Choose a timestep, $t \in 2 : T$, such that $d_I(t) > 0$

  end

  Update the Data to create Data′

  `Calculate` the proposal probabilities using

  $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{d_I(s)>0\}}$

  $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{d'_I(s)>0\}}$

  `Return(`Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$`)`

end

Subroutine 3.8 presents the algorithm for augmenting the S to E transition events by adding an additional event or removing an event, and accepting/rejecting using a Metropolis-Hastings step and recording the results.

Subroutine 3.8: Function: Propose to add or remove an S to E event

Input : Data = The states and events of the epidemic at all timesteps

Output : Data′ = The states and events of the epidemic at all timesteps after the update

Elements : t = A timestep in the data,
$\mathcal{S}(t-1)$ = The number of susceptibles used to generate the exposure events,
$\mathsf{p}_{\mathsf{exp}}(t)$ = The probability of exposure at time t

Function `Prop_AddRem_dE()`

  Generate $\Delta \in [-1, 1]$

  if $\Delta > 0$ **then**

    Choose a timestep, $t \in 2:T$, such that $\{\mathcal{S}(t-1) > 0$ **and** $\mathsf{p}_{\mathsf{exp}}(t) > 0\}$, weighted by $\mathsf{p}_{\mathsf{exp}}(t)$

  else

    Choose a timestep, $t \in 2:T$, such that $d_E(t) > 0$

  end

  `Calculate` the proposal probabilities using

  if $\Delta > 0$ **then**

    $q(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{X}_{\mathsf{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{\mathcal{S}(s-1)>0 \text{ and } p_{exp}(s)>0\}} \cdot \frac{p_{exp}(t)}{\sum_s p_{exp}(s)}$

    $q(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{X}_{\mathsf{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{d'_E(s)>0\}}$

  else

    $q(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{X}_{\mathsf{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{d_E(s)>0\}}$

    $q(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{X}_{\mathsf{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{\mathcal{S}'(s-1)>0 \text{ and } p'_{exp}(s)>0\}} \cdot \frac{p'_{exp}(t)}{\sum_s p'_{exp}(s)}$

  end

  `Return`(Data′, $q(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{X}_{\mathsf{prime}})$, $q(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{X}_{\mathsf{cur}})$)

end

Subroutine 3.9 presents the algorithm for augmenting the E to I transition events by adding an additional event or removing an event, and accepting/rejecting using

a Metropolis-Hastings step and recording the results.

Subroutine 3.9: Function: Propose to add or remove an E to I event

Input : Data = The states and events of the epidemic at all timesteps

Output : Data′ = The states and events of the epidemic at all timesteps after the update

Elements: t = A timestep in the data,
$\mathcal{E}(t-1)$ = The number of exposed used to generate the E to I events,

Function `Prop_AddRem_dI()`

- Generate $\Delta \in \{-1, 1\}$
- **if** $\Delta > 0$ **then**
  - Choose a timestep, $t \in 2:T$, such that $\{\mathcal{E}(t-1) > 0\}$
- **else**
  - Choose a timestep, $t \in 2:T$, such that $d_I(t) > 0$
- **end**
- `Calculate` the proposal probabilities using
- **if** $\Delta > 0$ **then**
  - $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{\mathcal{E}(s-1)>0\}}$
  - $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{d'_I(s)>0\}}$
- **else**
  - $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{d_I(s)>0\}}$
  - $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{\mathcal{E}'(s-1)>0\}}$
- **end**
- `Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

3.9 Results

Using the above algorithm and subroutines we ran inference for four discretisations of an epidemic simulated in a continuous-time GSE SEIR construction. Figure 3.1 shows the simulated epidemic at the 4 levels of discretisation. produced by counting the number of events of each type that occurred in each window of size of Δt. We use the data of this epidemic from time $t = 0$ to the 250^{th} removal event at time $t = 274$, which leaves us with an incomplete epidemic.

The epidemic was simulated to have 25% of the population infected by the final observation, and be ongoing. We assumed the removal events were observed, but not the exposure or infection events. We assumed the epidemic to be ongoing such that we did not know the total number of exposure and infection events, just that they were bounded below by the number of removals at any given timestep. There are three parameters of interest: β, δ, γ.

The population size was 1000 individuals, initialised with one infected individual. The priors were set to be $\beta \sim \mathsf{Gamma}(5, 0.05)$, $\delta \sim \mathsf{Gamma}(1.6, 0.05)$, and $\gamma \sim \mathsf{Gamma}(4.4, 0.05)$ such that the mean of the prior is the true value. The form of the Gamma distribution we are choosing to use has $\nu > 0$ as the shape parameter, and $\lambda > 0$ as the rate parameter.

We have chosen to explore 4 levels of discretisation that can be interpreted in the following way: $\Delta t = 0.2$ is an approximation to a continuous-time inference, $\Delta t = 1$ can be considered the standard discretisation for this epidemic, and can be interpreted as daily observations, $\Delta t = 7$ can be interpreted as weekly observations, and $\Delta t = 30$ can be interpreted as monthly observations. Simply counting the number of continuous events that occurred in each Δt window has the potential to invalidate an assumption of the epidemic than an individual can only experience one type of event in each timestep. As such we initialise the data for the inference by fixing the assumed known removal timesteps, and back-generating appropriate exposure and infection timesteps.

We ran the same MCMC process for each discretisation for 3 million samples.

Below we present the results of 2.5 million samples after burn-in for the $\Delta t = 1$ discretisation, with commentary and comparison to the results of the other resolutions.

The $\Delta t = 0.2$ inference is the most accurate, approximating the continuous-time inference with minimal biases introduced, however it is also the most inefficient. In fact it is more inefficient than the continuous-time model, as there are more timesteps/rows in the data than there are events. This model is the baseline against which we will compare all the results, and took 14759 seconds to run. This is compared to the 3256 seconds for the $\Delta t = 1$ model. The cost of this rapid increase in computation by using $\Delta t = 1$ was a small increase in the variance of the unimodal posteriors. We did not explicitly make inference for a continuous-time model in this instance, however it would have roughly 750 events/rows of data for which to compute a likelihood, compared to the $\Delta t = 0.2$ models roughly 1250, and the $\Delta t = 1$ models roughly 250 for context.

Overall the algorithm with $\Delta t = 1$ recovered uni-modal posterior distributions which contained the true values of the parameters. The true values of the parameters lie very close the areas of highest posterior mass. The shapes of the posteriors are uni-modal and distinct, without excessively long tails as we can see in Figure 3.2. The mixing was good, with large jumps and time spent exploring all areas of the posterior mass, though slightly worse than for the $\Delta t = 0.2$ model, as seen in Figure 3.4. The trade off between the exposure rate, onset of infection rate, and removal rate can be seen in Figure 3.3 with the strong elliptical shapes of the contour plots. The red dotted lines represent the true parameters, and the yellow dotted lines represent the position of the pair of parameters with the highest posterior density, calculated by dividing the state-space into a fine grid and finding the centroid of the bin with the greatest density. We can see that in all cases the values are within the main posterior mass. In particular we can see that larger S to E transition rates, leading to more exposed individuals quicker, are matched with larger I to R transition rates, meaning shorter infectious periods.

Table 3.2 presents the summaries of the marginal posterior distributions. The average acceptance rate of the parameter block draw was 30.44%. The average acceptance rate for moving S to E exposure events was 90.06%. The average acceptance rate for moving E to I infection events was 88.93%. The average acceptance rate for adding or removing S to E exposure events was 58.6%. The average acceptance rate for adding or removing E to I infection events was 21.93%. The effective sample size for β was 746.85. The effective sample size for δ was 1423.42. The effective sample size for γ was 677.57.

The parameter estimates for the $\Delta t = 0.2$, $\Delta t = 7$, and $\Delta t = 30$ models are presented in Tables 3.1, 3.3, and 3.4 respectively. The values were similar for the $\Delta t = 0.2$ model, noting that the effective sample sizes were reversed with δ being significantly higher and the other two being lower. Overall this lends credence to the discretisation efforts of $\Delta t = 1$, with roughly equal quality of inference and a significant speed increase.

The same cannot be said, however, for the more extreme discretisations. Whilst it is true that there is a notable speed increase, with $\Delta t = 7$ taking 831 seconds, and $\Delta t = 30$ taking a mere 237 seconds, the bias introduced has a noticeable impact on accuracy. As expected due to the constraint of each individuals transitions having to occur in different timesteps, a minimum waiting period is enforced for each event type equal to the timestep size. In this case those timesteps were too large, and led to a high levels of inaccuracy, with for instance β becoming extremely inflated and δ being severely deflated. The acceptance rates are roughly the same, as are the effective sample sizes, if not higher, but the inference is severely inaccurate, exacerbated by the informative priors which are aligned with the continuous model rather than each discretisation.

	True Value	Mean	95% CI	Std. Dev.	ESS
β	0.25	0.19615	(0.101, 0.325)	0.0582	1157.63
δ	0.08	0.09189	(0.0375, 0.198)	0.0410	144.73
γ	0.22	0.15918	(0.0801, 0.2660)	0.0478	1197.91

Table 3.1: The summary of the marginal posterior distributions for $\Delta t = 0.2$.

	True Value	Mean	95% CI	Std. Dev.	ESS
β	0.25	0.2126	(0.100, 0.393)	0.0762	746.85
δ	0.08	0.0979	(0.0383, 0.2130)	0.0446	1423.42
γ	0.22	0.1893	(0.0833, 0.3780)	0.0764	677.57

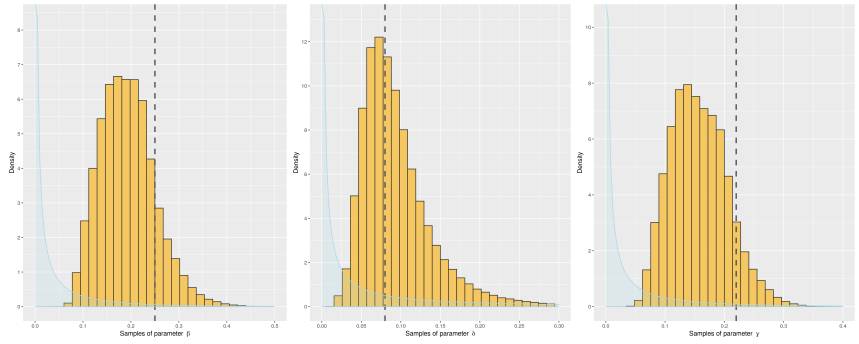
Table 3.2: The summary of the marginal posterior distributions for $\Delta t = 1$.

	True Value	Mean	95% CI	Std. Dev.	ESS
β	0.25	0.11066	(0.0657, 0.2200)	0.0191	476.53
δ	0.08	0.16779	(0.00835, 0.33500)	0.0686	1217.93
γ	0.22	0.14606	(0.0547, 0.2580)	0.0452	870.26

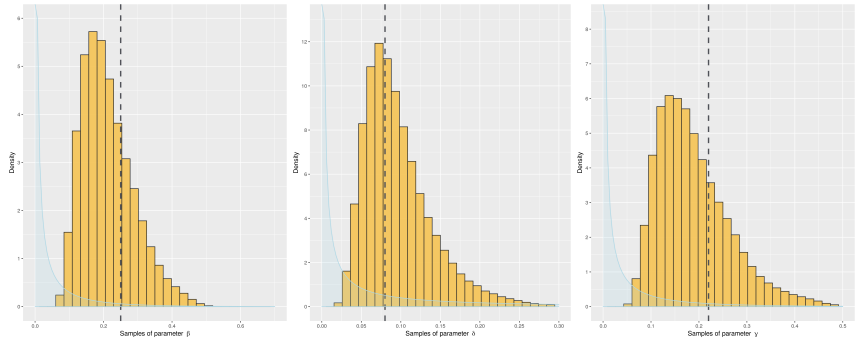
Table 3.3: The summary of the marginal posterior distributions for $\Delta t = 7$.

	True Value	Mean	95% CI	Std. Dev.	ESS
β	0.25	0.7894	(0.0284, 1.1100)	0.1570	260.03
δ	0.08	0.001818	(0.00153, 0.01110)	0.000162	1228.98
γ	0.22	0.13936	(0.00493, 0.40400)	0.1120	2056.68

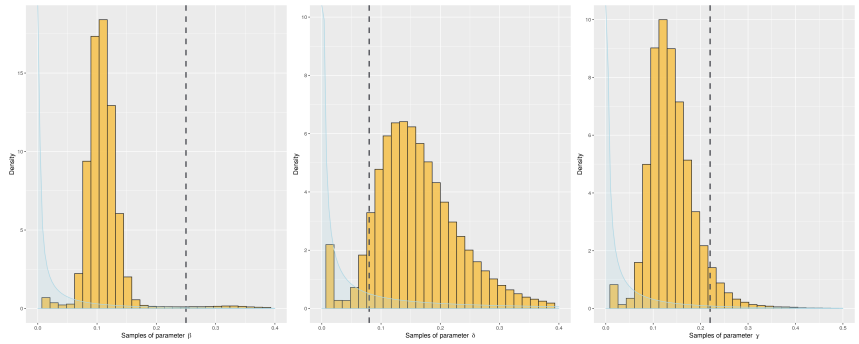
Table 3.4: The summary of the marginal posterior distributions for $\Delta t = 30$.



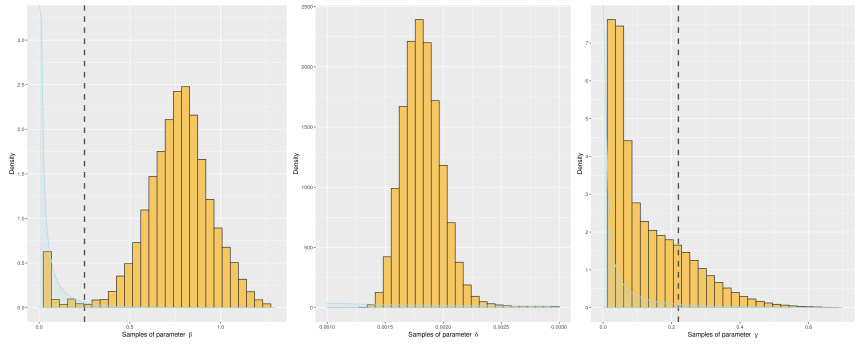
(a) Histogram for β, δ and γ with $\Delta t = 0.02$.



(b) Histogram for β, δ and γ with $\Delta t = 1$.



(c) Histogram for β, δ and γ with $\Delta t = 7$.



(d) Histogram for β, δ and γ with $\Delta t = 30$.

Figure 3.2: Results: The plots show the marginal posterior histograms for each of the parameters of interest with (a) $\Delta t = 0.02$, (b) $\Delta t = 1$, (c) $\Delta t = 7$, (d) $\Delta t = 30$. The true value of the parameter used to generate the simulation is represented by the dashed line. The prior distribution of the parameter is shown in blue.

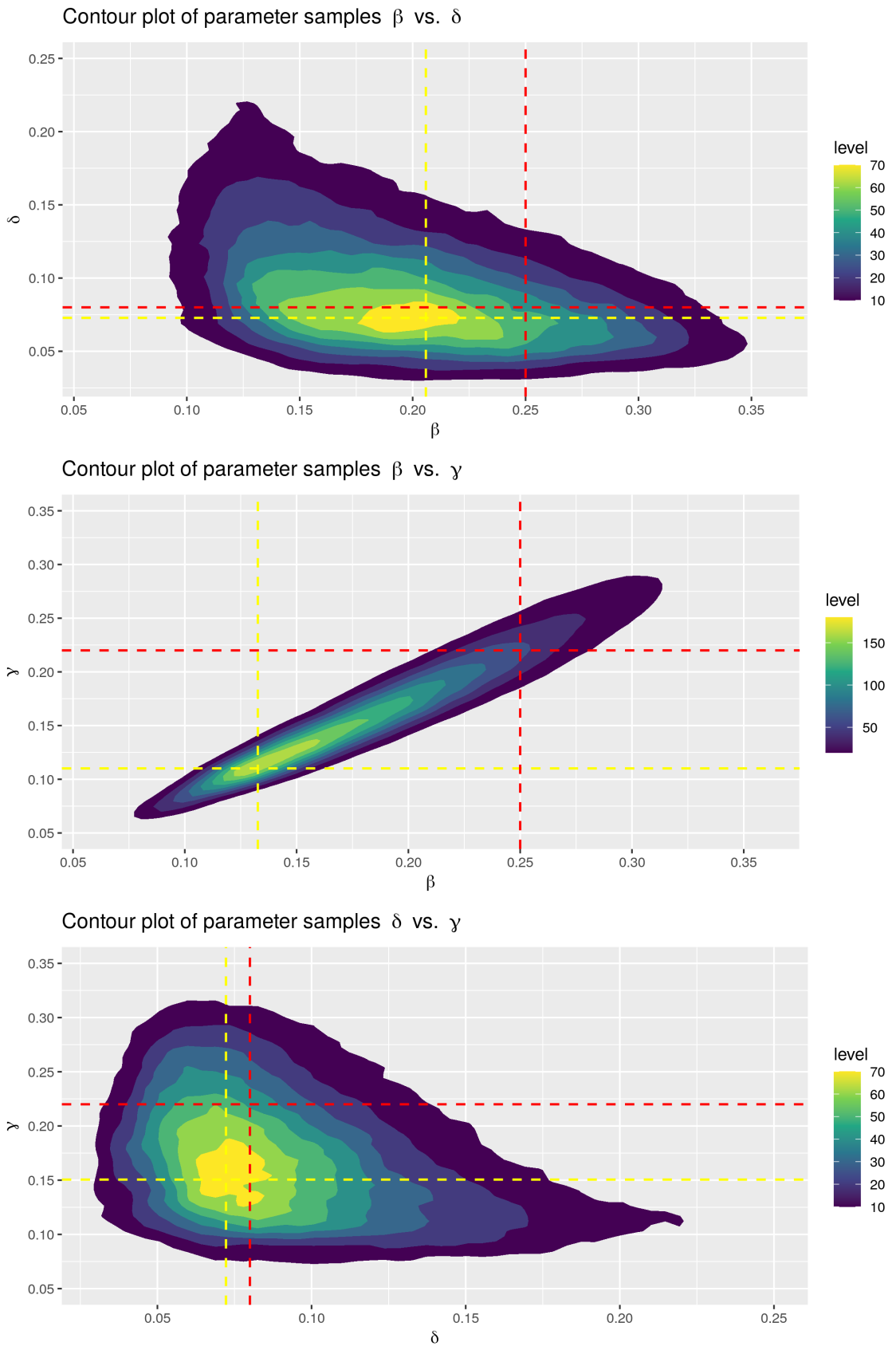
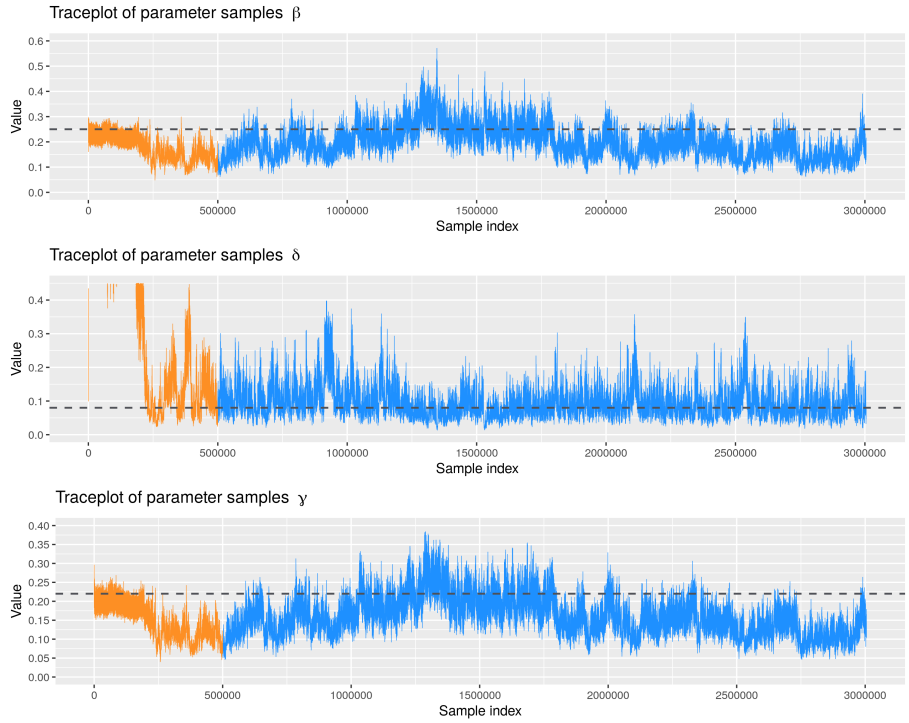
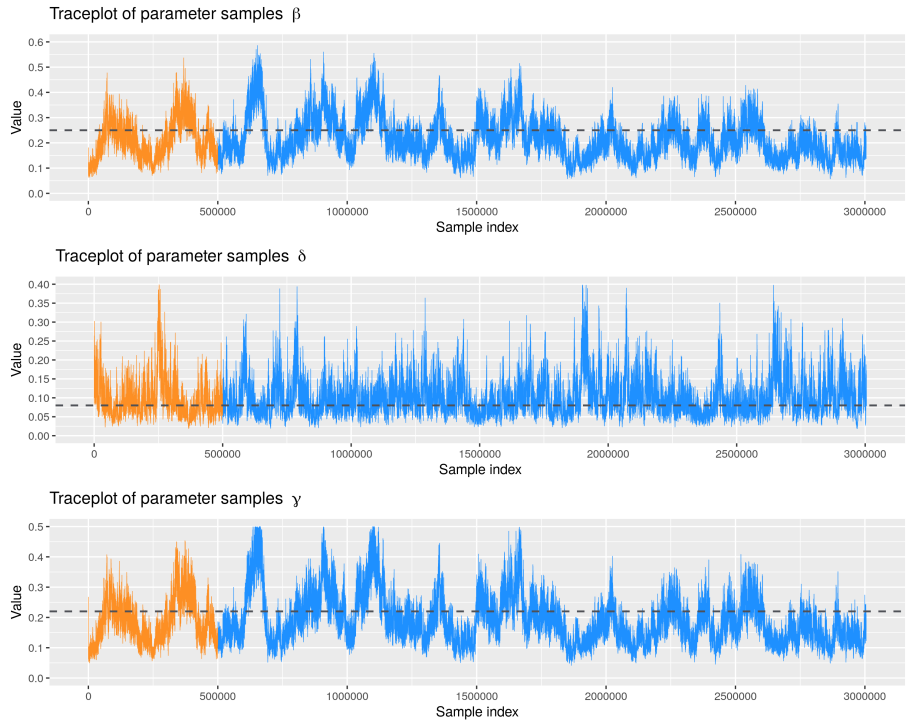


Figure 3.3: Results: Contour plots of the posterior samples for each pair of the parameters of interest with $\Delta t = 1$. Brighter contours represent areas of higher density. The yellow dashed lines show the pair-wise point of highest density on the contour plots, and the red dashed lines represent the true values of the parameters that generated the simulation. From top to bottom the plots show β *vs* δ, β *vs* γ, and δ *vs* γ.

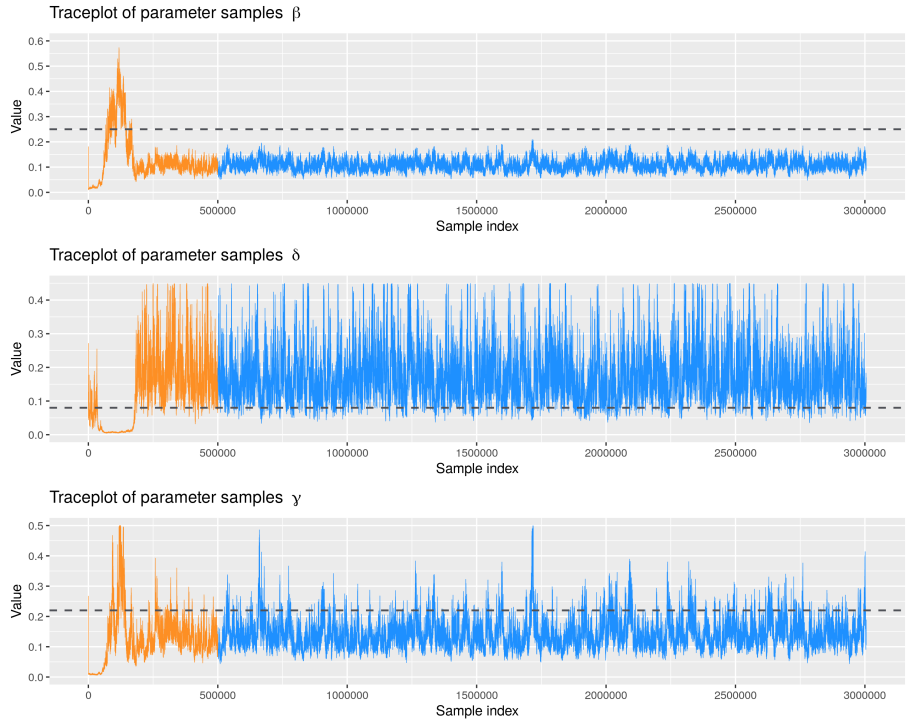


(a) Trace plot for β, δ and γ with $\Delta t = 0.02$.

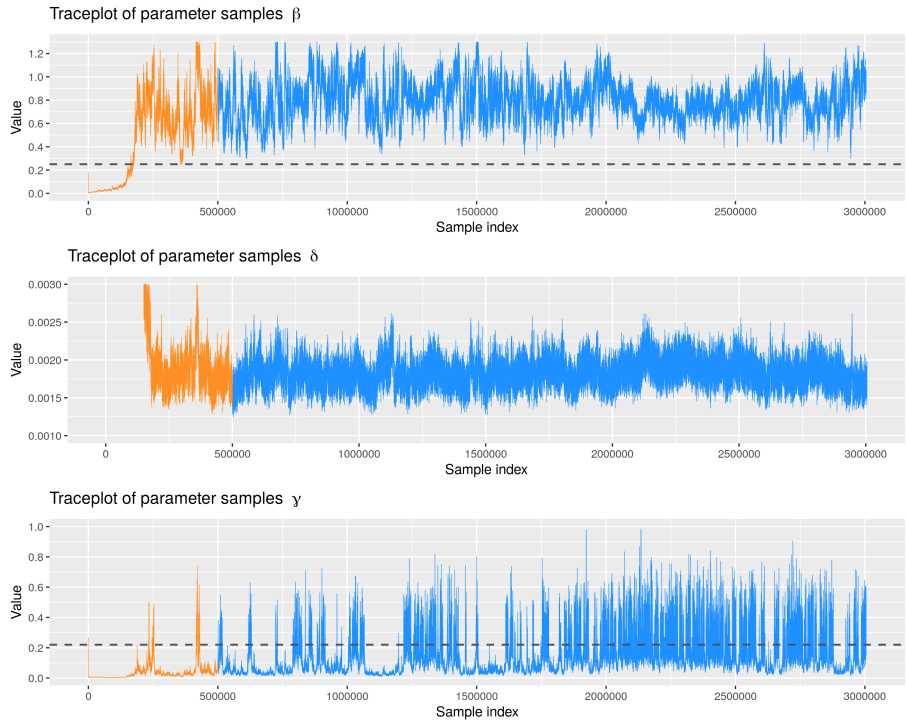


(b) Trace plot for β, δ and γ with $\Delta t = 1$.

Figure 3.4: Results: Trace plots of the posterior samples for (a) $\Delta t = 0.02$ and (b) $\Delta t = 1$. The initial burn-in is represented in orange, which gets discarded. The remainder of the chain in blue is assumed to represent the stationary distribution of the chain. The true value of the parameter is shown by the dashed line.



(a) Trace plot for β, δ and γ with $\Delta t = 7$.



(b) Trace plot for β, δ and γ with $\Delta t = 30$.

Figure 3.5: Results: Trace plots of the posterior samples for (a) $\Delta t = 7$ and (b) $\Delta t = 30$. The initial burn-in is represented in orange, which gets discarded. The remainder of the chain in blue is assumed to represent the stationary distribution of the chain. The true value of the parameter is shown by the dashed line.

3.10 Discussion

In this chapter we explored the Chain-Binomial discrete-time population-level epidemic model, and made inference on a continuous-time SEIR epidemic discretised at different resolutions. Following from Chapter 2 it was clear that we needed yet further methods for dealing with the challenges presented by big-data epidemics. Of most concern are the prohibitive computation costs and algorithm inefficiency arising from huge amounts of missing data. Our proposition is to discretise the epidemic data using an appropriate timestep size, and trade a small amount of accuracy for a large boost in performance. Our results showed that indeed we could still make accurate inference on the epidemic whilst realising a huge speed up. They also made it clear that over-discretising would lead to wild inaccuracies and uncertainty due to the differing assumptions in the construction of the continuous-time and discrete-time epidemic models. There is clearly an optimal level of discretisation that will balance accuracy and computational efficiency, which will likely be dependent on the dynamics of the epidemic, or more simply, the rates of events.

It must be noted however that the efficiency of the algorithm was poor overall, with an effective sample size of less than 700 for 2.5 million samples in some cases, which also may lead one to question the accuracy of the inference, however, there are lots of avenues for improvement. Better tuning of the data augmentation steps is likely to improve efficiency. In this demonstrative example we allowed for the minimum change possible during each update, moving, adding, or removing one event at one timestep, and we only allowed for one update of each type per sampling of the parameters. With acceptance rates of roughly 90% for the moving of events, and 59% for adding or removing S to E events, clearly we could update more at any given time, and run multiple updates between parameter draws. It is unlikely that the acceptance rate will scale linearly with these tuning parameters, and it is possible that any increase would lead to unacceptably low acceptance rates, but this does not prohibit running multiple sequential updates between parameter updates.

The other option is to improve the proposal distributions of the data augmen-

tation. For instance for moving S to E events in time, we weight each timestep, t, by the proportion of the total S to E events that occur during time t. We also considered a scenario where we weighted each timestep the probability of S to E events at time t, $p_{exp}(t)$, which we found to be less efficient. The basic alternative is to just sample t uniformly at random, which would be the least efficient method as described in Section 3.7. A more sophisticated proposal distribution would likely improve the efficiency of the algorithm, though without experimentation it is not obvious which would be best.

The inference between the different discretisations should not necessarily produce the same estimates however. This is in part due to the fact that each discretisation has different assumptions about the minimum time that is spent in each state, equal to the size of timestep, which then imposes a minimum amount of time each individual spends as part of the epidemic before recovery. With $\Delta t = 1$ this is 2 (1 in S, 1 in E, then removal), but with $\Delta t = 30$ this is 60 (30 in S, 30 in E, then removal). This could explain why the E to I transition rate, γ, has such a small posterior mean for $\Delta t = 30$, to ensure that individuals transition to the infectious state quick enough to align with the recoveries, especially with β being estimated so large.

Now that we have established that discretising the epidemic is a valid potential method for reducing the computational burden of making inference on epidemics whilst maintaining accuracy, we wish to explore it's effectiveness for complex big-data epidemics, for which we will need an example. In the following chapter we introduce our case study example, Bovine Tuberculosis in England and Wales, which consists of 100s of millions of records and complex disease dynamics and testing schemes. We explore the historic context of this disease, the work on this disease done by others, and finally explore the datasets provided to us by APHA with the intention of building our own epidemic model.

Chapter 4

Bovine Tuberculosis

4.1 Introduction

With over 20 million cattle, complex disease dynamics, multi-host populations, and a rich detailed data set, Bovine Tuberculosis in England and Wales serves as a great example of a large scale complex big data epidemic. In this chapter we introduce the context on which the rest of the thesis is based. We explore the historical and current state of Bovine Tuberculosis in England and Wales, its mechanisms and dynamics, and provide an overview of previous work that has attempted to model it's spread. This lays the foundation for our novel work presented in the following chapters making inference using full likelihood-based MCMC methods.

In Section 4.2 we provide an overview of the current literature around Bovine Tuberculosis and modelling its epidemics. In Section 4.2.3 we give a summary of Brooks-Pollock, Roberts, and Keeling, 2014 from which this work is inspired. Following this in Sections 4.3, 4.4, 4.5, 4.6, and 4.7 we present and explore the data provided to us by the Animal and Plant Health Agency (APHA) and explain its influence on our modelling choices. In the following chapters we go on to extend the methodologies presented in Chapters 1 and 3 and apply it to make inference on this data and our model.

4.2 Literature Review

4.2.1 Bovine Tuberculosis

In this subsection, unless otherwise specified, information was obtained from the Animal and Plant Health Agency (APHA, 2021). Bovine Tuberculosis is a chronic bacterial infectious disease of animals. It is primarily associated with cattle however all mammals are susceptible, including humans. It is caused by a bacterium called *Mycobacterium bovis* (*M. bovis*).

Transmission can occur directly through nose-to-nose contact and through contact with saliva, urine, faeces, and milk. It transmits very slowly between cattle, but has a high potential for spread due to the chronic nature of infection (Gordon and Barrow, 2018). The disease is transmissible to humans through unpasteurised milk or dairy products, inhaling bacteria breathed out by infected animals, or inhaling bacteria released from the carcasses or excretions of infected animals. Indeed the majority of humans infected with bTB each year in the UK work with cattle (Kirchhelle, 2020). Being a bacterial disease means that the pathogen can also survive outside of the host (Gordon and Barrow, 2018), with some studies suggesting *M. bovis* could remain on pastures, alive and virulent, for at least 49 days (Maddock, 1933). No study to date has successfully quantified the relative importance of direct contact, aerosol spread and indirect environmental spread on the risk of infection (Gordon and Barrow, 2018).

Bovine Tuberculosis is also hard to identify as symptoms often only develop in advanced stages of infection, and these symptoms can be similar to other diseases. The symptoms typically include wasting/getting thinner, light recurring fever, and cattle are weak, with a reduced appetite. However symptoms do not present in all infected cattle before slaughter, and the disease itself can have greatly varying incubation periods (Gordon and Barrow, 2018).

The long incubation periods and asymptomatic nature of the disease, combined with the workings of the cattle industry, mean that the majority of cases cannot

be observed naturally and need to be tested for. The primary test for bTB in cattle is Tuberculin testing (Gordon and Barrow, 2018). Tuberculin was discovered in 1890 and causes an immune reaction in cattle (Kirchhelle, 2020). Animals are given an injection of both bovine and avian tuberculin and animals that react to the bovine more than the avian tuberculin are considered as skin test reactors (Welsh Government, 2018). Skin test reactors are cattle that are considered to have tested positive for Bovine Tuberculosis. Post-mortems are also carried out at the abattoir, and the organisms are attempted to be cultured (Welsh Government, 2018). In rare cases Interferon-gamma blood tests are used to confirm suspected cases. Tuberculin skin tests are well known for having imperfect sensitivity, only being able to identify an estimated 70%-90% of infected cattle (Brooks-Pollock, Roberts, and Keeling, 2014; Green and Cornell, 2005; Monaghan et al., 1994; Conlan et al., 2012; de la Rua-Domenech et al., 2006). In addition, the test has reduced sensitivity for a period of time following infection (Brooks-Pollock, Roberts, and Keeling, 2014; Gordon and Barrow, 2018; Pollock et al., 2001).

In the early part of the 20th century Bovine Tuberculosis was considered a substantial risk to consumers due to contaminated milk and meat (Waddington, 2004). Between 25 and 40% of cattle in the UK were estimated to be infected with bTB (Waddington, 2004). Efforts to detect and eradicate the disease became an important part of public health policy as annual death rates in humans associated with bTB reached 3,000 with many more crippled, with children at greater risk (Waddington, 2004). In addition it was also a huge financial burden for the farming industry (Waddington, 2004). Tuberculin was adopted as the main diagnostic test for the disease and a test-and-slaughter scheme was introduced in the 1950s (Waddington, 2004). Eventually the transmission routes to humans were eradicated through the use of pasteurisation (Waddington, 2004). By the mid-1960s, bTB was restricted to a few pockets of infection in southwest England (Reynolds, 2006) and incidence fell to around 0.22% of herds (Brooks-Pollock, Roberts, and Keeling, 2014).

However between 1990 and the mid-2010s the level of Bovine Tuberculosis in the UK showed a rising trend, with the number of confirmed herd incidents increasing at a rate of 18% per year until 2001 (Reynolds, 2006). In 2001 a foot-and-mouth disease (FMD) epidemic majorly disrupted the testing and slaughter associated with bTB. Atypical movements to restock FMD-affected farms resulted in a huge spike in bTB cases across the country (Gopal et al., 2006), which increased the rate of spread across the UK dramatically (Reynolds, 2006). As shown in Figures 4.1 and 4.2, since around the mid-2010s the trend has plateaued or shown a minor decrease, but levels of infection remain high across the UK.

Bovine TB has been found in a number of wild animals including foxes, stoat, common shrew, yellow-necked mouse, wood mouse, field vole, grey squirrel, roe deer, red deer, fallow deer, muntjac (DEFRA, 2004) and badgers (Krebs et al., 1997, The Ministry of Agriculture, Fisheries, and Food, 1976). Studies have shown that the strains of the disease found in cattle and badgers in the same area are usually identical (Biek et al., 2012; Goodchild et al., 2012). It has also been shown that incidence of badger bTB infections are highly correlated with, and much more likely to occur around, confirmed cases in cattle (Goodchild et al., 2012). However the question of whether badgers are a driver of disease spread in cattle, and whether culling them is an effective strategy to reduce spread, is still highly contested (Kirchhelle, 2020).

Bovine TB was first found in wild badgers in 1971, and by 1975 a full-scale badger culling policy was introduced (Kirchhelle, 2020). The policy was marred by controversy and as the years went on animal advocates won legal protections for badgers and policy changed many times as evidence was deemed inconclusive (Kirchhelle, 2020). Over the decades changes in government led to repeated back and forths in badger culling policy and huge media outcry (Kirchhelle, 2020). Current policy is dedicated to a badger culling plan to reduce the spread of bTB, with licences being granted to different areas based on need (Natural England, 2022).

A randomised control trial on the culling of badgers to affect bovine TB spread

was conducted in 2007 by the Independent Scientific Group on Cattle TB (The Independent Scientific Group on Cattle TB, 2008). The trial concluded that "reactive" culling, targeting specific badger social groups which could have caused TB breakdowns in cattle, appeared to increase the incidence of confirmed cattle breakdowns (herds where at least one cow tested positive for Bovine Tuberculosis) by 27% due to increased badger displacement. On the other hand "proactive", or widespread, culling was associated with a reduction in the number of TB breakdowns by 23% inside the culling areas, but lead to increased levels in neighbouring areas. In addition they state that sustained culling over 5 years would lead to the beneficial effects outweighing the negative ones, but the costs associated greatly outweighed the modest beneficial effect. Indeed Jenkins, Woodroffe, and Donnelly, 2010 continued to monitor the sites and found that after a 5 year culling period farms within the culling area had roughly 30% lower incidence compared to non-culled areas, but 3 years post-culling any benefits inside culled areas were no longer detectable, and that the financial cost of sustained culling was up to 3.5 times higher than the savings from culling. It has been shown that repeated badger culling can in fact lead to an increase in bTB incidence in badgers, especially when the geography allows badgers from nearby lands to recolonise culled areas (Woodroffe et al., 2006). Separately it has been shown via the use of GPS tracking collars on badgers that culling can lead to badgers visiting 45% more fields each month, and a 20-fold increase in the odds of trespassing into neighbouring badger group territories, increasing contact (Ham et al., 2019).

The importance of each transmission pathway: cattle-to-cattle, badger-to-cattle, and environment-to-cattle, is still a highly contested topic. Using the randomised clinical trial for badger culling data, it has been estimated that up to 52% of herd-level infections were contributed to by badgers, but only 5.7% were directly caused by badgers (Donnelly and Nouvellet, 2013). In addition, increased herd size may correlate with increased risk of disease incidence and persistence (Brooks-Pollock and Keeling, 2009). Another analysis of outbreak and movement data suggested

that 16% of herd infections were due directly to cattle movements, and there was low cattle-to-cattle transmission but high local environmental effects (Green et al., 2008). Other sources examined roadkill of badgers in areas of emerging bTB spread, and equally struggle to identify links between bTB spread and badgers (Swift et al., 2021). Nor is it a simply a challenge of data availability, with models of the effect of culling depending on the social responses of badgers, but robust to the population density of cattle and badgers (Smith et al., 2016).

Current testing policy splits the country up into 3 areas; High Risk, Low Risk, and Edge. Regular testing is performed with intervals determined by the risk area of the county. There are 48-month cycles of testing for farms in the low risk area, 6-month cycles in the high risk area, and 6 to 12-month cycles in the Edge area (APHA, 2023a). A map of the risk areas can be found here: `https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1029652/pti-map.pdf`. We are unable to recreate this map in the thesis due to licensing agreements with Ordnance Survey.

If one or multiple reactors (cattle that react more to the bovine tuberculin injection than to the avian tuberculin injection) are found during routine testing, then the reactors are slaughtered and the farm is put under movement restrictions (APHA, 2022). The farm will then be required to conduct follow up tests every 60 days. After 2 consecutive tests with negative results the farm will regain Officially TB Free status and movement restrictions will be lifted. After this the herd will be required to be tested again after 6 and 18 months, before returning to its usual testing intervals (APHA, 2022).

Testing can also occur for other reasons. If bTB is suspected in the farm due to clinical signs in an animal, slaughter house inspection, or inconclusive reactors (IRs) found in the herd, the farm is put under movement restrictions until a negative culture result and a clear herd level test. If the culture is positive, or any TB skin test reactors are found, the farm will become a breakdown herd (APHA, 2023b) and follow the procedure above. Pre-movement and post-movement tests also occur for

individual animals.

Herd incidence, herd prevalence, new herd incidents, Non-Officially-TB-Free (NOTF), and number of animals slaughtered are the key metrics the APHA use to measure Bovine TB in the country. The following definitions are cited from APHA (APHA, 2023c):

- **Herd incidence** is the rate of new herd incidents per 100 herd years at risk. The rate is based around the total amount of time that herds tested were unrestricted and at risk of infection since the end of their last TB incident or negative herd test, rather than the total number of tests carried out on those herds.

- **Herd prevalence** is defined as the percentage of all registered herds which were not Officially TB Free (OTF) due to a TB incident.

According to the latest government reports available (APHA, 2023c), showcased in Figures 4.1 and 4.2, the herd incident rate was 8.4% between January 2022 and December 2022 in England overall and 6.5% in Wales. These were down in both cases compared to 2021. Herd prevalence was 4.5% in England and 5.3% in Wales, again down since 2021.

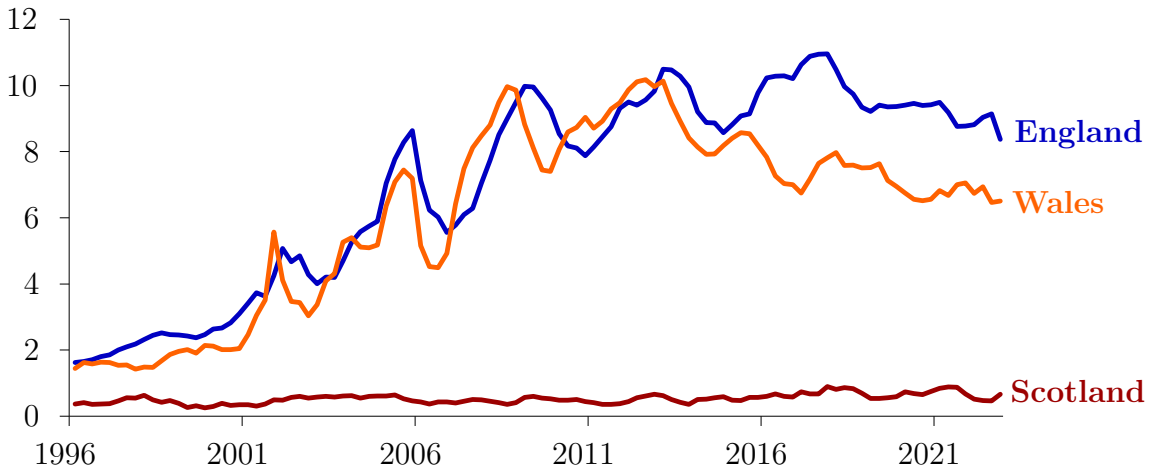


Figure 4.1: Long term view of new herd incidents per 100 herd years at risk of infection during the year. Recreated with permission from (https://www.gov.uk/government/statistics/historical-statistics-notices-on-the-incidence-of-tuberculosis-tb-in-cattle-in-g figures-to-december-2022-published-08-march-2023) under the Open Government Licence v3.0 (https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/).

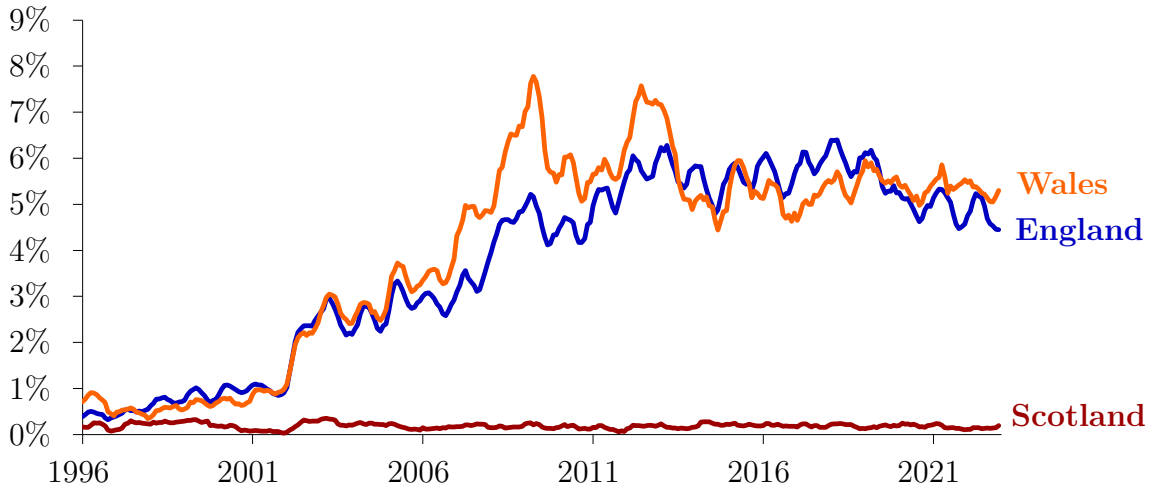


Figure 4.2: Long term view of number of herds which were non-OTF at the end of the period due to a TB incident as a percentage of registered and active herds. Recreated with permission from (https://www.gov.uk/government/statistics/historical-statistics-notices-on-the-incidence-of-tuberculosis-tb-in-cattle-in-g figures-to-december-2022-published-08-march-2023) under the Open Government Licence v3.0 (https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/).

However the number of new herd incidents increased by 3% in England from

2021 to 2022. In Wales there was a 10% decrease. Overall the number of herds NOTF decreased by 3% in England and 5% in Wales year on year.

At the same time was a 20% decrease in the number of cattle slaughtered due to a TB incident in England from 2021 to 2022, and an 11% decrease for Wales.

Estimates of the basic reproductive number, the number of additional expected cases caused by each infected animal, range from 1.01 - 4.9 for cattle (Cox et al., 2005; Conlan et al., 2012; Brooks-Pollock, Conlan, et al., 2013; Brooks-Pollock, Roberts, and Keeling, 2014), and 1.03 - 1.35 for badgers (Smith, 2001; Mathews et al., 2005; Wilkinson et al., 2004; Delahay et al., 2013). These numbers suggest we would expect the epidemic to continue to grow, though the extent and nature of the expected growth would depend on the model used.

4.2.2 The modelling landscape

There have been a large number of models presented for Bovine Tuberculosis, both in the UK and around the world. The main research questions of interest for bTB are the pathways of transmission - whether spread is mainly due to within farm cattle spread, cattle movements, or environmental reservoirs like badgers, and thus what intervention strategies will be most effective. The approaches to these questions vary significantly, but deterministic mathematical models dominate the literature over stochastic models. The models can be categorised in a number of different ways - some are population based (Brooks-Pollock, Roberts, and Keeling, 2014; Mathews et al., 2005; Cox et al., 2005; Brooks-Pollock, Conlan, et al., 2013), and some are agent based models (Moustakas and Evans, 2017; Wilkinson et al., 2004) dealing with individual cattle and badgers. Some models are mathematical and deterministic (Cox et al., 2005; Brooks-Pollock and Wood, 2015; Donnelly and Nouvellet, 2013; Mathews et al., 2005; Brooks-Pollock, Conlan, et al., 2013), and some are stochastic (Brooks-Pollock, Roberts, and Keeling, 2014; Dawson, Werkman, and Brooks-Pollock, 2015; Wilkinson et al., 2004; Moustakas and Evans, 2017). Of the stochastic models, some are simulation studies that aim to explore

qualities of the disease through simulating epidemics using a range of possible parameter values (Dawson, Werkman, and Brooks-Pollock, 2015; Wilkinson et al., 2004), and some are models that infer properties of the epidemics through identifying their best fitting parameters (Moustakas and Evans, 2017; Brooks-Pollock, Roberts, and Keeling, 2014). For those models that are fitted to data, the methods range from sensitivity analyses with latin hyper-cubes of parameters (Moustakas and Evans, 2017; Donnelly and Nouvellet, 2013) to approximate methods like Approximate Bayesian Computation (ABC) (Conlan et al., 2012; Brooks-Pollock, Roberts, and Keeling, 2014). At the time of writing we have yet to find a paper that uses a full likelihood based approach, or Bayesian fitting methods such as Markov Chain Monte Carlo (MCMC).

Further still are models that incorporated genetic information and testing to attempt to model the pathways of disease spread in cattle and badgers (Kao, Price-Carter, and Robbe-Austerman, 2016, Crispell et al., 2019, Rossi et al., 2022). We do not consider these methods in this thesis but incorporation of such data could provide a powerful tool for validating and improving the inference.

The models have a wide array of goals. Some focus on the network aspect of the disease created by the movements of cattle (Dawson, Werkman, and Brooks-Pollock, 2015), whilst others focus of the dynamics of the disease itself (Conlan et al., 2012; Mathews et al., 2005; Cox et al., 2005). Some focus primarily on the cattle and perhaps an abstract background infection (Green et al., 2008), whilst others explicitly model badger populations (Conlan et al., 2012; Brooks-Pollock, Roberts, and Keeling, 2014; Mathews et al., 2005; Wilkinson et al., 2004; Cox et al., 2005). Some focus on the disease burden in cattle (Cox et al., 2005; Brooks-Pollock, Roberts, and Keeling, 2014), whilst others opt to consider the economic factors (Brooks-Pollock and Keeling, 2009), or intervention schemes (Wilkinson et al., 2004). Some models are simple, aiming to give some insight into realistic boundaries of disease behaviour (Mathews et al., 2005; Cox et al., 2005), whilst others try to capture the complex dynamics of the disease in order to answer important questions

(Wilkinson et al., 2004; Brooks-Pollock, Roberts, and Keeling, 2014).

Our model can be considered a population level discrete time compartmental model. It is inspired by and goes beyond work presented by Brooks-Pollock, Roberts, and Keeling, 2014.

There are many supporting studies which help inform the modelling. For instance when tracked with GPS collars, badgers prefer cattle pastures but avoid cattle (Rosie Woodroffe et al., 2016). This gives support to models which have an environmental reservoir background infection but don't have direct spread between badger and cattle. When the 2001 FMD epidemic reduced bTB testing across the country, an increase in the prevalence of the disease in both badgers and cattle was observed - suggesting possible multi-directional pathways of disease spread (Woodroffe et al., 2006). In the case of big data epidemics where cattle movements play a key role, it has been shown that by targeting farms with the highest number of movements, accurate predictions on the size and spatial spread of epidemics can be made (Dawson, Werkman, and Brooks-Pollock, 2015), which may be one way to reduce computational burden. There is also another layer of questioning asking how much of infection is driven by animal movements, environmental reservoirs, and missed infectives due to low testing sensitivity. Some models estimate that 24-50% of recurrent breakdowns can be attributed to infection missed by tuberculin testing (Conlan et al., 2012).

4.2.3 Brooks-Pollock et al, 2014

Our model for the spread of Bovine Tuberculosis in UK farms is influenced by Brooks-Pollock, Roberts, and Keeling, 2014. In the following section we will detail the model and fitting methods used by Brooks-Pollock, Roberts, and Keeling, 2014 for reference. We will refer to this model as the BP-Model.

The BP-Model is a dynamic stochastic spatial model for Bovine Tuberculosis. It is a discrete time meta-population model that combines within-farm and between farm spread.

Within each farm there is an SEI process for the cattle driven by cattle-to-cattle transmission, and a local farm environmental reservoir effect for background infection. Between farm spread is primarily driven by cattle movements between farms, and a parish level environmental reservoir. The environmental reservoirs capture contaminated pastures and infected wildlife, though the two are inseparable. Animals are removed from farms through a testing process that mimics historical government policy.

The states of each farm are updated at the day level using the following equations:

$$
\begin{aligned}
S_i(t+1) &= S_i(t) + b_i(t) - \Lambda_{i,t} - \sum_j M^S_{i,j,t} + \sum_j M^S_{j,i,t} \\
E_i(t+1) &= E_i(t) + \Lambda_{i,t} - A_{i,t} - D^E_{i,t} - \sum_j M^E_{i,j,t} + \sum_j M^E_{j,i,t} \\
I_i(t+1) &= I_i(t) + A_{i,t} - D^I_{i,t} - \sum_j M^I_{i,j,t} + \sum_j M^I_{j,i,t}
\end{aligned}
$$

Where $S_i(t), E_i(t), I_i(t)$ are the number of Susceptible, Exposed, and Infectious cattle on farm i at the start of day t, $b_i(t)$ is the recorded number of births on farm i during day t. Then $\Lambda_{i,t}$ and $A_{i,t}$ are the number of newly exposed and newly infectious cattle on farm i during day t, $D^E_{i,t}$ and $D^I_{i,t}$ are the number of exposed and infectious cattle on farm i removed due to testing on day t. Finally $M^S_{i,j,t}, M^E_{i,j,t}, M^I_{i,j,t}$ are the number of susceptible, exposed, and infectious cattle that are moved from farm i to farm j during day t.

When the model is simulated, only the initial conditions (based on detections), the births, and the number of cattle movements between each pair of farms each week are considered known. Λ, A, D, and M are independent random variables and represent exposure, transition from exposed to infectious, detections, and movement states. The events are generated with the following independent distributions assumed:

$$\Lambda_{i,t} \sim \text{Binomial}(S_i(t), \lambda_i)$$
$$A_{i,t} \sim \text{Binomial}(E_i(t), \alpha)$$
$$D^E_{i,t} \sim \text{Binomial}(E_i(t), \rho\rho_E)$$
$$D^I_{i,t} \sim \text{Binomial}(I_i(t), \rho)$$
$$(M^S_{i,j,t}, M^E_{i,j,t}, M^I_{i,j,t}) \sim \text{Multinomial}(m_{i,j,t}, p^S_{i,t}, p^E_{i,t}, p^I_{i,t})$$

where the force of infection on farm i is given by $\lambda_i = 1 - \exp\left\{-\frac{\beta I_i(t)}{N_i(t)} - f v_i(t) - F V_i(t)\right\}$, $m_{i,j,t}$ is the total number of moves from farm i to j on day t, and $p^X_{i,t} = \frac{X_i(t)}{S_i(t)+E_i(t)+I_i(t)}$ is the probability of picking an animal of type $X \in \{S, E, I\}$ to move.

The parameters are defined as β, the cattle-to-cattle infection rate, f, the farm-to-cattle transmission scalar, and F, the parish-to-cattle transmission scalar. The detection parameter ρ represents the probability of detecting an infectious cow that is tested, and ρ_E is a scalar multiplier for ρ between 0 and 1 that represents the reduction in effectiveness of the test for exposed cattle. If $\rho_E = 0$ then exposed cattle cannot be detected, and if $\rho_E = 1$ then exposed cattle have the same probability of being detected as infectious cattle.

The environmental reservoir at the farm level is denoted v_i and at the parish level is denoted V_i. The environmental reservoirs decay at a constant rate of ϵ. Each timestep the current environmental reservoir decays with constant rate ϵ, and additional environmental reservoir is contributed by the proportion of infectious animals in the parish, and are updated using:

$$v_i(t+1) = v_i(t)\exp\{-\epsilon\} + \frac{I_i(t)}{\epsilon N_i(t)}[1 - \exp\{-\epsilon\}]$$
$$V_i(t+1) = V_i(t)\exp\{-\epsilon\} + \frac{\sum_{j\in\mathcal{P}(i)} I_j(t)}{\epsilon \sum_{j\in\mathcal{P}(i)} N_j(t)}[1 - \exp\{-\epsilon\}]$$

where $\mathcal{P}(i)$ is the parish of farm i, and $\sum_{j\in\mathcal{P}(i)}$ is the sum across all farms j in parish $\mathcal{P}(i)$, which includes farm i.

The reservoirs are a weighted average of the current and historical fraction

of the creatures in the farm/parish that are infectious, with the weight decaying exponentially.

The tests are a combination of real data and procedures informed by government policy. They included all surveillance tests which concern tests for the whole herd (WHT, WHT2, RHT, CT, CON, see Table 4.8 for details) and follow up tests (SI), but discluded any tests on individual cattle such as pre-movement tests, retests after inconclusive reactions, and all Interferon-gamma blood tests, an alternative to the Tuberculin skin test. Less than 1% of all reactors are disclosed during individual animal tests. The BP-model utilises the test dates for any routine tests, and generates follow up tests as appropriate inline with government policy.

The model is initialised by seeding with data from 1996-1998 due to the uncertain completeness of this data. The model is then run forward such that by the time of the relevant simulated data only secondary cases exist.

The movements are aggregated such that each cow only has a start and end location for each day, no transient moves are considered.

The model parameters are estimated using Sequential Monte Carlo Approximate Bayesian Computation. The method works by drawing a parameter set from a prior distribution and simulating an epidemic. If the epidemic is close enough to the true data based on a handful of summary statistics then the parameter set is accepted into a sample that forms an approximation to the posterior. In addition this process is repeated a number of times with the prior of each round being the posterior of the previous round, with the sample parameter sets weighted based on how well they fit the data, and the acceptance conditions becoming more strict. This results in an approximate inference of parameters which is not guaranteed to be accurate unless certain conditions of the posterior and model error are satisfied (Wilkinson, 2013, Li and Fearnhead, 2018, Frazier et al., 2018). The summary metrics used to evaluate the fit of a simulation were the "Number of reactors per county per year" and the "Number of failed herd tests per county per year", for the years 1999-2007. The authors state that least squares error and approximate likelihood produced similar

results.

The parameter estimates given by the model are as follows:

Parameter: Meaning	Prior	Posterior Estimate (CI)
β: cattle-to-cattle transmission rate (years^{-1})	$\mathsf{Gamma}(1, 3.65)$	0.61 (0.0503, 1.54)
Γ^{-1}: average latent period (years)	$\mathsf{Gamma}(1, 22)$	11.1 (3.29, 25.7)
ϵ: environmental decay rate (years^{-1})	$\mathsf{Gamma}(1, 18.25)$	7.23 (3.57, 12.6)
f: farm environment-to-cattle transmission rate (years^{-1})	$\mathsf{Gamma}(1, 0.73)$	0.154 (0.0488, 0.309)
F: parish environment-to-cattle transmission rate (years^{-1})	$\mathsf{Gamma}(1, 0.073)$	0.0136 (0.00288, 0.0337)
ρ: test sensitivity	$\mathsf{Beta}(11.5, 5)$	0.72 (0.633, 0.806)
ρ_E: relative test sensitivity for latent cattle	$\mathsf{Uniform}(0, 1)$	0.276 (0.136, 0.488)

Table 4.1: The biological meanings, prior distributions, point estimates (expected value from the posterior) and 95% intervals calculated from the marginal posterior distributions, recreated from Brooks-Pollock, Roberts, and Keeling, 2014. The Gamma distributions are defined by their shape and scale.

Brooks-Pollock, Roberts, and Keeling, 2014 modelled the spread of Bovine Tuberculosis from 1999 to 2007, which encapsulated a highly disruptive Foot and Mouth disease epidemic. We are looking at the same data sources but for the period 2012 to 2019. As such the modelling assumptions will be different and evaluated from first principles in light of this new context, though inspiration is taken from this previous work. In addition we must consider that we are using different inference methods to any used previously, and as shown in Chapter 2 the parameterisation and form of our model can have an effect on the efficiency of our algorithm. In the following sections we provide the reader an overview of the data sources provided to us by APHA and the highlights of an exploratory data analysis used to inform our model presented in Chapter 5. The features of most importance to us are location, movements (including births and deaths), and testing.

4.3 Data Description

The data is supplied by the Animal and Plant Health Agency (APHA). The data consists of 5 files. Below we list the files, a short description, and a description of their contents. In the next section we provide dictionaries of the contents of the

data sources. Where there are data that we have not utilised in our model, we omit some detail.

The file data sets include:

- Location data containing the details of each farm
- Historic Herd data which concerns a count of the number of animals on each farm at the beginning of each month
- Animal Details data containing the unique attributes of every cow
- Births, Deaths, and Movements data taken from the Cattle Tracing System (CTS)
- Test details and results taken from the VetNet national testing database.

We have the location of every holding (of the county-parish-holding sense) in England and Wales that held, traded, or tested animals between 01-Jan-2012 and 01-Jul-2019. There are 57 counties, $10,337$ parishes, and $79,559$ holdings (unique CPH ids) total. There are $82,208$ location entries in the data, which in addition to holdings includes data on slaughter-houses, showgrounds, and others detailed in Table 4.7.

4.4 Data Dictionary

This section presents an overview of the contents of each data set we have, in the form of a data dictionary. The information presented here is intended to give the reader an in depth insight into what information was available on which to build our model.

4.4.1 CTS Locations

The CTS Locations table (Table 4.2) includes the details of locations with cattle present during the data range. It includes the unique identifier of each holding, their

geographic location coordinates, and the premises type of the holding. The different premises types are detailed in Table 4.7.

Location Database		
Field Name	Data Type	Description
CPH_FMT	nvarchar(11)	Holding identifier with formatting
CPH	nvarchar(9)	Holding Identifier (no formatting)
MapX	Integer	X Map co-ordinate
MapY	Integer	Y Map co-ordinate
PremisesType	nvarchar(2)	Premises type

Table 4.2: CTS Locations.

Having the map co-ordinate of every farm allows us the potential to explore spatial kernels for spread at the farm level, however the vast quantity of farms in the database makes this infeasible, even if we introduced a locality effect, and to reference the Near vs Far model from Chapter 2, make the scalar of the infection rate beyond distance d equal to 0. In addition, we are not provided the geometry of the farm, just the centroid or the farmhouse, and this does not inform where cattle are housed or graze, and how likely they are to interact with other farms. For instance farms within proximity to each other could border, or even share pastures, making spread more likely, or could be separated by a road or a river. For this reason we have chosen to instead look at spatial spread using the County-Parish-Holding number of the farms, treating farms in the same parish as being spatially connected. This of course has its flaws, as farms on opposite side of the parish are less likely to affect each other than farms on the borders of neighbouring parishes, however, the computational burden can be vastly reduced compared to using a continuous spatial kernel, and by grouping by parish our model aligns well with testing policy. Finally the inclusion of premises type suggests the potential for different premises purposes to be more correlated with disease spread - depending on the distribution of premises type it may make sense to treat them differently, or focus on only a subset of premises types that make up the majority of locations.

4.4.2 Historic Herd Data

The Historic Herd Data table (Table 4.3) includes the details of the estimated number of cattle on each premises for each Month/Year. It includes the unique identifier of each holding, the month, the year, and the estimated number of cattle present on the farm at the beginning of the month.

Historic Herd Database		
Field Name	Data Type	Description
CPH_FMT	nvarchar(11)	Holding identifier with formatting
CPH	nvarchar(9)	Holding Identifier (no formatting)
HistoricYear	Integer	Year
MonthNumber	Integer	Month
NumberOfAnimals	Integer	Number of animals present on first day of the month

Table 4.3: Historic Herd Data.

The historic herd data will be of limited use beyond initialising the epidemic, as the first record plus the movement, testing removals, births, and deaths should result in the same counts, adjusting for artefacts in the data. That said it does allow us to initialise the data at any date, and sense check our work for consistency.

4.4.3 Animal Details

The Animal Details table (Table 4.4) includes the details of the animal details for animals tested or moved during the date range. It includes the unique identifier of each cattle, their date of birth, their date of death, sex, and breed type.

Animal Details Database		
Field Name	Data Type	Description
Eartag	varchar(35)	Animal Identifier (Unique ID)
DoB	Date	Date of Birth
DoD	Date	Date of Death
Sex	Char(1)	Male (M) or Female (F)
BreedCode	varchar(15)	Breed code (Details omitted)

Table 4.4: Animal Details.

As an individual level model of this scale is infeasible, we have limited use for individual level data on cattle. However, it is worth exploring in the initial phase to understand if there is a need to treat any breed (for instance, which in turn is likely to correlate with farm or location) differently. In our exploration we did not find a sufficient benefit to incorporating this data.

4.4.4 CTS Movements

The CTS Movements table (Table 4.5) includes the details of the movement of cattle between premises. It includes the unique identifier of each cattle, and the unique identifiers of the farms they moved off of and on to. It also includes details of any intermediate farms where they stayed during the movement, and how long they stayed at their 'onto' destination. It also includes when the movement occurred, whether it was a cattle movement, a birth event, or a death event. And finally the age at which they moved.

Movements Database		
Field Name	Data Type	Description
Eartag	varchar(25)	Animal Identifier (Unique ID)
MovementID	Integer	Movement ID (Move sequence if >1 on a single day)
MovementDate	Date	Date of Move
OffCPH	varchar(9)	Departing CPH (no formatting)
OnCPH	varchar(9)	Destination CPH (no formatting)
Birth	bit	Birth Move (1), Non-Birth Move (0)
Death	bit	Death Move (1), Non-Death Move (0)
TransCPH	varchar(9)	CPH of Transitory premises e.g. Market
TransPremType	nvarchar(2)	Premises type for Transitory premises
StayLength	Integer	Time spent on the Destination CPH (days)
AgeAtMove	Integer	Age in Months at time of move

Table 4.5: CTS Movements.

The level of detail in the movement records opens up a wide array of possibilities, both in terms of the individual level data and the choices of aggregation. Knowing the full path of each cow allows us to extract only those locations we think of as being the most relevant. For instance due to the slow spreading nature of the disease we may view transitory premises or stays of less than a day to be inconsequential. Or as mentioned in Section 4.2 there are studies that show only using the most active nodes in the network can still give accurate inference. Either way it is likely that we will need to reduce the data in some way for computational reasons. Aggregating the data to the weekly level, we can choose to only consider the first and last farm of a cow each week. Equally for births and deaths we can just take the count of each on each farm each week.

4.4.5 Animal Test

The Animal Test table (Table 4.6) includes the details of the cattle Bovine Tuberculosis tests. It includes the unique identifier of each cow, the unique identifier of the farm they were tested on, the date of their test, and the outcome. It also contain details on the test itself such as the reason for testing and the testing method.

The coding for the reasons for testing is given in Table 4.8. The coding for the testing methods is given in Section 4.5.3. The coding for the results of the test is given in Table 4.9. The coding for the actions following the test is given in Table 4.10.

Testing Database		
Field Name	Data Type	Description
Eartag	varchar(35)	Animal Identifier (Unique ID)
CPH	char(9)	Test CPH (no formatting)
TestDate	Date	Date of test
TestType	varchar(20)	Reason for test (See Table 4.8)
Category	varchar(50)	Testing method (See Section 4.5.3)
TestRes	varchar(5)	Test Result (See Table 4.9)
TestRes2	varchar(5)	Original Test Result if superseded
Action	char(1)	Action following Test Result (See Table 4.10)
LesSH	varchar(3)	Lesions found at Slaughterhouse (Details omitted)
AvianResult	Integer	Avian Skin test difference in mm
BovineResult	Integer	Bovine Skin Test difference in mm
ReactorType	varchar(2)	Standard or Severe
AgeAtTest	Integer	Age at test in months

Table 4.6: Animal Test.

Another data set of great interest, again the level of detail in the testing data opens up a wide array of possibilities. We have chosen to not incorporate any genetic testing data in the model, due to the limited number of records that were processed. In addition whilst many of the details of each test are present, including the presence of lesions and the results of follow up tests, due to the subjectivity of the test and the dependence on the clinicians judgement, we have chosen to simply focus on the action taken following the test. We can then aggregate this as a per farm per week count. The data allow us to explore in detail, however, the different tests used and how many cases can be attributed to each. This allows us to choose whether to model each or to focus on only the most relevant.

4.5 Data Coding

This section provides further details on the coding of the data. The information presented here is intended to give the reader a deeper understanding of the categorical data provided by the APHA.

4.5.1 Premises Type

The premises type is the designation of the farm and its purpose. A farm in the sense that most people would imagine it is an Agricultural Holding, but there are many other premises that hold cattle.

Premises Type Coding	
Coding	Premises Type Description
AH	Agricultural Holding
AI	AI Sub Centre
CA	Calf Collection Centre
CC	Collection Centre (for BSE material)
EX	Export Assembly Centre
HK	Hunt Kennel
KY	Knackers Yard
LK	Landless Keeper
MA	Market
SG	Showground
SR	Slaughterhouse (Red Meat)
SW	Slaughterhouse (White Meat)
TH	Temporary Holding

Table 4.7: Premises Type.

4.5.2 Reason for test

The reason for testing is the reason why the test was performed. This typically is either because of routine testing, or testing because of an event, such as before and after movement. Tests occur to either the Whole Herd (WH) or to Individual cattle (IA).

Test Reason Coding		
Coding	Coding	Test Reason Description
VE-12M	WH	12 months post-6M test
VE-6M	WH	6 Month test
VE-AI	IA	AI animal test
VE-CLINICAL	IA	Ancillary blood test
VE-CON	WH	Contiguous test
VE-CON12	WH	12 months post CON6-Contiguous test
VE-CON6	WH	6 months post Contiguous test
VE-CT	WH	Check test
VE-CT(EM)	WH	Carried out outside the normal testing frequency for the herd, to determine its disease status when there is a suspicion of infection.
VE-CT-HS1	WH	1st hotspot check test
VE-CT-HS2	WH	2nd hotspot check test
VE-CT-NH1	WH	1st new herd check test
VE-CT-NH2	WH	2nd new herd check test
VE-IFN_ANOM	IA	IFN Anomalous Reactions Procedure
VE-IR	IA	Inconclusive Reactor Retest
VE-OT	WH	Other test
VE-POSTMT	IA	Post movement testing
VE-PRMT	IA	Pre-movement testing
VE-RAD	WH	Radial Herd Test. Stock eligibility will be as with CON tests - all bovines except calves under 6 weeks old
VE-RAD12	WH	12 months post Radial Herd Test. Stock eligibility will be as with CON tests - all bovines except calves under 6 weeks old
VE-RAD6	WH	6 months post Radial Herd Test. Stock eligibility will be as with CON tests - all bovines except calves under 6 weeks old
VE-RHT	WH	Routine Whole herd test
VE-RHT12 (S)	WH	For herds that are on 12 monthly testing intervals. Eligibility is as per 48 month Routine Herd Tests
VE-RHT24/36	WH	Carried out in parishes with a 24, 36 month testing interval
VE-RHT48	WH	Routine surveillance test carried out every 48 months
VE-SI	WH	Short Interval test
VE-SLH	IA	Slaughterhouse case
VE-TR	IA	Traced Bovine Test
VE-WHT	WH	Whole herd test
VE-WHT2	WH	Yearly test in 2 yearly testing parishes

Table 4.8: Reason for test.

4.5.3 Test Method

There are 4 kinds of testing methods in the data set. We will only concern ourselves with the TBSKINTEST which accounts for over 98% of all tests. The TB Skin Test is a tuberculin test where cattle are injected with both bovine and avian tuberculin, and cattle that have a stronger reaction to the bovine are considered reactors. The test has imperfect sensitivity and relies somewhat on subjective clinical decisions. The others tests are an antibody test, a gamma interferon test, and a slaughter house post-mortem.

4.5.4 Test Result

The result of the test describes the clinical outcome of the test. This is typically a positive, negative, or inconclusive, but different test types have different coding.

Test Result Coding	
Coding	Test Result Description
CLEAR	Clear tested
CLIN	Clinical Case
DC	Dangerous Contact
IR	Inconclusive
N	Negative blood test (Gamma Only)
OTH	Not tested but slaughtered for other reasons. Being wild and unmanageable is one.
P	Blood test with pending result(Gamma Only)
R	Reactor (Skin or Gamma test)
SL	Slaughterhouse suspect case
UNK	This predates the OTH category so they can be taken as that

Table 4.9: Test Result.

4.5.5 Action following Test Result

The TB SKINTEST in particular is somewhat subjective, so the clinical test results don't always lead to the same outcome for the cattle if the clinician deems it so.

This is the reason for the distinction, and Action following Test Result represent what actually happened to the cattle.

Action Coding	
Coding	Action Description
N	None
S	Slaughter e.g. reactor or animal taken as reactor or dangerous contact
I	Isolate i.e. an inconclusive reactor (IR) will be isolated from the rest of the herd until it is retested (after 60 days)

Table 4.10: Action following Test Result.

4.6 Descriptive Statistics

In this section we provide a summary of the data we received from the APHA. The purpose of this section is to give the reader an insight into the scale and nature of the data, as well as how much information is relevant to the model we are intending to build and fit.

By choosing to model the spatial spread of the disease, given the movements of cattle, in parish units, we are able to introduce an independence between parishes, given movements. This allows us to scale down our inference to only one parish or collection of farms to assess the spread of the disease in one area, or to divide computation. For this reason we also present here the statistics for the parish of Cheshire, our chosen example. Cheshire was chosen due to its moderate size and its location in the 'Edge-risk' policy region.

4.6.1 Locations

The data is for England and Wales and ranges between the 01-Jan-2012 and 01-Jul-2019. The data consists of a total 57 counties, broken into 10337 parishes, and further broken into 79559 unique holdings. In the county of Cheshire there are 311

parishes, divided into 2172 unique holdings.

There are 10 types of premises, with the three most common premises types being Agricultural Holding (76,348 / 94.45%), Landless Keeper (2,046 / 2.56%), and Temporary Holding (1,482 / 1.85%). In Cheshire there are only 5 types of premises, with the top three being Agricultural Holding (2,072 / 95.18%), Temporary Holding (53 / 2.43%), and Landless Keeper (47 / 2.16%).

Due to the overwhelming proportion of premises types being Agricultural Holding, and the slow spreading nature of the disease making Temporary Holdings and Landless Keepers unlikely to be contributing significantly to disease spread, its makes sense for us to focus our efforts on Agricultural Holdings - or rather for the sake of simplicity - not distinguish between premises type.

4.6.2 Cattle

There are 21,977,071 unique cattle that were born, died, moved or tested in the data set. In Cheshire there are 2,811,616 unique cattle that were born, died, moved into or out of, or were tested there.

As explored in Chapter 2 and 3, 22 million, or even 3 million, individuals is too many to be able to make inference on, even for the simplest of models. Due to the nature of Bovine Tuberculosis the model will need complexities beyond those explored thus far. As a result it is clear we will need to use a population level model that only considers the number of individuals in each state. That said there are clear geographic differences, so it should not be as extreme as one population for the whole country.

4.6.3 Movements

There are 32,476,152 total movements in the data set that are associated with 17,377,240 total unique animals that have been moved. There are 1,660,178 unique movements into, within, and out of Cheshire, which is divided as 1,005,472 unique movements out of Cheshire, 344,148 unique movements into Cheshire, and 310,558

unique movements within Cheshire. There are $1,052,263$ total unique animals moved into, out of, or within Cheshire.

There are $16,534,913$ total births and $16,409,722$ total deaths in the data set. There are $787,135$ unique births and $119,711$ unique deaths that occur within Cheshire.

We require animal movement data to model the movements which relate to the spread of the disease. However, as mentioned in Section 4.2 it may not be necessary to use every movement. Following Brooks-Pollock, Roberts, and Keeling, 2014 there is no necessity to associate any parameters with the movements, though over 60 million records will still be a challenging task to accommodate and can still affect the inference of the other parameters. We can certainly operate in a discrete population and concern ourselves with only the counts of the number of movements between each pair of farms. We can also operate in discrete time and only concern our model with the first and last farm of each cow during a timestep. From the counts it appears that on average each cow moves twice, and given the difference in births and deaths in Cheshire, it is likely one of those movements is a move to a designated slaughter location.

4.6.4 Testing

There are $69,309,947$ total unique tests performed attributed to $16,025,845$ total unique animals tested in the data set. There are 4 methods of testing which divide the tests as 'TB Skin Test' with $68,426,837$ accounting for 98.73%, 'Gamma Test' with $868,884$ accounting for 1.25%, 'Postmortem Exam' with $10,911$ accounting for 0.02%, and 'Antibody Test' with $3,312$ accounting for 0.00%. There are 57 possible reasons that a test is administered, the top 10 reasons for the whole data set are given in the table below. The other 47 test reasons account for the other 7.44%.

Test Reasons (Whole data set)			
Code	Description	Units	Proportion
VE-SI	Short Interval test (Whole Herd)	26,573,580	38.34%
VE-WHT	Whole herd test (Whole Herd)	14,622,920	21.10%
VE-6M	6 Month test (Whole Herd)	6,192,727	8.93%
VE-PRMT	Pre-movement testing (Individual Animal)	4,718,895	6.81%
VE-CON	Contiguous test (Whole Herd)	4,076,907	5.88%
VE-12M	12 months post-6M test (Whole Herd)	2,985,597	4.31%
VE-CT(I-I)	Carried out outside the normal testing frequency for the herd, to determine its disease status after voluntary slaughter (Whole Herd)	1,652,173	2.38%
VE-RHT48	Routine surveillance test carried out every 48 months (Whole Herd)	1,464,542	2.11%
VE-CON12	12 months post CON6-Contiguous test (Whole Herd)	964,761	1.39%
VE-RAD6	6 months post Radial Herd Test (Whole Herd)	900,897	1.30%

Table 4.11: Top 10 Test Reasons.

There are $3,519,067$ unique tests attributed to $750,660$ total unique animals tested in Cheshire. There are 4 methods of testing which divide the tests as 'TB Skin Test' with $3,377,149$ accounting for 95.97%, 'Gamma Test' with $141,629$ accounting for 4.02%, 'Postmortem Exam' with 258 accounting for 0.01%, and 'Antibody Test' with 31 accounting for 0.00%. There are 39 possible reasons that a test is administered, the top 5 reasons for the whole data set are given in the table below. The other 34 test reasons account for the other 13.94%.

Test Reasons (Cheshire)			
Code	Description	Units	Proportion
VE-WHT	Whole herd test (Whole Herd)	1,419,991	40.35%
VE-SI	Short Interval test (Whole Herd)	990,253	28.14%
VE-6M	6 Month test (Whole Herd)	260,277	7.40%
VE-PRMT	Pre-movement testing (Individual Animal)	225,237	6.40%
VE-IFN_LOW_IN	IFN OTFW TB Breakdown in Lower TB Incidence Area - Investigation and Interpretation (Whole Herd)	132,704	3.77%

Table 4.12: Top 5 test reasons in Cheshire.

As the disease is a chronic bacterial disease which is asymptomatic on the timescales we are concerned with, testing is the only way of identifying infected cattle, and acts as an alternative to I to R transitions, with the distinction that they no longer occur at a given rate, but on a testing schedule which varies by location. Less than 1% of infected cattle are detected through individual level tests, and since Whole Herd Tests make up the vast majority of testing, we can follow Brooks-Pollock, Roberts, and Keeling, 2014 and focus solely on whole herd tests, which also restricts us to TB Skin Tests. We can take the testing dates from the data, and assuming all animals are tested, the reason for testing is no longer necessary to consider in the model.

4.7 Exploratory Data Analysis

4.7.1 Historic Herd

The historic herd data set contains counts on the number of animals on each farm each month. From these observations we can get an idea of the size of farms. The size of farms may contribute to the disease dynamics. For instance in larger farms the disease may spread more easily, as opposed to the same number of cattle divided between multiple farms. We can see from Figure 4.3 that the majority of farms have

between 21 and 200 cattle, but the majority of cattle reside on farms that have a population of 201 to 500 cattle compared to any other group.

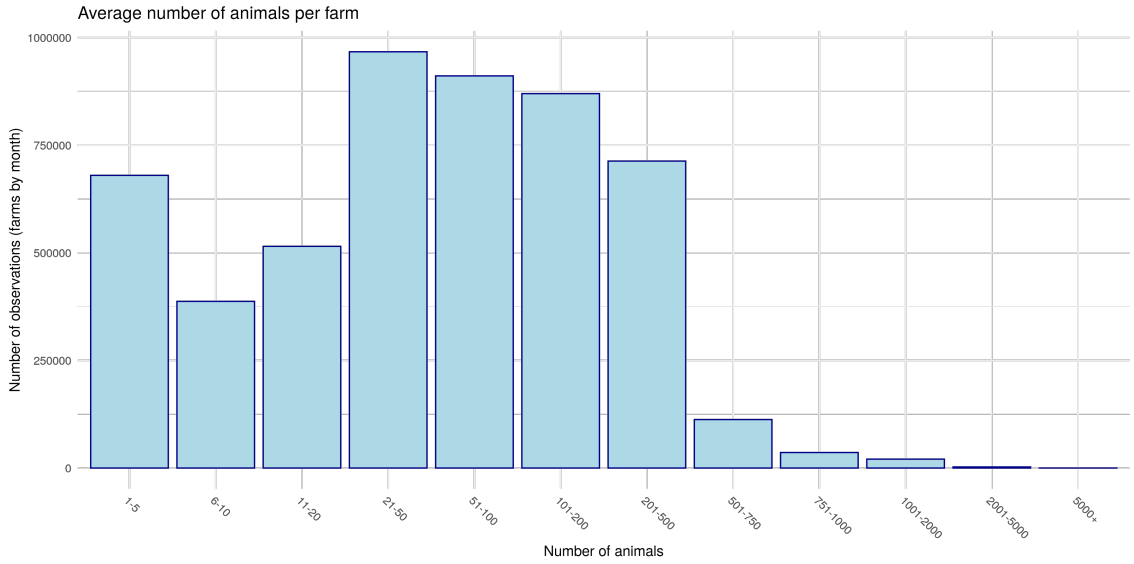


Figure 4.3: Average number of animals per farm

These unit sizes support the idea of dividing the population into discrete farm units, and modelling the epidemics within each farm and the movements of cattle between farms.

For Figure 4.4 we can see that the number of cattle kept in each county varies wildly, with Devon being the largest county and Greater London being one of the smallest. Our chosen case study, Cheshire, is one of the larger ones whilst still being of a manageable size for computations.

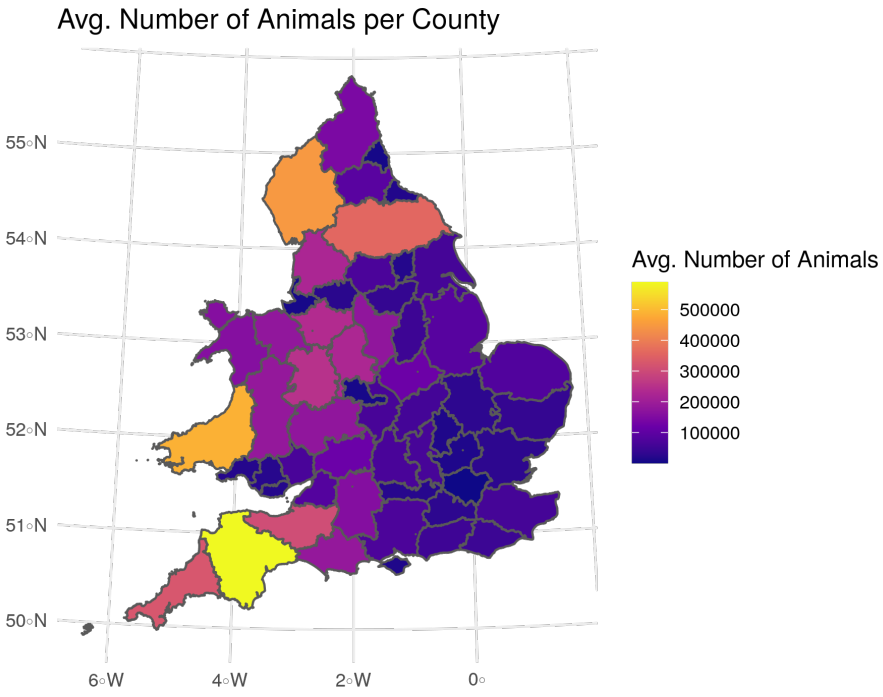


Figure 4.4: Average number of animals per county

This supports the modelling assumption of grouping the population of farms into parishes. Clearly North Yorkshire contains a large number of cattle and the surrounding counties don't, as such there is likely more infectious pressure coming from within North Yorkshire than without, and as such parish grouping makes more sense than geographic grouping for border farms.

As might be expected there is also a seasonality trend in the number of cattle, with more in the summer and less in the winter, in line with deaths. There is also a slow growth in the total number of cattle year on year. We have chosen not to adjust for this yearly seasonality as we do not believe it will have significant effect on the inference.

4.7.2 Movements

Of the animals that moved, Figure 4.5 shows that the majority of animals only move once or twice during their lifetime. These movements in many cases are also movements for slaughter. This has implications for the modelling of the spread of the disease. If movements are generally rare then this will affect how much they contribute to the spread.

There are however a handful of cattle that move a very large number of times, most likely bulls. If the disease were to spread easily this could be a cause for super spreading events, however, its well known that Bovine TB is very slow spreading disease, so this is unlikely to be a factor that needs much consideration. Equally it could contribute to persistence of the disease, or its introduction into new populations, but given the limited time a bull would spend on each farm and the slow spreading nature of disease, plus the small number of cattle that exhibit this movement behaviour, this is also unlikely to be a necessary consideration for our model.

Therefore most of the movements that actually relate the spread of the disease are the singular trade that most moved cattle experience in their life. These still total in the tens of millions, and so it is worth considering each, though as animals typically aren't hoping between farms weekly, it is less likely that we need concern ourselves with super spreaders or pathways. Thus the total counts of movements of animals in each state can just be considered, and in a sense the animals are exchangeable.

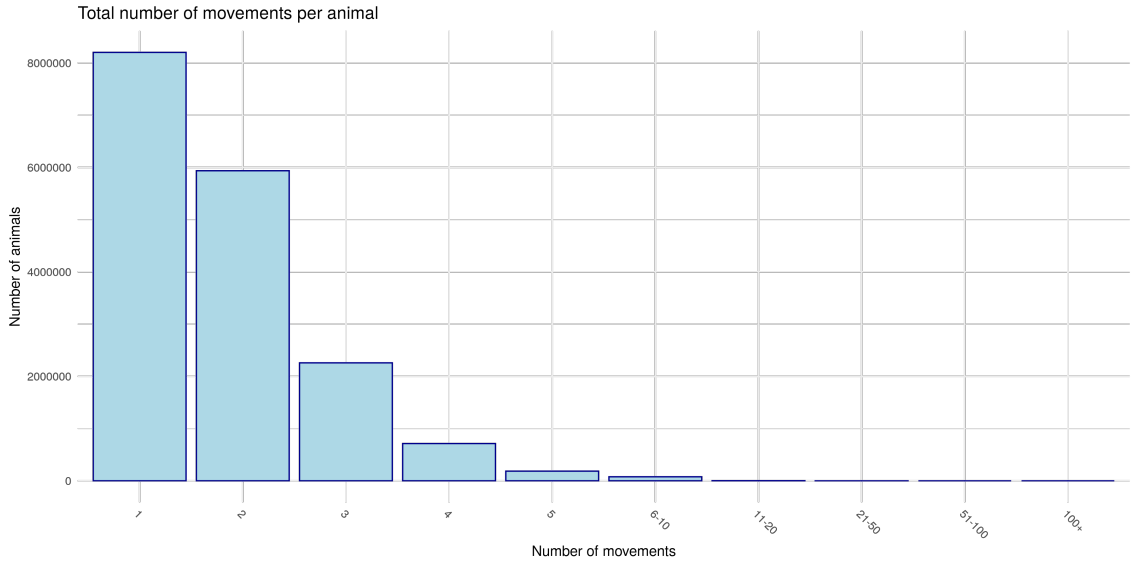


Figure 4.5: Total number of movements per animal

The number of animals moved into and out off each county, as seen in Figures 4.6 and 4.7, is clearly directly proportional to the number of animals within in each county. Rationally the number of movements off should have the same total mass as the number of movements on, however the graph clearly shows far fewer ons than offs. This is because most of the final destinations of cattle, such as slaughter houses and non-agricultural holdings, do not follow the same ID (CPH number for farms) pattern and so the counties of these destinations are not identifiable to us using the key provided by APHA. As such this plot should be interpreted as the number of movements into the county not for slaughter, ie. for animal trade. The bright yellow patch in the North east is North Yorkshire. The movement counts are not scaled by county size or farm density, so this is certainly a contributing factor.

As such the computation burden will scale proportional to the number of cattle in the county for all elements. Thus when considering how to scale the model, or where to apply it, starting with Cheshire is a reasonable action due to its size.

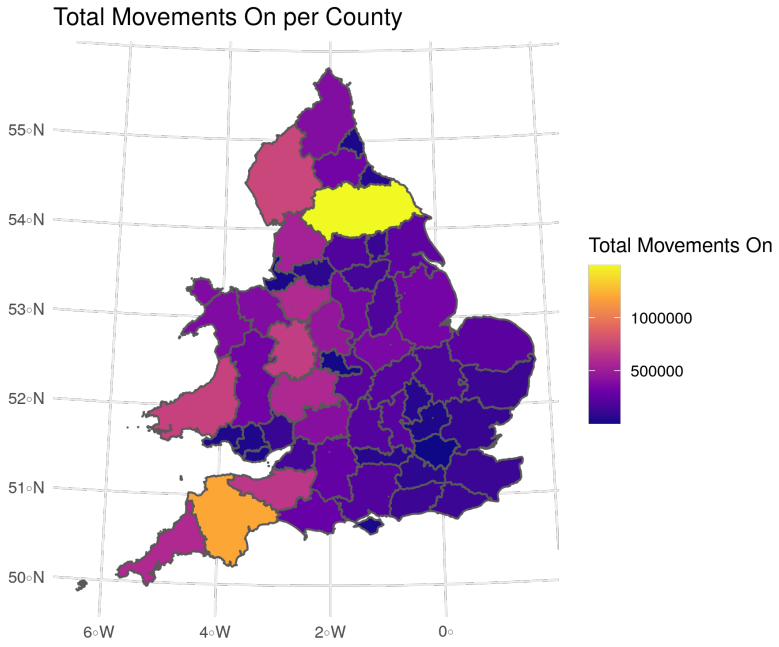


Figure 4.6: Total number of movements in per county

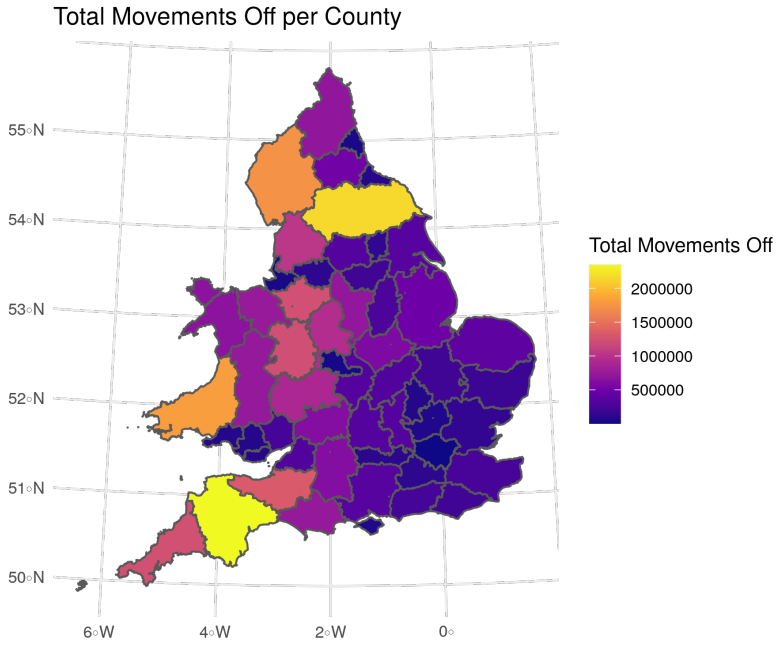


Figure 4.7: Total number of movements out per county

Figure 4.8 shows there is a clear seasonality trend in the movements per month with peaks just before the summer months and just before the Christmas period, though they don't directly align with the seasonality in births and deaths. We don't anticipate this having much effect on our inference, and the patterns seems consistent year on year.

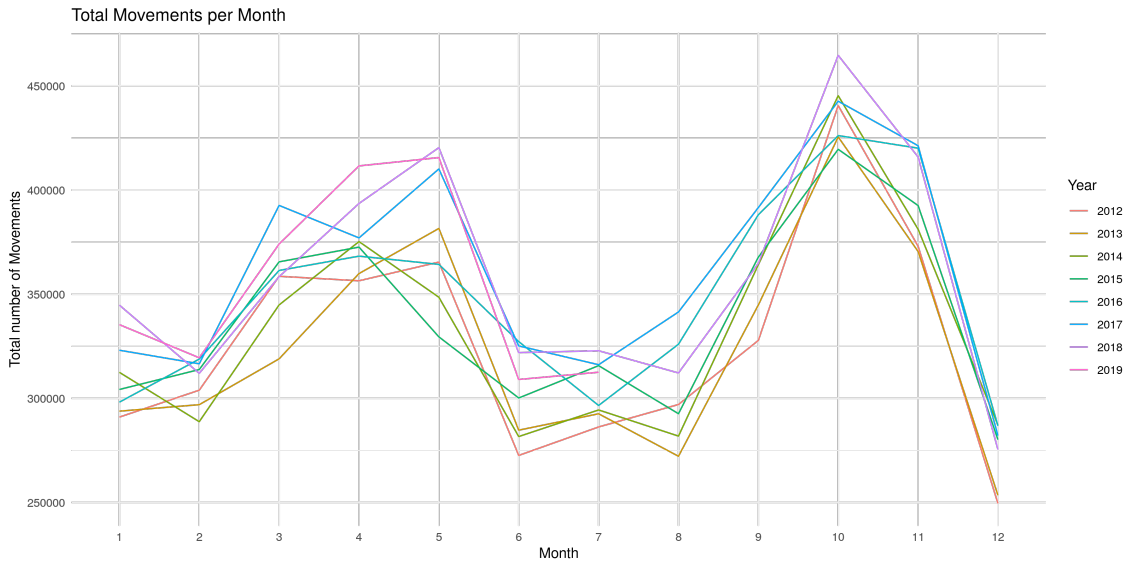


Figure 4.8: Total Movements per Month

Of those movements, we see that movements can happen every day, but there is definitely a strong preference for weekdays as seen in Figure 4.9. There is benefit here to aggregating the weekly level to remove this cycle.

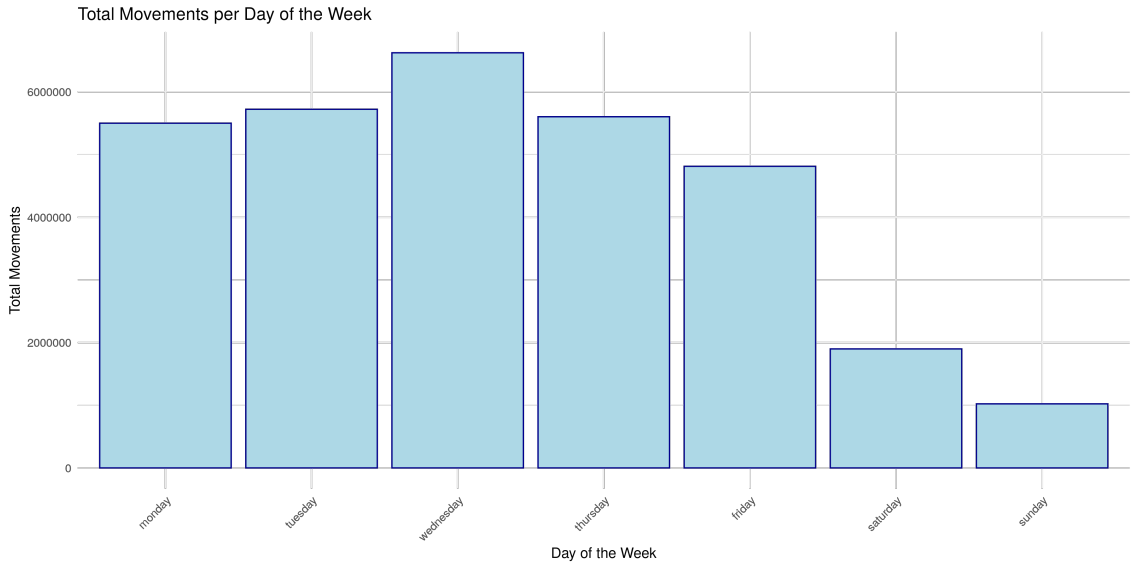


Figure 4.9: Total Movements per Day of the Week

4.7.3 Births and Deaths

The births naturally are proportional to the number of cattle in each county, as are the deaths.

There is clear seasonality in the number of births, with a large peak in March and April, and a small bump again in August and September as seen in 4.10. This is clearly an intentional process by the farms.

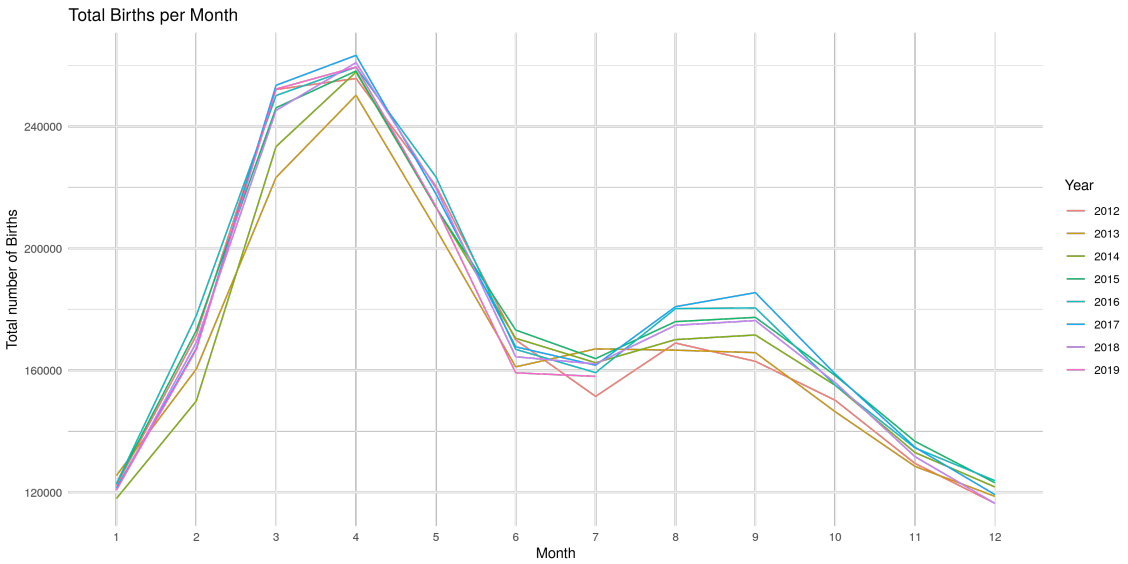


Figure 4.10: Total Births per Month

The deaths per month, as seen in Figure 4.11, however, have the most variability of any of the trends. There tends to be a peak in October or November following a lull in the middle of the year, but the number of deaths can change a lot from month to month. This could be due to changing demand in the population.

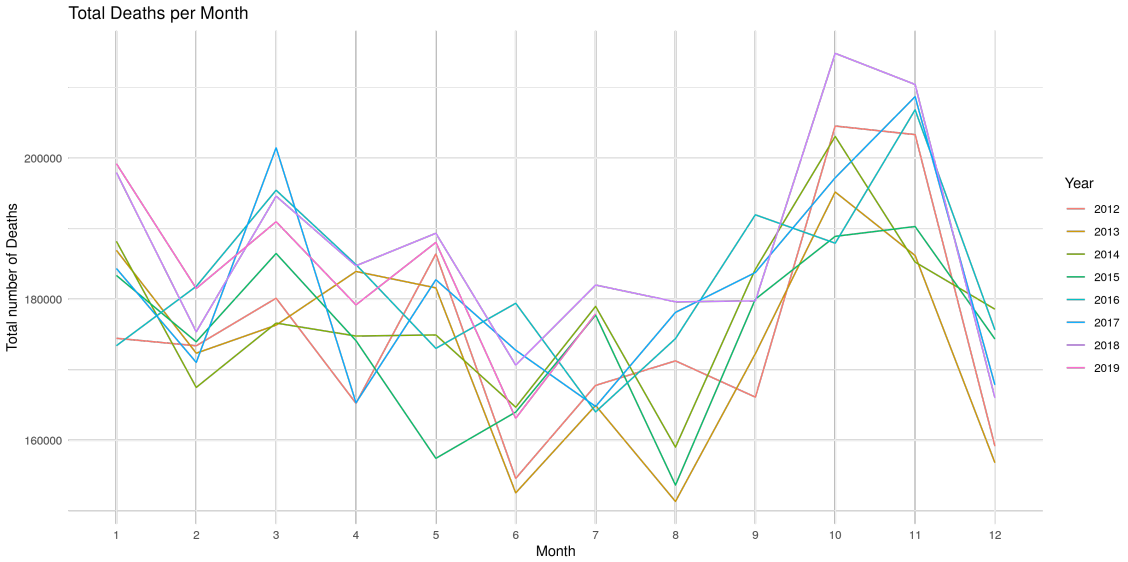


Figure 4.11: Total Deaths per Month

4.7.4 Testing

Seasonality can be seen in the number of cattle tested as well. In Figure 4.12 we can see that this strangely follows the reverse relationship of the number of cattle in the country. The dip mostly happens in the summer months. This could be related to the peak in deaths at the end of the previous year and the peak of births a few months prior combining with testing policy typically not testing cattle less than 6 months old and the reductions in movements. Without further data or understanding of the cattle industry it would not be possible to say what is the causal effect.

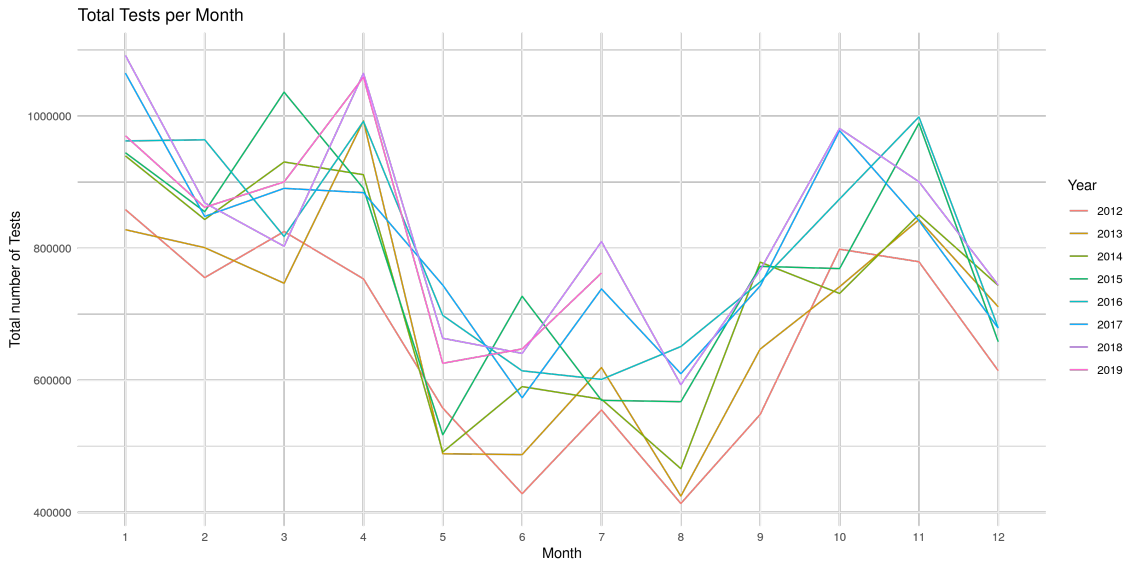


Figure 4.12: Total Tests per Month

From Figure 4.13 we can see that testing mostly occurs on Mondays or Tuesdays. This means aggregating these to one weekly testing event per farm is a reasonable assumption to make for our model, as the majority of other events such as movements and deaths are likely to take place on a different day to testing, before or after during the week.

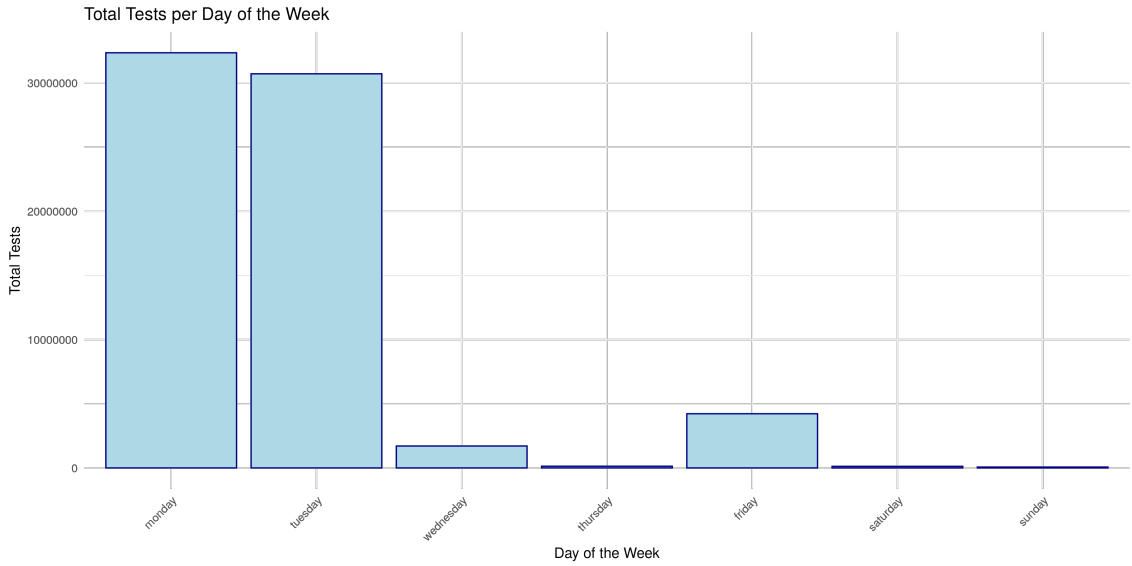


Figure 4.13: Total Tests per Day of the Week

Table 4.13 shows that almost all tests were TB Skin Tests. This justifies the generalising of the testing process in our model to only one kind. Of those 68,426,837 TB Skin Tests 67,986,890 of them were Negative (99.36%), 189,204 were inconclusive (0.28%), and 250,743 were positive (0.37%). This shows the low level of the sensitivity of the TB Skin Test, with almost as many inconclusive tests and positive tests.

Year	ANTIBODY	GAMMA	POSTMORTEXAM	TBSKINTEST
2012	-	30527	1736	7855115
2013	38	36895	1651	8292178
2014	-	72963	1494	8773117
2015	-	87454	1652	9207162
2016	-	90778	1318	9508833
2017	-	126809	1084	9465956
2018	884	234514	1248	9690454
2019	2390	188944	728	5634022
Total	3312	868884	10911	68426837

Table 4.13: Total Tests per Category per Year

4.8 Discussion

In this chapter we have laid out the context of Bovine Tuberculosis, our case study example for large scale complex epidemics. With data spanning 7 years, 20 million cattle, 60 million movements, and 70 million tests, the scale of the epidemic is challenging for even discretised models. Previous work has given insight into dynamics but none have used MCMC.

We have explored data from the APHA to inform the model choices presented in Chapter 5. Based on the number of animals in the population we choose to discretise the population, treating all individuals as identical and exchangeable, and only concerning ourselves with the number of individuals in each infectious state at any given time. Given the different prevalence's, testing policy, and populations across the country we wish to split the population into sub-populations. These could range anywhere from the county to farm level. We have chosen units of the

farm level to align with the reported statistics of herd breakdowns, herd incidence, and herd prevalence, as well as to better model movements. Testing also occurs on a farm-by-farm basis and as such is better analysed with farm units. We concern ourselves with only those premises with the type 'Agricultural Holding', which make up the vast majority. Similarly we only consider whole herd tests done using the TB Skin Test, which accounts for over 90% of tests and over 99% of detections. This reduces the computation burden of 20 million cattle to 80,000 farms. As for the spatial spread of the disease, there will be within farm spread, between farm spread due to movements, and between farm spread due to environment as evidenced in the literature. Brooks-Pollock, Roberts, and Keeling, 2014 choose to have environmental effects within the farm and the parish, however we do not see a reason for a within farm environmental effect and indeed in tests it lead to identifiability issues. With 80,000 farms a spatial kernel that takes into account the distance of between farms is impractical, combined with not having boundary data or data on where cattle are within a farm, we have chosen to have a group of farms share an environmental reservoir of infection, inline with Brooks-Pollock, Roberts, and Keeling, 2014. The default size of a group of farms is a parish, which aligns well with testing policy.

The data is already aggregated at the day level, with some additional data on the order of movements. As such a discrete time model is the natural choice, but the level of discretisation is still up for discussion. The larger the timesteps, the less computational burden, but the more inaccuracies introduced, however it is not a linear scale. Due to the slow spreading nature of the disease, it is reasonable to consider timesteps longer than 1 day. In addition, there are cycles in the movements and testing that can be removed by aggregating to the week level. At this level we can also reduce the number of movements by just considering the first and last location of a cattle each week. This is acceptable because of the unlikeliness of super spreaders and an average of one trade per lifetime per cattle.

With small adjustments we can also focus our analysis on a subset of farms initially to reduce the computational burden, and then scale up the methodology.

Cheshire, or a subset of Cheshire, appears a good choice due to its geographic location in the edge-risk zone, and its medium scale cattle population.

In Chapter 5 following, we introduce our first model for Bovine Tuberculosis based on the findings of this chapter, which utilises a simulated badger population and partially simulated testing to get a partially simulated epidemic with known parameters. From this we develop an MCMC inference methodology and demonstrate its efficiency in this context.

Chapter 5

Our Bovine Tuberculosis Model

5.1 Introduction

In this chapter we demonstrate how we model the spread of a large scale big data epidemic, using full likelihood methods to fit the model to partially simulated data.

Our case study disease is Bovine Tuberculosis (bTB). The challenges are many; the disease has complex dynamics, we have vast quantities of data on all the cattle in England and Wales, and there is a question of the role of Badgers as a reservoir for the disease, which is of particular political interest, but data on this aspect is scarce.

In this chapter we explore how we have processed the cattle movement, testing, birth, and death data into weekly batches, developed a model inspired by that of Brooks-Pollock, Roberts, and Keeling, 2014, simulated an epidemic in this framework guided by data, and developed an MCMC inference scheme.

We begin in Section 5.2 with a description of our model and justify our modelling decisions based on research and our previous data exploration in Section 5.3. In Section 5.4 we provide a glossary of model terms to aid the reader before explaining in Section 5.6 how we simulate new epidemics that are guided by the real data. We present our likelihood and subsequent posterior distributions for this new model in Sections 5.7 and 5.8 respectively. In Section 5.9 we walk through our MCMC

inference schema, with the methods of data augmentation explored in 5.10. Finally we present the result of running this MCMC schema for our simulated data and validate the effectiveness of its inference in Section 5.11, laying the foundations for the following chapter where we apply it to the full real data. In the appendix we present functions and algorithms for MCMC schema.

5.2 Our Bovine Tuberculosis Model

In this section we describe our model for Bovine Tuberculosis in England and Wales. Our model is

- Discrete-time: Each of the processes - epidemic, movement, testing, births, and deaths - is considered at the week level.
- Hierarchical: Due to the discrete time nature of the model, each event set is considered in turn. First the movements, then the infection process in each population, then testing (if it occurs), then births and deaths.
- Meta-population: Each farm is a distinct population of cattle and badgers that generates its own epidemic process. The populations are connected through a network of movements and an environmental reservoir of infection at the parish level (collection of farms).
- Population-level: Within each farm only the counts of cattle in each infectious state on each farm are considered, and the cattle are considered identical and exchangeable.
- SEI(R): The epidemic process within each farm divides the population into 4 groups based on infectious status and models the transitions between the groups.
- Testing: The disease is asymptomatic on the timescales concerned, so a testing process is required to identify infectious cattle and remove them.

- Badgers: There is also a population of badgers on each farm with their own SEIR process independent of the cattle, but also contribute to the background environmental reservoir of infection.

The model is concerned with modelling the spread of Bovine Tuberculosis in England and Wales whilst taking account of the movements of cattle and environmental reservoirs of disease. The model is composed of 4 stochastic processes; the movement process (cattle), the epidemic process (cattle and badger), the testing process (cattle), and the birth and death process (cattle and badger), with the number of cattle movements, births, and deaths being based on data. The processes are parameterised by 9 parameters of interest - 5 are associated with the infection process, 2 with the testing process, and 2 with the badger birth and death process.

Within each farm we have a count of the number of susceptible (S), exposed but not infectious (E), and infectious (I) cattle. We model the transitions between these groups. The rate at which cattle move from susceptible to exposed is given by the infectious pressure on farm i at time t which is calculated using:

$$\lambda^c_{i,t} = \left\{ -\beta_c \frac{x^I_{i,t}}{N^c_{i,t}} - F \frac{V_{p(i),t}}{A_{p(i)}} \right\},$$

where β_c is the within-farm infectious contact rate of cattle, $x^I_{i,t}$ is the number of infectious cattle on farm i at the beginning of time step t, $N^c_{i,t}$ is the total number of cattle, $V_{p(i),t}$ is the environmental pressure on farm i in parish p at time t, $A_{p(i)}$ is the size of the parish p with $i \in p$, and F is the scalar of the environmental reservoir infectious pressure. And similarly the infectious pressure within the badger infectious process is given by

$$\lambda^b_{i,t} = \left\{ -\beta_b \frac{y^I_{i,t}}{N^b_{i,t}} - F \frac{V_{p(i),t}}{A_{p(i)}} \right\},$$

where β_b is the within-farm (assuming one social-group of badgers per farm) infectious contact rate of badgers, $y^I_{i,t}$ is the number of infectious badgers on farm i at the beginning of time step t, and similarly $N^b_{i,t}$ is the total number of badgers.

The rate of transition of cattle from the exposed to infectious state is dependent on the parameter δ_c, and the rate at which badgers move from the exposed state to the infectious state is given by δ_b. We make the assumption that there is no difference between species and that $\delta_c = \delta_b = \delta$. This is reasonable as the literature shows that the badger-to-badger infection rate may be higher due to the confined living situation of badgers in setts, like a household effect, but the long incubation periods are still present, with up to 80% being outwardly asymptomatic (Bhuachalla et al., 2014). We only refer to δ from this point forward.

The environmental reservoir is represented by a scalar in $\mathbb{R}^+$ that represents the background level of infection in each parish contributed to by both cattle and badgers. It represents the bacterial presence of bTB on pastures, in water, in excrement, and so forth, in and around the farm. Each week an amount of additional environmental infectious pressure is generated by the infectious animals in the parish and is added to the current amount, in addition, the current amount decays and is reduced by a random amount. The amount of infectious pressure that remains from one week to the next is dependent on the parameter ϵ, which is the environmental reservoir decay probability. This is inspired by the Brooks-Pollock, Roberts, and Keeling, 2014 model described in Chapter 4, however we have changed it from a deterministic process to a stochastic process, to improve the efficiency of our inference methods (see details in Section 5.6.3.8).

Unlike the typical epidemic models, our epidemic process for Bovine Tuberculosis does not include the natural transition of cattle from the infectious state to the removed state. The disease is chronic and eventually fatal, however due to the government testing programme, the disease is in most cases externally asymptomatic on the timescales we are considering. With this in mind the transition from the infectious to removed state is driven through an independent observation based detection process. An entire herd of cattle is tested for bTB using an insensitive Tuberculin test as described in Chapter 4. The tests occur based on national testing policy, with a frequency of approximately once every 6 - 24 months, with follow up

tests more frequent on farms where bTB is detected. The true sensitivity of the test is unknown and as such we have two parameters that define the process. The first is ρ which is the test sensitivity for cattle in the infectious state. However, cattle in the exposed state are less likely to be detected, thus we suggest a scalar ρ_E between 0 and 1 which generates the detection probability of exposed cattle as $\rho\rho_E$, given they are tested.

Finally, data on badger populations is unavailable at this time, and as such the badger populations we refer to in this chapter are fully simulated. Along side the epidemic process, we also introduce a badger birth and death process to manage the populations. The deaths are unrelated to the epidemic process. We again assume that badgers do not die due to Bovine Tb infection on the scales we are interested in, but die for other reasons before that is possible. The badger rate of birth is given by η_b and the badger rate of death is given by η_d.

Table 5.1 below summarises the parameters of the model.

Parameter	Description
β_c	The within-farm infectious contact rate of cattle
β_b	The within-farm infectious contact rate of badgers
δ	The exposed to infectious transition rate
ϵ	The environmental reservoir decay probability
F	The scalar of the environmental reservoir infectious pressure
ρ	The detection probability for a infectious cattle
ρ_E	The scalar of the detection probability for cattle in the exposed state
η_b	The birth rate of badgers
η_d	The death rate of badgers

Table 5.1: The parameters of interest of our Bovine Tuberculosis model for partially simulated data.

5.2.1 Model updating process

At time t for farm i in parish p, we generate the states of the animals on farm i at $t+1$ in the following way:

1. Use the initial states of the farm at time t to generate the states of the animals that were moved off given the known number of movements.

2. Update the initial states with the movements off of and onto the farm.

3. Use the post movements states to generate the number of newly exposed and infectious animals.

4. Update the states with the newly exposed and infectious animals.

5. If it is a test week, use the post Exposure and Infection states to generate the number of newly detected Exposed and Infectious animals.

6. Update the states with the newly detected Exposed and Infectious animals.

7. Use the post detection states to generate the states of the animals that died of non-testing causes, given the known number of deaths.

8. Finally, update the states with the newly born (all Susceptible) animals and the newly died animals to get the final states.

Figure 5.1 provides a diagram for the typical week on a farm, assuming that testing occurs. Figure 5.2 shows how this farm interacts with the wider parish, including movements and the environmental reservoir effect.

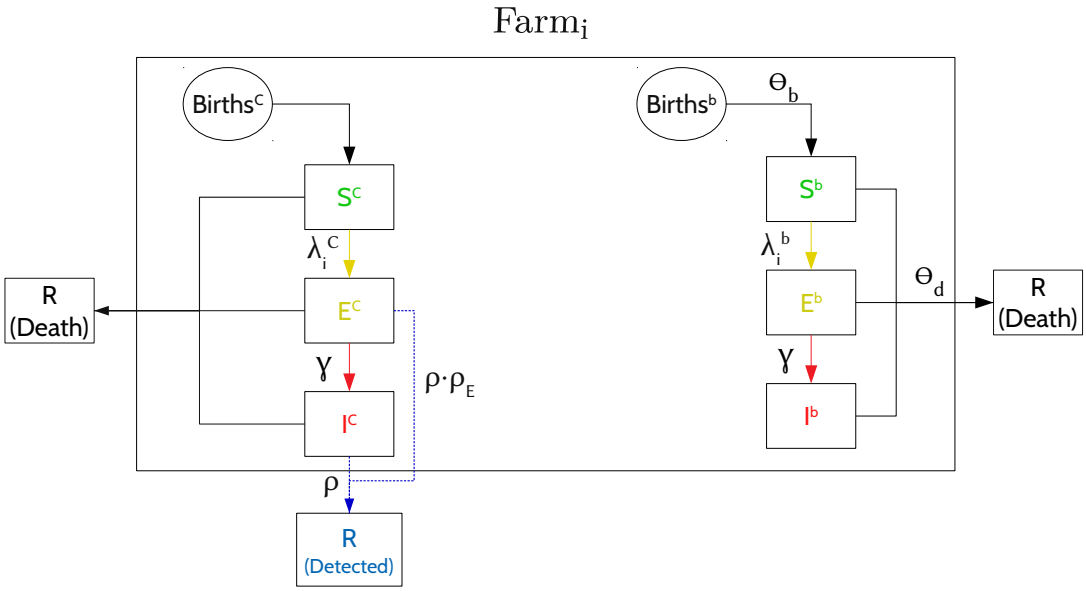


Figure 5.1: A visual representation of the events on one farm for one timestep. The red and yellow arrows relate to the infection process, the dark blue arrows relate to the detection process, and the black arrows relate to non-disease related death. The parameters of the model are detailed in Table 5.1.

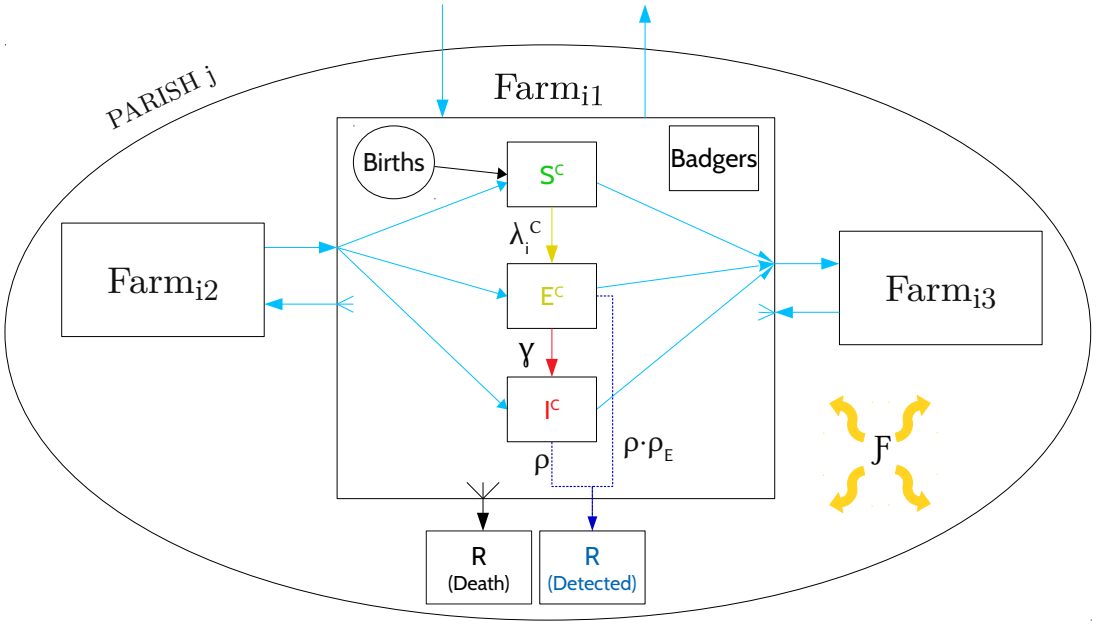


Figure 5.2: A visual representation of the events on one parish for one timestep. The light blue arrows are movements, the red and yellow arrows relate to the infection process, the dark blue arrows relate to the detection process, and the black arrows relate to non-disease related death. The parameters of the model are detailed in Table 5.1.

5.3 Modelling Decisions

The two greatest challenges of modelling an epidemic of this scale are the computational burden of calculating the likelihood, and augmenting the missing data. Our initial choice to address both of these elements is to consider the epidemic as discrete time, aggregating at the week level. This reduces both the size the of the likelihood and the number of nuisance parameters. Alongside this we chose to operate at the farm level, as opposed to an individual level. This makes cattle exchangeable which in many ways simplifies the construction of the likelihood, and allows us to focus on aggregate statistics like counts, rather than the individual states of 20 million cattle. As we are not considering cattle at the individual level we did not take into account any co-variate information such as breed, sex, or genetics.

By aggregating the data and considering the process at the week level, we

have removed the ability to have different processes happening concurrently in our likelihood function. As such we have introduced a hierarchical structure to the events that happen each week. Each week begins with the cattle in an initial set of states; the first events that occur are the movements and create an intermediate set of states. This is followed by the infectious process and further updates to the states, then the testing process and another update if it is a test week, and finally the births and deaths.

Aggregating to week level we simplify the movements such that we only consider the first and last farm a cattle is on each week. This means we do not consider stays on farms for short durations, due to the slow spreading nature of the disease these stays are unlikely to affect the epidemic process. It also addresses weekly cycles in the moving and testing.

Within each farm we chose to implement an SEI process for each cattle population. The majority of cattle in the UK are detected through a testing scheme so we did not consider natural I to R transitions as part of the model. The badger population has their own SEI, with deaths occurring at the their own rate unrelated to the epidemic process, again due to the long time scales of the epidemic and evidence in the literature that suggests the majority of badgers are killed by causes unrelated to the disease (Bhuachalla et al., 2014). The disease is cross-species communicable but we decided not to have the two populations interact directly because of research done by Rosie Woodroffe et al., 2016, which tracked badgers and cattle using GPS collars and showed that badgers prefer cattle pastures but tend to avoid cattle. For this reason, as well as the inspiration of Brooks-Pollock, Roberts, and Keeling, 2014, we introduced an environmental reservoir term at the parish level. This reservoir is contributed to by both infectious badger and cattle populations, and then contributes to the force of infection for both. An additional farm-level environmental reservoir term for each farm had identifiability issues with the parish-level term, and we could not find a strong reason to justify its presence in the model beyond the parish level term. We make the assumption that the

environmental reservoir decays stochastically at some rate each week, but is also added to by any infectious animals in the parish each week.

For the testing, we chose to focus on herd level tests and assume that at all test events every cattle on a farm is tested. This is due to less than 1% of cattle being detected via individual tests. It is known from the literature that the TB Skin Test has imperfect sensitivity and cattle must be infected for a time before the test becomes effective. For this reason we introduced the probability of detecting an infectious cattle upon testing through parameter ρ, and scaled this by ρ_E for exposed cattle.

We made the assumption that cattle births create susceptible cattle, as there is very little evidence that the disease can be transmitted in-utero. We assumed deaths can occur to cattle in any state and are unrelated to disease status or testing, as the disease is asymptomatic on the time scales we are considering.

5.4 Notation

In this section we present a consolidated list of short hand notation used throughout the chapter to represent different states and events associated with the epidemic model. The first table, Table 5.2, presents the notation for the different intermediate sets of states of the cattle on each farm at each time point. Similar notation exist for the badgers. The second table, Table 5.4, presents the notation used for the events associated with the process. These events may be randomly generated as part of the process, or may depend on known data. The purpose of this glossary is to make further explanations, equations, and derivations more compact and readable.

States

Starting Conditions	
$V_{p(i),0}$	The level of infection in the parish environment at time $t = 0$, $i \in p$
$x^S_{i,0}$	The number of Susceptible cattle on farm i at the beginning of the simulation
$x^E_{i,0}$	The number of Exposed cattle on farm i at the beginning of the simulation
$x^I_{i,0}$	The number of Infectious cattle on farm i at the beginning of the simulation
$X_{i,0}$	$[x^S_{i,0}, x^E_{i,0}, x^I_{i,0}]$

Initial States	
$x^S_{i,t}$	The number of Susceptible cattle on farm i at the beginning of time t
$x^E_{i,t}$	The number of Exposed cattle on farm i at the beginning of time t
$x^I_{i,t}$	The number of Infectious cattle on farm i at the beginning of time t
$X_{i,t}$	$[x^S_{i,t}, x^E_{i,t}, x^I_{i,t}]$
$y^S_{i,t}$	The number of Susceptible badgers on farm i at the beginning of time t
$y^E_{i,t}$	The number of Exposed badgers on farm i at the beginning of time t
$y^I_{i,t}$	The number of Infectious badgers on farm i at the beginning of time t
$Y_{i,t}$	$[y^S_{i,t}, y^E_{i,t}, y^I_{i,t}]$

Post Movement States	
$x'^S_{i,t}$	The number of Susceptible cattle on farm i after movements have occurred during time t
$x'^E_{i,t}$	The number of Exposed cattle on farm i after movements have occurred during time t
$x'^I_{i,t}$	The number of Infectious cattle on farm i after movements have occurred during time t
$X'_{i,t}$	$[x'^S_{i,t}, x'^E_{i,t}, x'^I_{i,t}]$

Post Exposures and Infections States	
$x''^S_{i,t}$	The number of Susceptible cattle on farm i after the epidemic process during time t
$x''^E_{i,t}$	The number of Exposed cattle on farm i after the epidemic process during time t
$x''^I_{i,t}$	The number of Infectious cattle on farm i after the epidemic process during time t
$X''_{i,t}$	$[x''^S_{i,t}, x''^E_{i,t}, x''^I_{i,t}]$
$y'^S_{i,t}$	The number of Susceptible badgers on farm i after the epidemic process during time t
$y'^E_{i,t}$	The number of Exposed badgers on farm i after the epidemic process during time t
$y'^I_{i,t}$	The number of Infectious badgers on farm i after the epidemic process during time t
$Y'_{i,t}$	$[y'^S_{i,t}, y'^E_{i,t}, y'^I_{i,t}]$

Table 5.2: The notation for the different intermediate sets of states of the cattle on each farm at each time point.

States

Post Detections States	
$x'''^{S}_{i,t}$	The number of Susceptible cattle on farm i after the testing process during time t
$x'''^{E}_{i,t}$	The number of Exposed cattle on farm i after the testing process during time t
$x'''^{I}_{i,t}$	The number of Infectious cattle on farm i after the testing process during time t
$X'''_{i,t}$	$[x'''^{S}_{i,t}, x'''^{E}_{i,t}, x'''^{I}_{i,t}]$

Post Births and Deaths States	
$x''''^{S}_{i,t}$	The number of Susceptible cattle on farm i after the birth-death process during time t
$x''''^{E}_{i,t}$	The number of Exposed cattle on farm i after the birth-death process during time t
$x''''^{I}_{i,t}$	The number of Infectious cattle on farm i after the birth-death process during time t
$X''''_{i,t}$	$[x''''^{S}_{i,t}, x''''^{E}_{i,t}, x''''^{I}_{i,t}]$
$y''^{S}_{i,t}$	The number of Susceptible badgers on farm i after the birth-death process during time t
$y''^{E}_{i,t}$	The number of Exposed badgers on farm i after the birth-death process during time t
$y''^{I}_{i,t}$	The number of Infectious badgers on farm i after the birth-death process during time t
$Y''_{i,t}$	$[y''^{S}_{i,t}, y''^{E}_{i,t}, y''^{I}_{i,t}]$

Table 5.3: The notation for the different intermediate sets of states of the cattle on each farm at each time point.

Events

Movement Events	
$m_{i,j,t}$	The total number of animals that farm i moved to farm j during time t
Infection Events	
$dE^c_{i,t}$	The number of Exposed cattle that farm i generated during time t
$dI^c_{i,t}$	The number of Infectious cattle that farm i generated during time t
$dE^b_{i,t}$	The number of Exposed badgers that farm i generated during time t
$dI^b_{i,t}$	The number of Infectious badgers that farm i generated during time t
Detection Events	
$H^E_{i,t}$	The number of Exposed cattle that were detected on farm i during time t
$H^I_{i,t}$	The number of Infectious cattle that were detected on farm i during time t
Birth and Death Events	
$b^c_{i,t}$	The total number of cattle births that occurred on farm i during time t
$d^c_{i,t}$	The total number of cattle deaths that occurred on farm i during time t
$D^{cS}_{i,t}$	The number of Susceptible cattle that died on farm i during time t
$D^{cE}_{i,t}$	The number of Exposed cattle that died on farm i during time t.
$D^{cI}_{i,t}$	The number of Infectious cattle that died on farm i during time t.
$D^c_{i,t}$	$[D^{cS}_{i,t},\ D^{cE}_{i,t},\ D^{cI}_{i,t}]$
$B^b_{i,t}$	The total number of badger births (Susceptibles) on farm i during time t
$D^{bS}_{i,t}$	The number of Susceptible badger that died on farm i during time t
$D^{bE}_{i,t}$	The number of Exposed badger that died on farm i during time t.
$D^{bI}_{i,t}$	The number of Infectious badger that died on farm i during time t.
$D^b_{i,t}$	$[D^{bS}_{i,t},\ D^{bE}_{i,t},\ D^{bI}_{i,t}]$

Table 5.4: The notion used for the events associated with the process.

5.5 Computational Considerations

The data in most instances are at the resolution of cow and day. Due to the slow moving nature of the disease, and other aspects of the cattle movement network and

testing schema, a modelling decision has been made to reduce the computational burden of the inference by modelling at the population and discrete-time week level. For this reason we need to aggregate the data. The aggregation depends on the data source, but the aim in each is to have a combined count for each variable of interest for each farm each week. The data provided to us by the APHA includes the details of each cow, a record of every movement for each cow, the details and result of every test done on each cow, and the birth and death date of each cow. Later in this chapter some testing data are replaced with simulated records for the purpose of validating the inference methodology, but are used in Chapter 6 when fitting to the full data. The following section details the data cleaning and aggregation choices made to process the data into this new format.

We used the same week batching schema across all data sources, where weeks run Sunday - Saturday, with the first week starting on Sunday 1st of January 2012. We chose to start weeks on Sunday as it aligns well with all data sources.

5.5.1 Movements

The raw movement data contains one record for each cow (eartag) movement from one farm (CPH) to another, possibly with a transition farm (CPH). For each week, for each farm pair we are interested in identifying how many cattle initialised the week on Farm A and ended the week on Farm B. Due to the slow spreading nature of the disease we have elected to ignore short stays on farms, including in-between destinations and transition farms. We also chose to remove any movements that occurred after a cow was assigned the slaughter action after testing positive, as most of these movements took the cattle outside of Cheshire and would indirectly include data in the simulation that is not intended.

First we removed any identical duplicate records. Next we grouped records by eartag and week, and identified the first CPH that each cattle left that week, and the last CPH that they arrived at, and aggregated these results into one record. Finally we removed any records that did not start or end their weeks movement in

Cheshire, and removed any records that had cattle start and end at the same farm.

After this we grouped by first off CPH, last on CPH, and week, and counted the number of movements between each directional farm pair. Note here A $\rightarrow$ B in week t is a unique record to B $\rightarrow$ A in week t.

The final aggregated data has one record for each A $\rightarrow$ B farm pair that contains the number of cattle that started on A and ended on B that week.

5.5.2 Births

The raw births data is very clean and contains one record per eartag that was born in Cheshire, and the date and week.

We aggregated the data to have each record be the number of cattle born on each farm (CPH) during each week.

5.5.3 Deaths

The raw deaths data is very clean and contains one record per eartag that died in Cheshire, and the date and week. Death could be due to multiple reasons, the main ones being processed for meat and testing positive for Bovine TB.

In our simulation, testing will be a randomly generated process and will not rely on historic data except for the initial conditions. For this reason we first removed any deaths that occur after a cow (eartag) had been assigned the "Slaughter" action following a test.

We then aggregated the data to have each record be the number of cattle that died on each farm (CPH) during each week.

5.5.4 Testing

The raw testing data contains one record for each test applied to a cow (eartag). These tests can vary based on date applied, reason for testing, type of test, outcome of test, action following test, and many other variables. Multiple tests of the same

cattle can occur on the same day such as when multiple different test methods are used.

First we grouped by many variables and removed any identical duplicate tests. The majority of records indicated only one test per cattle per CPH per week. In a small number of cases cattle would have multiple tests. The reasons for this could include a follow up test due to uncertainty in the result, or with a higher sensitivity test. We rationalised that we are mostly interested in how many cattle were tested, and how many were detected. For this reason we reduced these records to one per cow per CPH per week, favouring records that had the "Slaughter" action and used the TB Skin Test method.

After we had one record per cattle per CPH per week, we aggregated the records to count the number of cattle tested, and the number of cattle assigned the slaughter action, on each CPH each week.

5.5.5 Initial Conditions

We will not use the testing data during the simulation, as the epidemic and testing process will be stochastic. We will however use the testing data along with the historic herd data to initialise the process. Testing of farms can occur anywhere between once every 6 months and 4 years, so not all farms have a testing set from near the beginning of the process. Equally, testing of large farms may occur over several weeks, or sporadic tests may be ordered. We needed to choose a way of decided how many cattle are infected on each farm at the beginning of the process.

A number of methods were explored, ranging from simple to complex. The final decision chose a simple transparent solution. After processing the testing data into week batches, we added up the total number of tests and total number of slaughter actions from the first 26 weeks. The test does not have perfect sensitivity, roughly around 70% according to previous studies, but the data consists of a small number of higher sensitivity tests and some cattle may have been tested more than once in those first 6 months. For this reason when estimating the proportion of infected

cattle on each farm we assumed an 80% detection rate.

Finally we multiplied this adjusted proportion infected by historical herd record of the number of cattle on each farm, and rounded up. We made the choice to be conservative and potentially over-estimate the number of initial infected cattle.

5.6 Data Generating Process

In this chapter our aim is to make inference on a partially simulated epidemic dataset in order to build and validate our inference method. In this section we detail how to simulate from the data generating process for our bovine Tuberculosis epidemic model to build our partially simulated dataset, whilst also providing the likelihood term for each subprocess. The overall likelihood is then defined in section 5.7 based on the data generating process.

5.6.1 Initialising the simulation

The first step is to initialise every farm with the number of cattle on the farm at the beginning of the first week, and the number of those cattle that are infectious. We take the historic herd data from each farm in January 2012 as the number of animals. As described in Section 5.5 we estimate the number of infected cattle based on the testing data.

In addition we are also considering a situation with an explicit badger population on each farm. We have no data on the badger populations, so we randomly generate the population of between 10 and 100 susceptible badgers per farm, and a random number of initial infected badgers based on the size of the susceptible badger population and the proportion of infectious cattle.

Then during each timestep we evolve the population of cattle and badgers on each farm using 4 processes; the epidemic process, the movement process, the testing process, and the birth and death process. The epidemic process is fully simulated. In this chapter we replace the observed testing process with a simulated testing

process, but assume the same data is observed. That is we assume all cattle are tested, simulate how many exposed and infectious cattle are detected, and only observe the total number of detections. The birth data comes directly from APHA, and the non-testing related deaths do as well.

For the testing process we only make use of the the initial testing date and its results for each farm provided by APHA to initialise the simulation, and generate future test dates in line with government policy, and simulate the test results. We do this because aligning the data from the testing, movement, and historic herd sources with the stochastic elements of the simulation is a very challenging task, and the testing is easy to simulate. On the initial testing date, we simulate the number of number of detections of cattle in each state, and from then on generate future testing dates based on government policy, assuming all cattle are tested at each timestep.

We use as much of the cattle movement data (m_t) as possible given the stochasticity of the process. How much of the cattle movements can be used depends partially on the testing process. Stochastically generated test results mean that some movements will not occur, as farms that test positive are put under movement restrictions where the APHA did not have them under movement restrictions. Likewise there is a possibility that some farms that did not have moves occur in the real data due to movement restrictions are able to move cattle in this simulated data. We have chosen not to simulate or otherwise adapt the movements to account for this as it introduces uncertainty into the model arising from the simulation choices. Instead this simulated data set can be thought of as having a less dense movement network than the observed APHA data.

The model is parameterised by 9 parameters; $[\beta_c,\ \beta_b,\ \delta,\ F,\ \epsilon]$ relating to the infection process, $[\rho,\ \rho_E]$ relating to the testing process, and $[\eta_b,\ \eta_d]$ relating to the badger birth and death process.

This simulation process will generate a data set that partially contains data from APHA, and partially simulated data, with the noted advantage that we know the

unobserved data values not present in the APHA data, such as the infectious states of all the cattle, and the states of the cattle that are detected during testing and that die, as well as the values of the parameters that generated those data. This allows us to validate the inference method at various levels of data availability, but making different assumptions about what data is observed and what are unobserved latent variables.

5.6.2 Kernels

In this section we define the overall process for simulating from the data generating process in terms of transmission kernels. Transition kernels, K, apply a series of operations to a set of states to produce a new set of states; $K :: X_t \rightarrow X_{t+1}$

The kernels are,

- $K_M :: X \rightarrow X'$ (Cattle Movements)
- $K_E :: X \rightarrow X'$ (Cattle Epidemic)
- $K_T :: X \rightarrow X'$ (Cattle Testing)
- $K_D :: X \rightarrow X'$ (Cattle Births and Deaths)
- $K_I :: Y \rightarrow Y'$ (Badger Epidemic)
- $K_L :: Y \rightarrow Y'$ (Badger Births and Deaths)
- $K_V :: V \rightarrow V'$ (Environmental Reservoir)

In addition there is also the supporting function, P, to calculate event probabilities; $P :: [X_t, Y_t, V_t] \rightarrow Q_t$.

Using these kernels we can now define K_0 to be the kernel for generating a new timestep in the epidemic; $K_0 = K_M \circ K_E \circ K_T \circ K_D \circ K_I \circ K_L \circ K_V :: [X_t, Y_t, V_t] \rightarrow [X_{t+1}, Y_{t+1}, V_{t+1}]$. The details of K_0 are given in Algorithm 11. The details of the other kernels are given in Section 5.6.3.

Algorithm 2: Generate states for timestep $t+1$

Input : X_t = Cattle states at beginning of timestep t,
Y_t = Badger states at beginning of timestep t,
V_t = Environmental reservoir at beginning of timestep t,
θ = Model parameters,
$M_t,\ h_t,\ b_t,\ d_t$ = Data during timestep t.

Output : X_{t+1} = Cattle states at beginning of timestep $t+1$,
Y_{t+1} = Cattle states at beginning of timestep $t+1$,
V_{t+1} = Cattle states at beginning of timestep $t+1$.

Elements: Q_t = Event probabilities during timestep t,
$X',\ X'',\ X''',\ X'''',\ Y',\ Y'',\ V'$ = Intermediate states.

$K_0(X_t, Y_t, V_t)$::
  Probabilities
  $Q_t = P(X_t, Y_t, V_t)$
  Cattle
  $X' = K_M(X_t,\ M_t)$
  $X'' = K_E(X',\ Q_t)$
  $X''' = K_T(X'',\ h_t)$
  $X'''' = K_D(X''', b_t, d_t)$
  Badgers
  $Y' = K_I(Y_t, Q_t)$
  $Y'' = K_L(Y')$
  Environment
  $V' = K_V(V_t, X'''', Y'')$
Return X'''', Y'', V'

5.6.3 Details of Kernels

In this section we provide the context and details of each of the subroutine and transition kernels necessary for Algorithm 11. We also define the likelihood

component for each function.

5.6.3.1 The Probability Function

For each farm, i, we first calculate the exposure and infection probabilities during timestep t given the animal states at beginning of timestep t.

Algorithm 3: Generate event probabilities for timestep t

Input : X_t = Cattle states at beginning of timestep t,
Y_t = Badger states at beginning of timestep t,
V_t = Environmental reservoir at beginning of timestep t,
$[\beta_c, \beta_b, \delta, F] \in \theta$ = Model parameters.

Output : Q_t = Event probabilities during timestep t.

Elements: $p^c_{exp}(i,t), p^b_{exp}(i,t), p_{inf}$ = Event probabilities during timestep t,
ω = The set of farms of interest.

$P(X_t, Y_t, V_t, \beta_c, \beta_b, \delta, F)$**::**
  foreach $i \in \omega$ **do**
    $p^c_{exp}(i,t) = 1 - \exp\left\{-\beta_c \frac{x^I_{i,t}}{N^c_{i,t}} - F\frac{V_{p(i),t}}{A_{p(i)}}\right\}$
    $p^b_{exp}(i,t) = 1 - \exp\left\{-\beta_b \frac{y^I_{i,t}}{N^b_{i,t}} - F\frac{V_{p(i),t}}{A_{p(i)}}\right\}$
  end
  $p_{inf} = 1 - \exp\{-\delta\}$
  $Q_t = [p^c_{exp}(i,t),\ p^b_{exp}(i,t),\ p_{inf}] \ \ \forall \ \ i$
  `Return` Q_t

5.6.3.2 The Cattle Movement Kernel

We know the number of movements off of each farm i during each timestep, and their destination farm j. For the movements off of farm i at time t, given there are enough cattle on farm i, and we know their infectious status, then the simulation process is straightforward, and the subsequent likelihood is simple. The process would be, given m movements off of farm i, decide on the states of those cattle moved, and assign those cattle to their destination farms. For this we use a series of

Multivariate Hypergeometric draws, denoted $\text{MHG}(N, n)$ where N is the population vector and n is the number of draws. However, in reality due to the discretisation and imperfect data, the true subroutine is necessarily more complex.

The full set of movement data has been processed into records containing the number of animals moved from farm i to farm j during time step t, with j to i being a distinct record.

The movements can then be processed into four disjoint sets:

$$M = \{(\omega^c \to \omega^c) \cup (\omega^c \to \omega) \cup (\omega \to \omega) \cup (\omega \to \omega^c)\}.$$

where,

- $\omega^c \to \omega^c$ represent movements from a farm outside our set of interest to another farm outside our set of interest. We don't track these farms so these movements can be ignored.
- $\omega^c \to \omega$ represent movements from outside our set of interest to a farm inside our set of interest. These movements only add animals to our tracked population.
- $\omega \to \omega$ represent movements from inside to inside. These movements maintain the total number of animals in the tracked population.
- $\omega \to \omega^c$ represent movements from inside to outside. These movements remove animals from the tracked population to farms that are untracked.

The movement data is recorded on the day scale at the individual cattle level, with additional data on the order of movements for each cattle. By aggregating to the farm and week level, we lose this implicit ordering. We process all movements from a farm in a timestep at once, however in reality there is a flow of animals between farms during each week. For this reason we will encounter circumstances where the number of animals on a farm at the beginning of a timestep are insufficient to complete its movements - they have to receive animals from other farms first. This

is made worse by artefacts in the data leading to misalignments, some moves not occurring due to movement restrictions resulting from the stochastic testing process in the partially simulated data, and the potentially cascading effect of incomplete movements. Thus our goals when utilising the aggregated movements is to do so in a way that enables the most movements to occur, that is, get the farm populations as large as possible before attempting to move animals off, and utilise intermediary states that allow farms to move animals that moved on during that timestep.

For this reason, we discard the $(\omega^c \to \omega^c)$ movement, process the $(\omega^c \to \omega)$ first, then the $(\omega \to \omega)$ movement, noting that if a movement isn't possible when processing is attempted, to skip it and try it again after more farms have been processed, and them finally process the $(\omega \to \omega^c)$ movements, as these only remove animals, as since ω^c is not tracked and will have the least impact on the simulation if they are unable to occur.

We define M_t to be the full set of movements during time t, with $m_{i,j}(t)$ being the number of animals moved from farm i to farm j during time t.

The kernel can then be written as:

Algorithm 4: Generate event probabilities for timestep t

Input : X_t = Cattle states at beginning of timestep t,
M_t = Movement data during timestep t.
Output : X' = Post movement cattle states during timestep t.
Elements: X^* = Intermediate cattle states,
R = Set of farms that were unable to be processed during the previous loop,
R' = Set of farms that were unable to be processed during the current loop,
ω = The set of farms of interest.

$K_M(X_t,\ M_t)$::
  Step 1: $(\omega^c \to \omega)$
    foreach $i \in \omega^c,\ j \in \omega$ **do**
      $a_{i,j,t} = \text{Multinomial}\left(m_{i,j,t},\ [0.9,\ 0.05,\ 0.05]\right)$
      $X^* = X_t + a_t$
    end
  Step 2: $(\omega \to \omega)$
    Set $R = \varnothing$, $R' = \varnothing$
    for $i \in \omega$ **do**
      if $\sum_{j\in\omega} m_{i,j,t} > N^*_{i,t}$ **then**
        $R' = \{R' \cup i\}$
        next
      else
        $b_{i,t} = \text{MHG}\left([x^{*S}_{i,t},\ x^{*E}_{i,t},\ x^{*I}_{i,t}],\ \sum_{j\in\omega} m_{i,j,t}\right)$
        $x^*_{i,t} = x^*_{i,t} - b_{i,t}$
        for $j \in \omega$ **do**
          $c_{i,j,t} = \text{MHG}\left(b_{i,t},\ m_{i,j,t}\right)$
          $b_{i,t} = b_{i,t} - c_{i,j,t}$
          $x^*_{j,t} = x^*_{j,t} + c_{i,j,t}$
        end
      end
    end
  Step 3: Retry farms that did not work
    while $R \neq R'$ **do**
      Set $R = R'$, $R' = \varnothing$
      for $i \in R$ **do**
        Do **Step 2:** $(\omega \to \omega)$
        and generate x^{**}
        Note: If $R = R' \neq \varnothing$, discard the moves for $i \in R$ as the movements cannot occur.
      end
    end
  Step 4: $(\omega \to \omega^c)$
    for $i \in \omega$ **do**
      if $\sum_{j\in\omega^c} m_{i,j,t} > N^*_{i,t}$ **then**
        skip, and discard the moves for $i \in \omega, j \in \omega^c$
      else
        $d_{i,t} = \text{MHG}\left([x^{**S}_{i,t},\ x^{**E}_{i,t},\ x^{**I}_{i,t}],\ \sum_{j\in\omega^c} m_{i,j,t}\right)$
        $x^{**}_{i,t} = x^{**}_{i,t} - d_{i,t}$
        for $j \in \omega$ **do**
          $e_{i,j,t} = \text{MHG}\left(d_{i,t},\ m_{i,j,t}\right)$
          $d_{i,t} = d_{i,t} - e_{i,j,t}$
          $x^{**}_{j,t} = x^{**}_{j,t} + e_{i,j,t}$
        end
      end
    end
  $X' = x^{**}$
  Return X'

The number of cattle moved off of each farm i at time t, and their destination farm j, is set by the data. The random variable we are concerned with is the states of the animals moved.

The movements are generated in sequence, with movements potentially depending on those that have been processed before in the same timestep. However, if we know the initial states of the system at the beginning of the timestep, and the details of every event, then we also know the states that generated every event. If we know the state of each farm when its movement events were generated, then the likelihood of this process simplifies to a series of Multinomial and Multivariate HyperGeometric distributions,

$$\begin{aligned}\mathbb{P}\left[M_{i,t} \mid X_{i,t}\right] = \frac{m_{-1,i,t}!}{a^S_{-1,i,t}!\; a^E_{-1,i,t}!\; a^I_{-1,i,t}!} \cdot (0.9)^{a^S_{-1,i,t}} \cdot (0.05)^{a^E_{-1,i,t}} \cdot (0.05)^{a^I_{-1,i,t}} \\ \times \frac{\binom{x^{*S}_{i,t}}{b^S_{i,t}}\binom{x^{*E}_{i,t}}{b^E_{i,t}}\binom{x^{*I}_{i,t}}{b^I_{i,t}}}{\binom{x^{*S}_{i,t}+x^{*E}_{i,t}+x^{*I}_{i,t}}{\sum_{j\in\omega} m_{i,j,t}}} \times \prod_{j\in\omega} \frac{\binom{b^S_{i,t}}{c^S_{i,t}}\binom{b^E_{i,t}}{c^E_{i,t}}\binom{b^I_{i,t}}{c^I_{i,t}}}{\binom{b^S_{i,t}+b^E_{i,t}+b^I_{i,t}}{m_{i,j,t}}} \\ \times \frac{\binom{x^{**S}_{i,t}}{d^S_{i,t}}\binom{x^{**E}_{i,t}}{d^E_{i,t}}\binom{x^{**I}_{i,t}}{d^I_{i,t}}}{\binom{x^{**S}_{i,t}+x^{**E}_{i,t}+x^{**I}_{i,t}}{\sum_{j\in\omega^c} m_{i,j,t}}} \times \prod_{j\in\omega^c} \frac{\binom{d^S_{i,t}}{e^S_{i,t}}\binom{d^E_{i,t}}{e^E_{i,t}}\binom{d^I_{i,t}}{e^I_{i,t}}}{\binom{d^S_{i,t}+d^E_{i,t}+d^I_{i,t}}{m_{i,j,t}}}\end{aligned}$$

The assumption for 5% of the external population being in the exposed and infectious state respectively is derived from the average herd breakdowns across the country. Small changes in this value is unlikely to have much impact, as if the values are a bit low for instance, then the overall likelihood will prefer draws with more exposed and infected, and vice versa.

5.6.3.3 The Cattle Epidemic Kernel

Using the post movement states and the epidemic process probabilities calculated using the initial states, we can generate the number of new exposed and infectious cattle.

Algorithm 5: Generate cattle epidemic events for timestep t

Input : X' = Post movement cattle states during timestep t,
Q_t = Event probabilities during timestep t.

Output : X'' = Post infection process cattle states during timestep t.

Elements: $dE^c_{i,t}$ = S to E events on farm i during timestep t,
$dI^c_{i,t}$ = E to I events on farm i during timestep t,
ω = The set of farms of interest.

$K_E(X', Q_t)$::
  foreach $i \in \omega$ **do**
    $dE^c_{i,t} = \text{Bin}\left(x'^S_{i,t}, p^c_{exp}(i,t)\right)$
    $dI^c_{i,t} = \text{Bin}\left(x'^E_{i,t}, p_{inf}\right)$
    $X'' = X' + [-dE^c_{i,t},\ dE^c_{i,t} - dI^c_{i,t},\ dI^c_{i,t}]$
  end
  Return X''

The likelihood for this subprocess that generates the number of newly exposed and infectious cattle on farm i at time t has the form of two binomial draws,

$$\mathbb{P}\left[dE^c_{i,t},\ dI^c_{i,t} \mid X',\ Q_t\right] = \binom{x'^S_{i,t}}{dE^c_{i,t}} \cdot (p^c_{exp}(i,t))^{dE^c_{i,t}} \cdot (1 - p^c_{exp}(i,t))^{x'^S_{i,t} - dE^c_{i,t}} \times \binom{x'^E_{i,t}}{dI^c_{i,t}} \cdot ({p_{inf}}^{dI^c_{i,t}}) \cdot (1 - p_{inf})^{x'^E_{i,t} - dI^c_{i,t}}$$

where

$$p^c_{exp}(i,t) = 1 - \exp\left\{-\beta_c \frac{x^I_{i,t}}{N^c_{i,t}} - F\frac{V_{p(i),t}}{A_{p(i)}}\right\}$$

is the probability of a susceptible cattle transitioning to the exposed state, and

$$p_{inf} = 1 - exp\{-\delta\}$$

is the probability of an exposed cattle transitioning to the infectious state.

5.6.3.4 The Cattle Testing Kernel

The initial test week for each farm is set based on the data. After this, if infected animals were detected, the next test week is set for 4 weeks time and movement restrictions are put in place, otherwise the next test week is set for 24 weeks time. Given a test week occurs, the whole herd is assumed to be tested, and the number of cattle in the exposed and infectious states, respectively, that are detected are drawn from Binomial distributions with the appropriate detection rate parameters.

Algorithm 6: Generate cattle detection states for timestep t

Input : X'' = Post infection process cattle states during timestep t,
$[\rho, \rho_E] \in \theta$ = Model parameters.

Output : X''' = Post testing process cattle states during timestep t.

Elements: $H_{i,t}$ = Cattle detection states during timestep t,
ω = The set of farms of interest.

$K_T(X'', \rho, \rho_E)$::
  foreach $i \in \omega$ **do**
    $H^E_{i,t} = \text{Bin}\left(x''^E_{i,t},\ \rho\rho_E\right)$
    $H^I_{i,t} = \text{Bin}\left(x''^I_{i,t},\ \rho\right)$
    $X''' = X'' - [0,\ H^E_{i,t},\ H^I_{i,t}]$
  end
  `Return` X'''

The likelihood for this subprocess that generates the number of animals that are detected from each state on farm i at time t has the form of a Multivariate Hyper-Geometric. The test days are known, but the number of animals that are detected from each state are unknown and generated from binomial distributions. The probability is only valid on days when tests occur, as we assume all cattle present on the farm at the time of testing are tested once. Exposed cattle are detected at rate $\rho \times \rho_E$ and infectious cattle are detected at rate ρ. So the probability of the detections is given by:

$$\mathbb{P}\left[H^E_{i,t},\ H^I_{i,t} \mid X'',\ \rho,\ \rho_E\right] = \left[\binom{x''^E_{i,t}}{H^E_{i,t}} \cdot (\rho\rho_E)^{H^E_{i,t}} \cdot (1-\rho\rho_E)^{x''^E_{i,t}-H^E_{i,t}} \times \binom{x''^I_{i,t}}{H^I_{i,t}} \cdot (\rho)^{H^I_{i,t}} \cdot (1-\rho)^{x''^I_{i,t}-H^I_{i,t}} \right]^{\mathbb{1}_{i,t}}$$

The indicator function, $\mathbb{1}_{i,t}$, here is equal to 1 if it is a test day on farm i, and 0 otherwise.

5.6.3.5 The Cattle Births and Deaths Kernel

The number of cattle births and cattle deaths are taken directly from the data. All cattle births produce new susceptible cattle. Cattle deaths can come from any of the states on the farm, and as such are drawn from a Multivariate-Hypergeometric distribution.

Algorithm 7: Generate cattle death states for timestep t

Input : X''' = Post detection cattle states during timestep t,
$b^c_{i,t}$ = Total number of cattle births on farm i during timestep t,
$d^c_{i,t}$ = Total number of deaths on farm i during timestep t.

Output : X'''' = Post births and deaths cattle states during timestep t.

Elements: $D_{i,t}$ = Cattle states for deaths during timestep t,
ω = The set of farms of interest.

$K_D(X''',\ b^c_t,\ d^c_t)$**::**
  foreach $i \in \omega$ **do**
    $D^c_{i,t} = [D^{cS}_{i,t},\ D^{cE}_{i,t},\ D^{cI}_{i,t}] = \text{MHG}\left([x'''^S_{i,t},\ x'''^E_{i,t},\ x'''^I_{i,t}],\ d^c_{i,t}\right)$
    $X'''' = X''' + [b^c_t,\ 0,\ 0] - [D^{cS}_t,\ D^{cE}_t,\ D^{cI}_t]$
  end
  `Return` X''''

The likelihood for this subprocess that generates the number and states of cattle births and deaths on farm i at time t has the form of a Multivariate Hyper-Geometric. The cattle births are known. The number of cattle deaths, $d^c_{i,t}$, are known but not the states of the animals that died. The probability of the observed

death states is given by

$$\mathbb{P}\left[D^c_{i,t} \mid X''',\ b^c_t,\ d^c_t\right] = \frac{\binom{x'''^S_{i,t}}{D^{cS}_{i,t}}\binom{x'''^E_{i,t}}{D^{cE}_{i,t}}\binom{x'''^I_{i,t}}{D^{cI}_{i,t}}}{\binom{N'''^c_{i,t}}{d^c_{i,t}}}.$$

5.6.3.6 The Badger Epidemic Kernel

Using the initial badger states and the calculated epidemic transition probabilities, we can generate the number of new exposed and infectious badgers.

Algorithm 8: Generate badger epidemic events for timestep t

Input : Y_t = Badger states at beginning of timestep t,
Q_t = Event probabilities during timestep t.

Output : Y' = Post infection events badger states during timestep t.

Elements: $dE^b_{i,t}$ = S to E events on farm i during timestep t,
$dI^b_{i,t}$ = E to I events on farm i during timestep t,
ω = The set of farms of interest.

$K_I(Y_t, Q_t)$::
  foreach $i \in \omega$ **do**
    $dE^b_{i,t} = \text{Bin}\left(y^S_{i,t}, p^b_{exp}(i,t)\right)$
    $dI^b_{i,t} = \text{Bin}\left(y^E_{i,t}, p_{inf}\right)$
    $Y' = Y_t + [-dE^b_{i,t},\ dE^b_{i,t} - dI^b_{i,t},\ dI^b_{i,t}]$
  end
  Return Y'

The likelihood for this subprocess that generates the number of newly exposed and infectious badgers on farm i at time t has the form of two binomial draws,

$$\mathbb{P}\left[dE^b_{i,t},\ dI^b_{i,t} \mid Y_t,\ Q_t\right] = \binom{y^S_{i,t}}{dE^b_{i,t}} \cdot (p^b_{exp}(i,t))^{dE^b_{i,t}} \cdot (1 - p^b_{exp}(i,t))^{y^S_{i,t} - dE^b_{i,t}} \times \binom{y^E_{i,t}}{dI^b_{i,t}} \cdot (p_{inf})^{dI^b_{i,t}} \cdot (1 - p_{inf})^{y^E_{i,t} - dI^b_{i,t}}$$

where

$$p^b_{exp}(i,t) = 1 - \exp\left\{-\beta_b \frac{y^I_{i,t}}{N^b_{i,t}} - F\frac{V_{p,t}}{A_{p(i)}}\right\}$$

is the probability of a susceptible badgers transitioning to the exposed state, and

$$p_{inf} = 1 - exp\{-\delta\}$$

is the probability of an exposed badgers transitioning to the infectious state.

5.6.3.7 The Badger Births and Deaths Kernel

We have no data on the badger birth and death process, so we can generate events based on the current population of badgers on each farm. The births are always susceptible and are generated from a Poisson distribution parameterised by the number of badgers and the badger birth rate, η_b. The deaths can come from any state, but the number is unknown, so we generate the number of deaths in each state through Binomial distributions parameterised by the badger death rate η_d. This is an simplification and in reality it is likely that both processes are seasonal - for instance the birth rate would be noticeably higher in spring.

Algorithm 9: Generate badger birth and death events for timestep t

Input : Y' = Post infection process badger states during timestep t,
$[\eta_b, \eta_d] \in \theta$ = Model parameters.

Output : Y'' = Post births and deaths badger states during timestep t.

Elements: $B_{i,t}$ = Number of badger births on farm i during timestep t,
$D_{i,t}$ = States of badger deaths on farm i during timestep t,
ω = The set of farms of interest.

$K_L(Y')$::
  foreach $i \in \omega$ **do**
    $B^b_{i,t} = \text{Po}\left((y'^S_{i,t} + y'^E_{i,t} + y'^I_{i,t}) \times \eta_b\right)$
    $D^{bS}_{i,t} = \text{Bin}\left(y'^S_{i,t},\ \eta_d\right)$
    $D^{bE}_{i,t} = \text{Bin}\left(y'^E_{i,t},\ \eta_d\right)$
    $D^{bI}_{i,t} = \text{Bin}\left(y'^I_{i,t},\ \eta_d\right)$
  end
  $Y'' = Y' + [B^b_t,\ 0,\ 0] - [D^{bS}_t,\ D^{bE}_t,\ D^{bI}_t]$
  `Return` Y''

The likelihood for this subprocess that generates the number and states of badger births and deaths on farm i at time t has the form of a Poisson distribution multiplied by three Binomial distributions. All badger births and deaths are unknown.. The probability of the observed births and death events is given by

$$\begin{aligned}\mathbb{P}\left[B^b_{i,t},\ D^b_{i,t} \mid Y'\right] = &\frac{e^{-\left((y'^S_{i,t}+y'^E_{i,t}+y'^I_{i,t})\times\eta_b\right)} \cdot \left((y'^S_{i,t} + y'^E_{i,t} + y'^I_{i,t}) \times \eta_b\right)^{B^b_{i,t}}}{B^b_{i,t}!} \\ &\times \binom{y'^S_{i,t}}{D^{bS}_{i,t}} \cdot {\eta_d}^{D^{bS}_{i,t}} \cdot (1-\eta_d)^{y'^S_{i,t} - D^{bS}_{i,t}} \\ &\times \binom{y'^E_{i,t}}{D^{bE}_{i,t}} \cdot {\eta_d}^{D^{bE}_{i,t}} \cdot (1-\eta_d)^{y'^E_{i,t} - D^{bE}_{i,t}} \\ &\times \binom{y'^I_{i,t}}{D^{bI}_{i,t}} \cdot {\eta_d}^{D^{bI}_{i,t}} \cdot (1-\eta_d)^{y'^I_{i,t} - D^{bI}_{i,t}}\end{aligned}$$

5.6.3.8 The Environmental Kernel

Finally once the states of all farms have been updated to the final set for that time step, we calculate the change in the background infection environmental parish reservoir for each parish. The environmental reservoir is an integer that is then scaled by the size of the parish. In each time-step the environmental reservoir, represented by a cumulative sum of infectious pressure, will decay and be reduced, and will be added to by the current infected animals. We generate how much of the current time steps environmental pressure will remain to the next time step by drawing the remaining pressure from a Binomial distribution with the $n =$ (The current environmental pressure) and $p = 1-$(the decay rate), which is then scaled by the size of the parish again. The new pressure is drawn from a Poisson distribution with mean equal to the total number of infectious cattle and badgers in the parish, which is then scaled by the size of the parish again, and added to the remaining pressure.

Thus, this stochastic environmental pressure, $V_{p(i),t}$, is (approximately) the deterministic environmental pressure of Brooks-Pollock, Roberts, and Keeling, 2014, multiplied by the parish population. We made these changes for two primary reasons. The first is that a stochastic model allows for a more efficient MCMC algorithm. In a deterministic setting, a single change to the environmental pressure at one farm at one time-step, would lead to a cascading effect that required the recalculation of the entire likelihood. A stochastic process allows one to alter the value of the environmental pressure at a time point and reflect that as a different random draw on the event, thus only needing to recalculate a small section of the likelihood. The second reason follows from the first, in that pulling out one over the size of the parish as a common factor makes generating the events and calculating the likelihood simpler.

Algorithm 10: Generate the environmental reservoir for timestep $t+1$

Input : X'''' = Post births and deaths cattle states during timestep t,
Y'' = Post births and deaths badger states during timestep t,
V_t = Environmental reservoir at beginning of timestep t,
$\epsilon \in \theta$ = Model parameters.

Output : V' = Environmental reservoir at the end of timestep t.

Elements: $R_{p,t}$ = Remaining environmental pressure on farm i at the end of timestep t,
$N_{p,t}$ = New environmental pressure on farm i at the end of timestep t,
ω = The set of farms of interest, with p being the parish groupings.

$K_V(V_t,\ X'''',\ Y'')$::
  foreach $p \in \omega$ **do**
    $R_{p,t} = \text{Bin}\left(V_{p,t},\ 1-\epsilon\right)$
    $N_{p,t} = \text{Po}\left(\sum_{i \in p} X''''_{i,t} + \sum_{i \in p} Y''_{i,t}\right)$
  end
  $V' = R_t + N_t$
  `Return` V'

The likelihood for this subprocess that generates the infectious pressure in the environmental reservoir for parish p at time t has the form of a Poisson distribution multiplied by a Binomial distribution. The environmental reservoir is generated by choosing a random amount to remain from the previous time step based on the decay rate, and a random amount to be added based on the infectious animals in the parish. The probability of the environmental reservoir pressure is thus given by

$$\mathbb{P}\left[V_{p(i),(t+1)}|\ V_{p(i),t},\ X'''',\ Y''\right] = \frac{e^{-\left(\sum_{i\in p} X''''_{i,t}+\sum_{i\in p} Y''_{i,t}\right)} \cdot \left(\sum_{i\in p} X''''_{i,t} + \sum_{i\in p} Y''_{i,t}\right)^{R_{p(i),t}}}{R_{p(i),t}!} \times \binom{V_{p(i),t}}{N_{p(i),t}} \cdot (1-\epsilon)^{N_{p(i),t}} \cdot (\epsilon)^{V_{p(i),t}-N_{p(i),t}}$$

It is worth noting that all the farms i in parish p share the value of $V_{p(i),t}$.

5.7 Likelihood

The likelihood of the epidemic defines the probability of observing the given states of the cattle. For all farms i over times t, assuming that all the data are observed, the likelihood is given by:

$$\prod_{t=1}^{T}\left[\prod_{i=1}^{N_H}\mathcal{L}\Big(X_{i,(t+1)}\ Y_{i,(t+1)},\ V_{p(i),(t+1)}\mid X_{i,t},\ Y_{i,t},\ V_{p(i),t}\ [\beta_c,\beta_b,\delta,\epsilon,F,\rho,\rho_E,\eta_b,\eta_d],\ m_{i,t},\ test_{i,t},\ d^c_{i,t}\Big)\right],$$

where N_H is the total number of farms, *test* is *true* if testing occurred during timestep t and $false$ otherwise, and all other parameters can be found in the glossary in Section 5.2.

We can represent the likelihood here in terms of the number of transitions between each of the state pairs at timestep t, given the initial state.

Let $\boldsymbol{\theta}$ denote the parameters of the model. The likelihood above is also equivalent to,

$$= \mathcal{L}\Big(\mathbf{M},\ \mathbf{dE}^c,\ \mathbf{dI}^c,\ \mathbf{dE}^b,\ \mathbf{dI}^b,\ \mathbf{H}^E,\ \mathbf{H}^I,\ \mathbf{D}^c,\ \mathbf{B}^b,\ \mathbf{D}^b,\ \mathbf{V}\mid \mathbf{X}_0,\ \mathbf{Y}_0,\ \mathbf{V}_0,\ \boldsymbol{\theta},\ \mathbf{m},\ \mathbf{test},\ \mathbf{b}^c,\ \mathbf{d}^c\Big).$$

That is, the joint likelihood of all the events from $t = 1 : T$, given the initial states and the counts of each event.

5.7.1 The form of the likelihood

The likelihood, assuming all data is known, can thus be broken down as the product over all timesteps $t \in 1:T$ for:

$$\begin{aligned}
&\mathcal{L}\Big(\mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{D}^c{}_t,\ \mathbf{B}^b{}_t,\ \mathbf{D}^b{}_t,\ \mathbf{V}_{t+1} \mid \\
&\qquad \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t,\ \boldsymbol{\theta},\ \mathbf{m}_t,\ \mathbf{test}_t,\ \mathbf{b}^c_t,\ \mathbf{d}^c_t\Big) \\
&\quad = \mathbb{P}\left[\mathbf{M}_t \mid X_t,\ \mathbf{m}_t\right] \\
&\qquad \times \mathbb{P}\left[\mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t \mid \mathbf{M}_t,\ X_t,\ Y_t,\ \mathbf{V}_t,\ \boldsymbol{\theta}\right] \\
&\qquad \times \mathbb{P}\left[\mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t \mid \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t,\ \mathbf{test}_t,\ X_t,\ Y_t,\ \mathbf{V}_t,\ \boldsymbol{\theta}\right] \\
&\qquad \times \mathbb{P}\left[\mathbf{D}^c{}_t,\ \mathbf{B}^b{}_t,\ \mathbf{D}^b{}_t \mid \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{d}^c_t,\ X_t,\ Y_t,\ \mathbf{V}_t,\ \boldsymbol{\theta}\right] \\
&\qquad \times \mathbb{P}\left[\mathbf{V}_{(t+1)} \mid \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{D}^c{}_t,\ \mathbf{B}^b{}_t,\ \mathbf{D}^b{}_t,\ X_t,\ Y_t,\ \mathbf{V}_t,\ \boldsymbol{\theta}\right]
\end{aligned}$$

5.8 Posteriors

Using the likelihood derived in Section 5.7 we can derive the conditional posterior distribution, assuming all data is observed, used in the MCMC algorithm explored in Section 5.9 by specifying priors for the parameters.

The full joint conditional posterior is given by

$$\begin{aligned}
\pi\left(\beta_c, \beta_b, \delta, \epsilon, F, \rho, \rho_E, \eta_b, \eta_d \mid \mathbf{X}_t, \mathbf{Y}_t, \mathbf{V}_t\right) = &\mathcal{L}\Big(\mathbf{M},\ \mathbf{dE}^c,\ \mathbf{dI}^c,\ \mathbf{dE}^b,\ \mathbf{dI}^b,\ \mathbf{H}^E,\ \mathbf{H}^I,\ \mathbf{D}^c,\ \mathbf{B}^b,\ \mathbf{D}^b,\ \mathbf{V} \mid \\
&\qquad \mathbf{X}_0,\ \mathbf{Y}_0,\ \mathbf{V}_0,\ \boldsymbol{\theta},\ \mathbf{m},\ \mathbf{test},\ \mathbf{b}^c,\ \mathbf{d}^c\Big) \\
&\times \pi\left(\beta_c\right) \times \pi\left(\beta_b\right) \times \pi\left(\delta\right) \times \pi\left(\epsilon\right) \times \pi\left(F\right) \\
&\times \pi\left(\rho\right) \times \pi\left(\rho_E\right) \times \pi\left(\eta_b\right) \times \pi\left(\eta_d\right)
\end{aligned}$$

where we are setting the priors to be:

- $\pi(\beta_c) \sim \mathsf{Gamma}(\vartheta_{\beta_c}, \sigma_{\beta_c})$
- $\pi(\beta_b) \sim \mathsf{Gamma}(\vartheta_{\beta_b}, \sigma_{\beta_b})$
- $\pi(\delta) \sim \mathsf{Gamma}(\vartheta_{\delta}, \sigma_{\delta})$
- $\pi(\epsilon) \sim \mathsf{Gamma}(\vartheta_{\epsilon}, \sigma_{\epsilon})$
- $\pi(F) \sim \mathsf{Gamma}(\vartheta_{F}, \sigma_{F})$
- $\pi(\rho) \sim \mathsf{Beta}(\vartheta_{\rho}, \sigma_{\rho})$
- $\pi(\rho_E) \sim \mathsf{Beta}(\vartheta_{\rho_E}, \sigma_{\rho_E})$
- $\pi(\eta_b) \sim \mathsf{Gamma}(\vartheta_{\eta_b}, \sigma_{\eta_b})$
- $\pi(\eta_d) \sim \mathsf{Beta}(\vartheta_{\eta_d}, \sigma_{\eta_d})$

The form of the Gamma distribution has $\vartheta > 0$ as the shape parameter, and $\sigma > 0$ as the rate parameter. The gamma priors align with the positive real support of the parameters, whilst also being malleable to adapt to weak or strong prior knowledge. The detection parameters, ρ (a probability) and ρ_E (a scalar in [0,1]), both have support on the real line between 0 and 1, so a Beta prior is an appropriate choice, combined with it's malleability. The choice of Gamma and Beta prior for the badger birth and death rate respectively is due to being conjugate priors for the likelihood terms, and as such open up the possibility for sampling directly from the posterior using a Gibbs sampler in our MCMC algorithm. A death rate for badgers greater than 1 seems extremely unlikely in this context, so a Beta distribution is appropriate.

5.8.1 The Infection Process Parameters; $[\beta_c, \beta_b, \delta, F, \epsilon]$

The conditional joint posterior likelihood of the infection process parameters is given by the product for $t \in 1:T$ of:

$$\begin{aligned}\pi(\beta_c, \beta_b, \delta, \epsilon, F | \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t) =& \mathbb{P}\left[\mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t \mid \mathbf{M}_t,\ \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t,\ \boldsymbol{\theta}\right] \\ & \times \mathbb{P}\left[\mathbf{V}_{(t+1)} \mid \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{D}^c{}_t,\ \mathbf{B}^b{}_t,\ \mathbf{D}^b{}_t,\ \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t,\ \boldsymbol{\theta}\right] \\ & \times \pi(\beta_c) \times \pi(\beta_b) \times \pi(\delta) \times \pi(\epsilon) \times \pi(F)\,.\end{aligned}$$

5.8.2 The Detection Process Parameters; $[\rho, \rho_E]$

The conditional joint posterior likelihood of the detection process parameters is given by the product for $t \in 1:T$ of:

$$\pi\left(\rho, \rho_E | \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t\right) = \mathbb{P}\left[{\mathbf{H}^E}_t,\ {\mathbf{H}^I}_t \mid \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t,\ \mathbf{test}_t,\ \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t,\ \boldsymbol{\theta}\right] \times \pi\left(\rho\right) \times \pi\left(\rho_E\right)$$

5.8.3 The Badger Birth/Death Process Parameters; $[\eta_b, \eta_d]$

The conditional joint posterior likelihood of the badger birth and death process parameters is given by the product for $t \in 1:T$ of:

$$\pi\left(\eta_b, \eta_d | \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t\right) = \mathbb{P}\left[{\mathbf{B}^b}_t,\ {\mathbf{D}^b}_t \mid \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{dE}^b_t,\ \mathbf{dI}^b_t,\ {\mathbf{H}^E}_t,\ {\mathbf{H}^I}_t,\ \mathbf{d}^c_t,\ \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t,\ \boldsymbol{\theta}\right] \times \pi\left(\eta_b\right) \times \pi\left(\eta_d\right)$$

The Gamma prior for η_b is conjugate, thus we get,

$$\pi\left(\eta_b | \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t\right) \propto \text{Gamma}\left(\left(\sum_{t=1}^{T} B^b_t + \phi_{\eta_b}\right),\ \left(\frac{1}{\sum_{t=1}^{T} Y'_t + \frac{1}{\sigma_{\eta_b}}}\right)\right)$$

The Beta prior for η_d is also conjugate, thus,

$$\pi\left(\eta_d | \mathbf{X}_t,\ \mathbf{Y}_t,\ \mathbf{V}_t\right) \propto \text{Beta}\left(\left(\sum_{t=1}^{T} D^b_t + \phi_{\eta_d}\right),\ \left(\sum_{t=1}^{T}\left[Y'_t - D^b_t\right] + \sigma_{\eta_d}\right)\right)$$

5.8.4 Data Augmentation

Thus far we have presented the likelihood assuming that all of the data is observed. This is not the case in the real data, and whilst we have access to the true data for

the partially simulated dataset, we will be assuming some of the data is unobserved to validate the methodology. For missing event data, as demonstrated in chapter 3, if we augment the data using the 'moving an event in time' method then we will need to calculate all posterior terms between the time the event moved from to the time an event moved to. If we augment the data using the 'adding/removing an event' method, then well will need to calculate all posterior terms from the time the event was added to the end of the process. Further details are given in Section 5.10.

5.9 MCMC Methodology

5.9.1 Adaptive Block MCMC

The MCMC schema for making inference on our model of Bovine Tuberculosis is an adaptive block random walk metropolis on the log scale, as explained in Chapter 3.

First we group parameters into sets. The sets are made up of parameters that contribute to the same process(es), or that will be strongly correlated. Thus they will benefit from a proposal distribution that takes into account the inter-dependence.

The two groups are the infection process parameters $[\beta_b, \beta_c, \delta, F, \epsilon]$ and the parameters related to the detection process, $[\rho, \rho_E]$. A third group is the badger birth-death process parameters, $[\eta_b, \eta_d]$, but for now we will not consider them.

Let $\boldsymbol{\theta}$ be a set of parameters, with $\boldsymbol{\theta}^*$ being the current accepted values, and $\boldsymbol{\theta}'$ being the proposed values. For each group we run an adaptive block random walk Metropolis MCMC on the log scale that occurs in two stages. For the first 5000 samples we propose updates from the following joint log-proposal distributions of the parameters, which we call Mixture 1:

$$\log(\boldsymbol{\theta}') \sim \text{Multivariate-Normal}(\log(\boldsymbol{\theta}^*), \frac{1}{d}\lambda^2\mathbf{I}),$$

where d is the dimension of the parameter space, and λ is a tuning parameter specific to the parameter set, and $\mathbf{I}$ is the $d \times d$ identity matrix.

For the infection parameters $d = 5$ and $\lambda = \lambda_{inf}$, and for the detection parameters $d = 2$ and $\lambda = \lambda_{det}$. We update the value of λ_{inf} and λ_{det} every 25 samples for the first 5000 samples, and then they are fixed.

After the first 5000 samples, with probability 0.95 we propose from the following joint log-proposal distributions of the parameters, which we call Mixture 2, else we propose from Mixture 1:

$$\log(\boldsymbol{\theta}') \sim \text{Multivariate-Normal}(\boldsymbol{\theta}^*, m^2\Sigma_{it}),$$

where m is a tuning parameter specific to the parameter set, and Σ is the covariance matrix for the accepted parameters in the parameter set.

For the infection parameters $m = m_{inf}$ and $\Sigma = \Sigma^{inf}$, and for the detection parameters $m = m_{det}$ and $\Sigma = \Sigma^{det}$. We update m_{inf}, Σ^{inf}, m_{det}, and Σ^{det} after every sample.

5.9.2 Updating the tuning parameters

We have two classes of tuning parameters to update, λ and m.

For λ, let us define δ_k as the log-rate of adaptation of the tuning parameter λ. We let each 'batch' of 25 samples be denoted by the subscript k. Then let ψ_k be the proportion of Metropolis-Hastings samples that were accepted in batch k. Then, at the start of each new batch, update λ using the formula:

$$\log(\lambda) = \log(\lambda) + \nu_k$$

where,

$$\nu_k = \begin{cases} -\min\left(0.05, \frac{1}{\sqrt{k}}\right), & \text{if } \psi_{k-1} < 0.33, \\ +\min\left(0.05, \frac{1}{\sqrt{k}}\right), & \text{if } \psi_{k-1} \geqslant 0.33, \end{cases}$$

Then for m, define the rate of adaptation to be $\Delta_m = \frac{m_0}{100}$. We begin to tune m after the first 5000 iterations, and do so every iteration. Each iteration, it, if the proposal came from Mixture 1, m does not get updated. Otherwise, if the proposed

parameters are Metropolis-Hastings rejected, then set

$$m = m - \left(\frac{\Delta_m}{\sqrt{it}}\right)$$

and if the proposed parameters are Metropolis-Hastings accepted, then set

$$m = m + 2.3\left(\frac{\Delta_m}{\sqrt{it}}\right).$$

We use a unique value of ν_k, Δ_m, and batch acceptance rate for each set of parameters. The forms of these update functions are chosen to optimise for an acceptance rate of 30% as per Sherlock, Fearnhead, and Roberts, 2010.

This is the core methodology we will use to make inference on the parameters given all data is known, however there are a number of events in this process that are not observed. In the next section we explain which events are not observed, and the methodology we will use to augment this missing data.

5.10 Data Augmentation

Thus far we have assumed all data was observed and available, but in reality this is not the case. There are a number of unobserved elements of the process that the posterior of the parameters depends on. In this section we distinguish between what data we are fitting our model to (observations) and what is unobserved and augmented through data augmentation (latent variables). We also distinguish between data that is observed by APHA, data that is simulated but assumed observed, and data that is unobserved but necessary for fitting the model (latent variables).

5.10.1 Data, Latent Variables, and Parameters

We categorise the model components into one of three classes which are relevant to the fitting process; data is observed and used the fit the model, latent variables

are unobserved but inferred using data augmentation during the MCMC process, and parameters are inferred through the MCMC process. Some of the data is taken directly from APHA, other parts are simulated but taken to be observed in this example, and some are simulated and considered unobserved.

The data that we are fitting the model to include, for each farm i at time t in our set of farms of interest, ω: The number of cattle and badgers ($\mathbb{C}_{i,t}$, $\mathbb{B}_{i,t}$), the number of cattle movements from farm i to j ($m_{i,j,t}$), whether or not it is a test day ($\text{test}_{i,t}$), assuming all cattle are tested, the number of cattle that were detected ($H_{i,t}$), the number of cattle births ($b^c_{i,t}$), the number of cattle deaths ($d^c_{i,t}$), the number of badger births ($b^b_{i,t}$), and the number of badger deaths ($d^b_{i,t}$).

The latent variables that we infer through data augmentation include, for each farm i at time t in our set of farms of interest, ω: The initial states of the cattle ($\mathbf{X}_{i,0}$), the initial states of the badgers ($\mathbf{Y}_{i,0}$), the initial level of environmental infection ($V_{p(i),0}$), the states of the animals that moved to each destination ($M_{i,j,t}$), the number of newly exposed and infectious cattle and badgers ($dE^c_{i,t}$, $dI^c_{i,t}$, $dE^b_{i,t}$, $dI^b_{i,t}$), the number of detected exposed and infectious cattle from testing ($H^E_{i,t}$, $H^I_{i,t}$), the states of the cattle and badgers that died ($D^c_{i,t}$, $D^b_{i,t}$), and the additional and remaining environmental pressure ($N_{p(i),t}$, $R_{p(i),t}$).

The model is parameterised by 9 parameters; [β_c, β_b, δ, F, ϵ] relating to the infection process, [ρ, ρ_E] relating to the testing process, and [η_b, η_d] relating to the badger birth and death process. The parameters are as in Table 5.1.

Thus we can reformulate the posterior presented in Section 5.8 to distinguish between the observed data, the latent variables, and the parameters. Denote the parameters $\boldsymbol{\theta}$. For the product of times t in $1:T$:

$$\begin{aligned}\pi\left(\boldsymbol{\theta} \mid \mathbf{X}_t, \mathbf{Y}_t, \mathbf{V}_t\right) =& \mathcal{L}\Big(\mathbb{C},\ \mathbb{B},\ \mathbf{m},\ \mathbf{test},\ \mathbf{H},\ \mathbf{b}^c,\ \mathbf{d}^c,\ \mathbf{B}^b,\ \mathbf{D}^b \mid \\ & \qquad \mathbf{X}_0,\ \mathbf{Y}_0,\ \mathbf{V}_0,\ \mathbf{M},\ \mathbf{dE}^c,\ \mathbf{dI}^c,\ \mathbf{dE}^b,\ \mathbf{dI}^b,\ \mathbf{H}^E,\ \mathbf{H}^I,\ \mathbf{D}^c,\ \mathbf{N},\ \mathbf{R},\ \boldsymbol{\theta}\Big) \\ & \times \pi\left(\mathbf{X}_0,\ \mathbf{Y}_0,\ \mathbf{V}_0,\ \mathbf{M},\ \mathbf{dE}^c,\ \mathbf{dI}^c,\ \mathbf{dE}^b,\ \mathbf{dI}^b,\ \mathbf{H}^E,\ \mathbf{H}^I,\ \mathbf{D}^c,\ \mathbf{N},\ \mathbf{R} \mid \boldsymbol{\theta}\right) \\ & \times \pi\left(\boldsymbol{\theta}\right)\end{aligned}$$

As a rough summary, for the cattle we know how many animals there are, and how many of each type of event, and the latent variables are the states of the animals and the states of the animals affected by the events. The increase and decay of the environmental reservoir are also latent variables.

5.10.2 Missing Data

We now assume that much of the process is not observed and as such we do not have data through which to calculate the likelihood. These data are called nuisance parameters and we will need to use data augmentation MCMC methodologies to fill in the missing pieces intelligently. From this point forward we will assume that all data relating to badgers is still known. Given the form of the likelihood in Section 5.7, we can see that our list of nuisance parameters will consist of:

5.10.2.1 Initial Conditions

- $V_{p(i),0}$ - The level of infection in the parish environment for each parish at time $t = 0$.
- $X_{i,0} = [x^S_{i,0}, x^E_{i,0}, x^I_{i,0}]$ - The initial states of each farm i.

5.10.2.2 Events

- $M_{i,t}$ - All data relating to the movements off of farm i, including animal states and populations that generated movements.

- $dE^c_{i,t}$ - The number of Exposed cattle that farm i generated during each of the timesteps $t \in 1:T$.
- $dI^c_{i,t}$ - The number of Infectious cattle that farm i generated during each of the timesteps $t \in 1:T$.
- $H^E_{i,t}$ - The number of Exposed cattle that were detected through testing on farm i during each of the timesteps $t \in 1:T$.
- $H^I_{i,t}$ - The number of Infectious cattle that were detected through testing on farm i during each of the timesteps $t \in 1:T$.
- $D^c_{i,t} = [D^{cS}_{i,t},\ D^{cE}_{i,t},\ D^{cI}_{i,t}]$ - The states of the animals that died of causes unrelated to testing at each timestep $t \in 1:T$ for each farm i.
- Remaining environmental pressure $= R_{p(i),t}$ - The amount of environmental pressure remaining at timestep t after decaying from timestep $t-1$ for the parish of farm i.
- New environmental pressure $= N_{p(i),t}$ - The additional environmental pressure resulting from the infectious animals in the parish at time step t for farm i.

If we assume we have the 2172 farms of Cheshire, and a process run for 360 timesteps (weeks), the 15 nuisance parameters per timestep per farm (treating the movements as only 6 nuisances parameters) which represent the transitions, and the additional 4 nuisance parameters of the initial states, results in $2172 \times (4 + 360 \times 15) = 11,737,488$ nuisance parameters for Cheshire alone. There are in fact closer to $80,000$ farms in the dataset. Even with all of the discretisation this is still an exceptionally challenging and computationally expensive problem.

5.10.3 Update Steps

We are interested in using two distinct data augmentation update steps to thoroughly explore the state space. Given we have initialised the inference scheme

with a valid set of nuisance parameters (unknown data), we can update them either by moving the events that already exist through time, or by adding or removing events (sometimes referred to as exchanging animal states depending on the event). This is fundamentally the methodology presented in Chapter 3, though the complexity of this model adds additional considerations as we will explore.

When we move an event in time, we take an event that occurs at time $t = \tau$ and change it to occur at time $t = \tau + \Delta$, and update the data accordingly. In this case only the animal states between τ and $\tau + \Delta$ will be affected, as the total number of events remains the same. This is demonstrated in Figure 5.3.

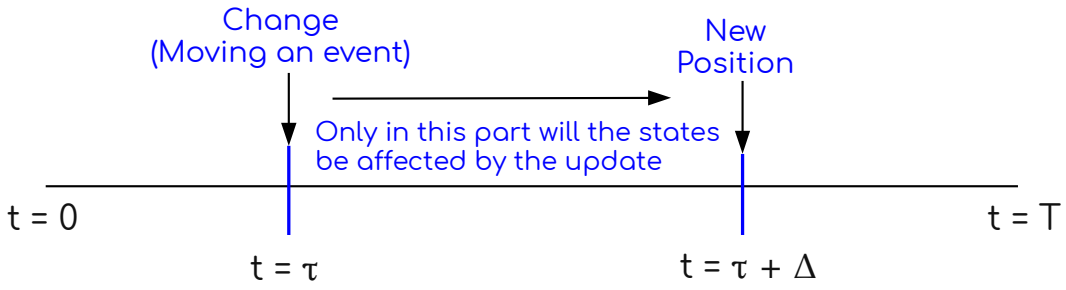


Figure 5.3: A timeline representing moving an event in time. Only states between τ and $\tau + \Delta$ will be affected by the update.

When we add or remove an event, we are changing the total number of events. We choose a timestep τ and add or remove an event there, after which point the total number of events has changed, and as such there are no guarantees the states of the farm will remain the same at any future timestep. This is demonstrated in figure 5.4.

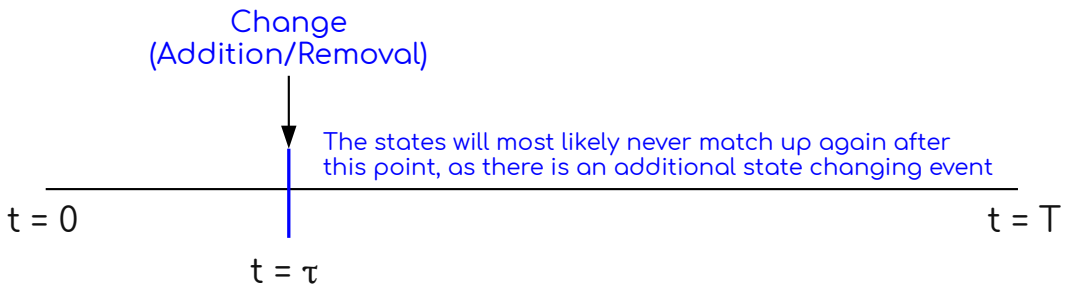


Figure 5.4: A timeline demonstrating the changes from an addition/removal data update. All states after time τ are assumed to be affected, as there is an additional state changing event.

We are mainly concerned with three aspects here, especially as they relate to computational complexity: Which types of update can be used for each type of event? What timesteps of the likelihood need to be calculated for each event-update type pair? Which farms in the likelihood need to be calculated for each event-update type pair? We will considered each of the event types in turn addressing these three questions.

5.10.4 Movement events

For a given timestep, t, the number of movements is taken directly from the data. Our nuisance parameters (our missing data) are the states of the animals that were moved. For this reason, we are unable change when events occur by moving them through time. We can only exchange the states of the animals that were moved, for instance, we can remove a susceptible movement and add an exposed movement. To do this we generate a new set of movement states from its distribution.

The elements of the likelihood that need to be calculated depend on the farms involved in the movement, and the states of the animals. We consider here the most extreme case of moving infectious animals between two farms in two different parishes within our collection of farms of interest. This update requires us to calculate the full likelihood for both farms from time t onward as per Figure 5.4, and due to moving infectious animals between parishes, the total counts of infectious animals on each will change, and as such the parish level terms of the likelihood (environmental reservoir) will need to be recalculated.

5.10.5 Exposure and Infection transition events

For both S to E transition events, and E to I transition events, we can both move the existing events through time and add/remove events, as long as all following events of every kind are still valid.

When moving an event from t to $t + \Delta$ we only need to calculate the likelihood between the time it was moved from and the time it was moved to. When we add/remove events at time t, we will again need to calculate the full likelihood for all events from that time forward.

The infection process is local to each farm, so we will only need to calculate the likelihood for the given farm. In the case of E to I transition events, the number of infectious cattle will also change, and so the parish level likelihood terms will also need to calculated for this farms parish.

5.10.6 Detection events

In this partially simulated example, we took the initial test dates from each farm from the data, and then simulated the test results and future test dates based on government policy. To align with the real data scenario, we will assume we know when the events occurs (the simulated test dates), how many animals were tested (all animals present on the farm), and how many animals were detected (total), but not the states of the animals on the farm or how many were detected in each state.

This means similarly to the movement events, detection events cannot be moved through time, as the number per day are observed, and we can only exchange the detection of an exposed cattle with the detection of an infectious cattle, as the states are unobserved latent variables. Again we would do this by generating a new set of detections from an appropriate distribution.

As it is an add/remove step, we need to calculate all the likelihood terms for the test farm from the test date onwards. As long as the epidemic remains valid after the update this will not effect other farms, but the number of infectious cattle on the farm will change and so the parish level likelihood terms for that farms parish will need to be updated also.

5.10.7 Birth and Death events

Cattle birth events all come from the data, and they always produce a susceptible individual, so there is nothing to augment here.

The number of death events also come from the data, so again it is the states of the animals that died that are our nuisance parameters. As with movements and detections, these cannot be moved through time, but the states of the animals detected can be exchanged by generating a new realisation from the appropriate distribution. We will again need to calculate the full likelihood for all events on that farm from that time forward. It is possible for the number of infectious cattle to change as well so the parish level terms will also need to be updated.

5.11 Results for Partially Simulated Data

Using the process in Section 5.6 we generated a partially simulated epidemic data set, utilising the initial conditions from the real data and the real movement data. Thus we knew the true states of all animals at all timesteps, the events of the epidemic process, and the true parameters that generated the epidemic.

Using the processes laid out in Section 5.9 onwards we performed inference

for this simulated epidemic. As previously stated we assumed all data relating to badgers to be known, and did not make inference on the badger birth and death parameters. All other data that was considered unknown is detailed in Section 5.10.2.

We made inference on the parameters using the block adaptive MCMC methodology, grouping the parameters into two groups; the infection process parameters $[\beta_c, \beta_b, \delta, F, \epsilon]$ and the detection process parameters $[\rho, \rho_E]$.

We focused our inference on 307 farms, in 24 parishes, in the county of Cheshire, for the 360 weeks from the 1st January 2012. The algorithm was initialised with the true data.

The true values of the parameters were $[\beta_c, \beta_b, \delta, F, \epsilon] = [0.002, 0.004, 0.015, 0.004, 0.05]$ and $[\rho, \rho_E] = [0.75, 0.2]$.

The priors were set to be:

- $\beta_c \sim \mathsf{Gamma}(2, 0.001)$
- $\beta_b \sim \mathsf{Gamma}(2, 0.002)$
- $\delta \sim \mathsf{Gamma}(3, 0.005)$
- $F \sim \mathsf{Gamma}(2, 0.002)$
- $\epsilon \sim \mathsf{Gamma}(1, 0.05)$
- $\rho \sim \mathsf{Beta}(1.5, 0.5)$
- $\rho_E \sim \mathsf{Beta}(0.4, 1.6)$

The below results are the output of $400,000$ samples after burn-in. The following table presents the summaries of the marginal posterior distributions:

	True Value	Mean	95% CI	Std. Dev	ESS
β_c	0.002	0.00269	(0.0016, 0.0040)	0.000624	694.3
β_b	0.004	0.00386	(0.0027, 0.0052)	0.000634	1548.5
δ	0.015	0.01099	(0.0099, 0.0123)	0.000613	4969.9
F	0.004	0.00389	(0.0035, 0.0044)	0.000233	1573.5
ϵ	0.05	0.0496	(0.0491, 0.0503)	0.000305	73456.6
ρ	0.75	0.7461	(0.719, 0.773)	0.013700	9636.9
ρ_E	0.2	0.1402	(0.123, 0.159)	0.009230	6869.2

Table 5.5: The summary of the marginal posterior distributions.

Overall the algorithm recovered tight uni-modal and symmetric posteriors around each of the parameters, as we can see in Figures 5.5 and 5.6. In most cases the true

parameter values lie well within the posterior mass and usually very close to the posterior mode. However, for δ and ρ_E the true values are well outside the posterior mass. The parameters however are still within realistic bounds, and due to the stochastic nature of the epidemic process it is reasonable that events would occur like this. We feel the posterior estimates are valid.

From Figures 5.7 and 5.8 we can see that the mixing was good, with large jumps and time spent exploring all areas of the posterior mass. The algorithm struggled the most with β_c, but still performed well overall.

Figures 5.9 and 5.10 show the pairwise contour plots for all parameters in the infection process set, and then the detection process set. The red dotted lines represent the true parameters, and the yellow dotted lines represent the position of the pair of parameters with the highest posterior mass. Again we see that the areas of highest posterior mass align well with the true parameters except for the E to I transition rate, δ. The correlation between the parameters does not present strongly in these plots, nor did it in the trace plots inspected.

The effective sample sizes for each of the parameters is given in Table 5.5. The average acceptance rate of the infection parameters was 31.01%. The average acceptance rate of the detection parameters was 30.10%.

For the data augmentation, the average acceptance rate for: moving S to E exposure events was 94.52%, moving E to I infection events was 56.53%, adding or removing S to E exposure events was 0.13%, adding or removing E to I infection events was 0.04%, exchanging detection events was 0.12%, exchanging cattle death events was 0.00%, adding or removing environmental reservoir pressure was 0.56%, and exchanging movement events was 91.41%.

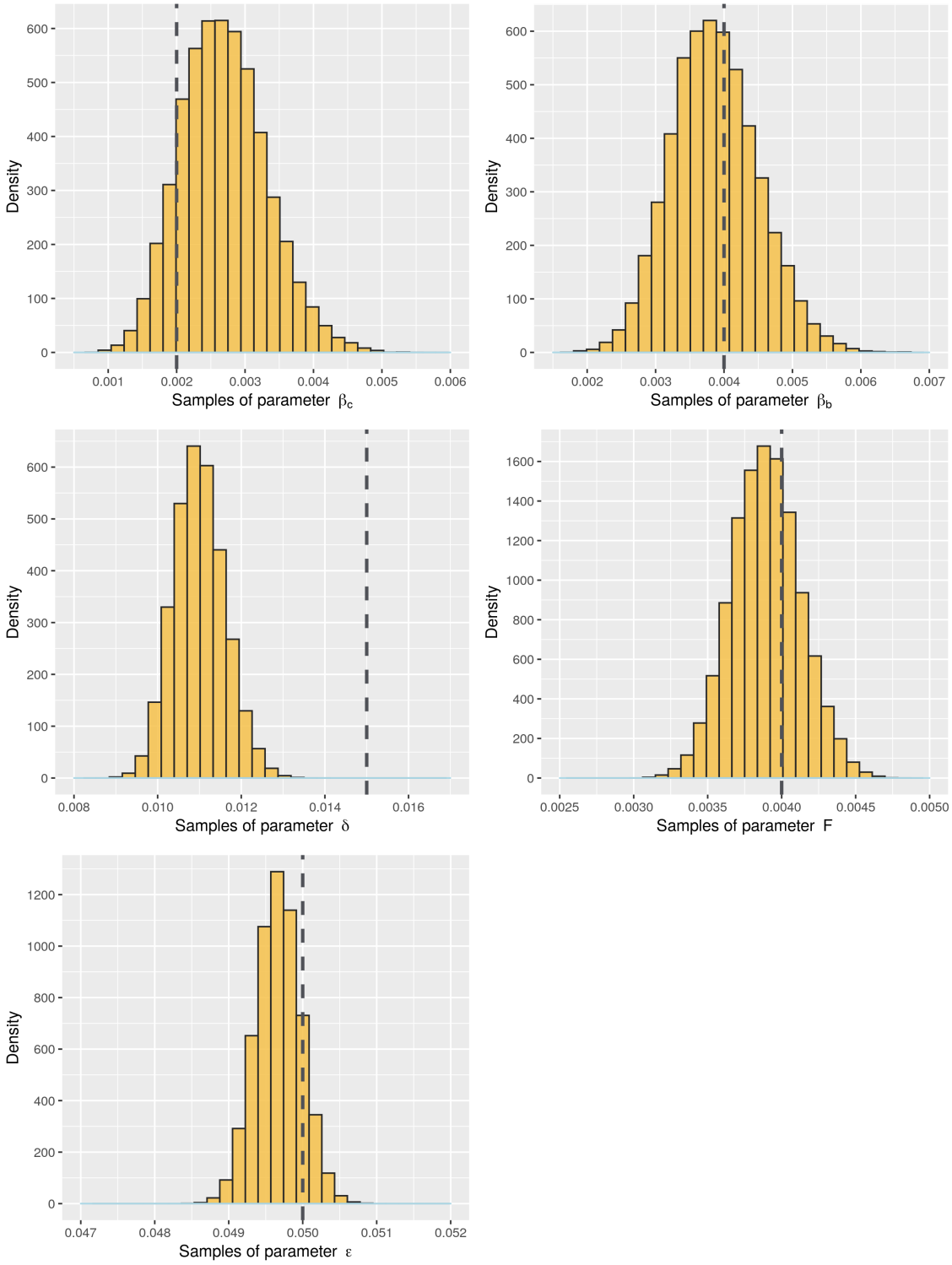


Figure 5.5: Results: The posterior samples of the infection process parameters displayed as their marginal distributions represented in a histogram. The dashed line represents the true value that generated the partially simulated data set. The priors are shown in blue.

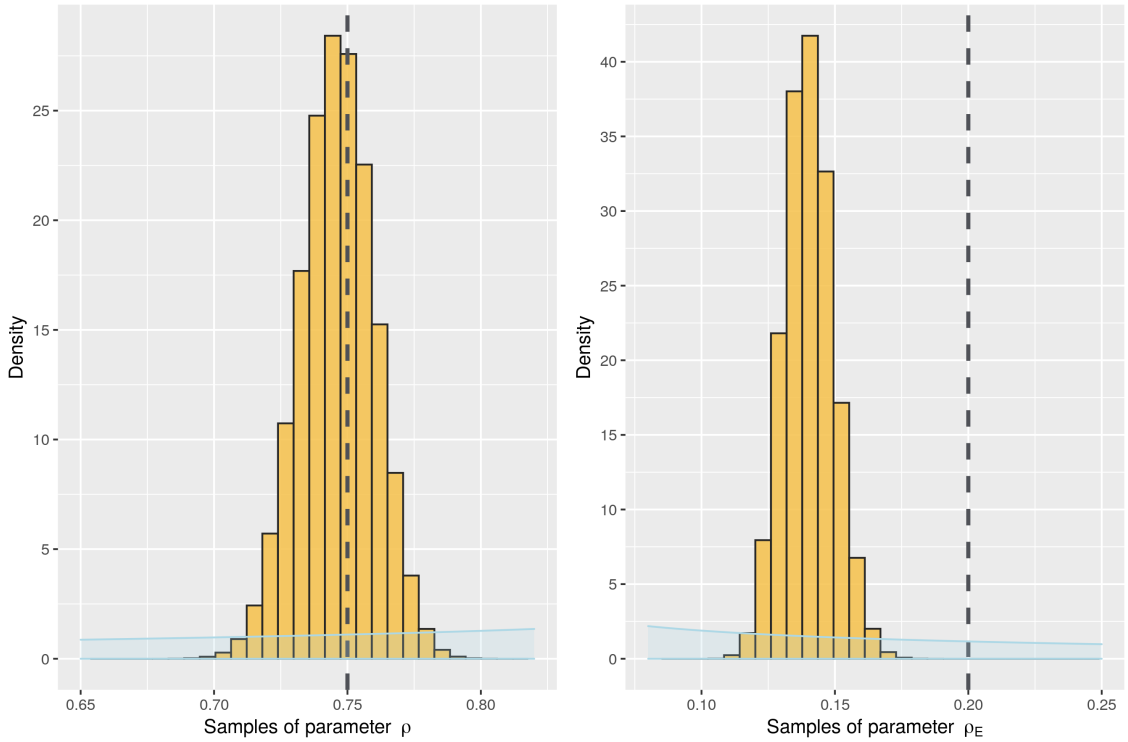


Figure 5.6: Results: The posterior samples of the detection process parameters displayed as their marginal distributions represented in a histogram. The dashed line represents the true value that generated the partially simulated data set. The priors are shown in blue.

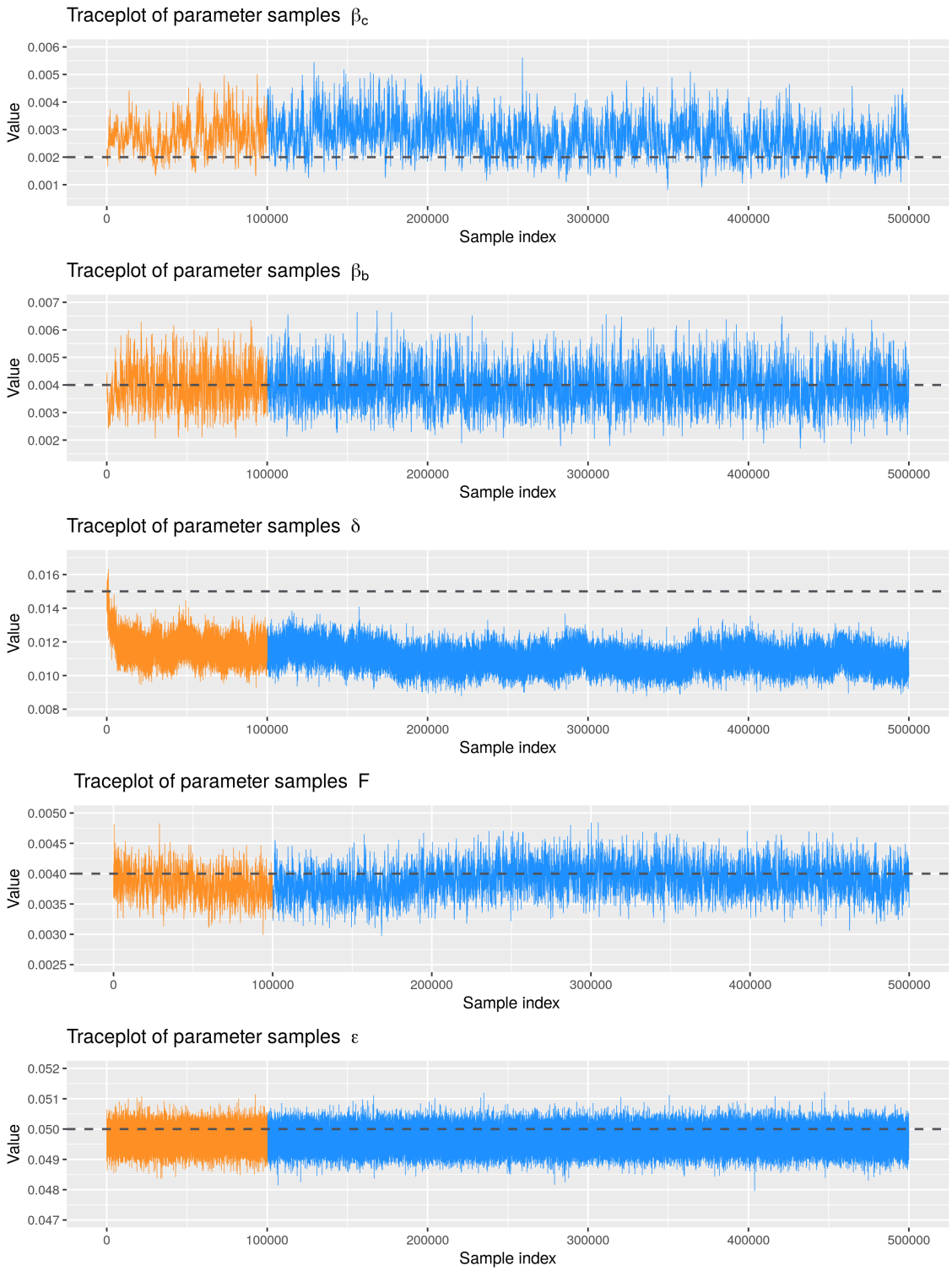


Figure 5.7: Results: Trace plot for β_c, β_b, δ, F, and ϵ. The orange area represents a portion of the burn-in, and the dashed line represents the true value that generated the partially simulated data set.

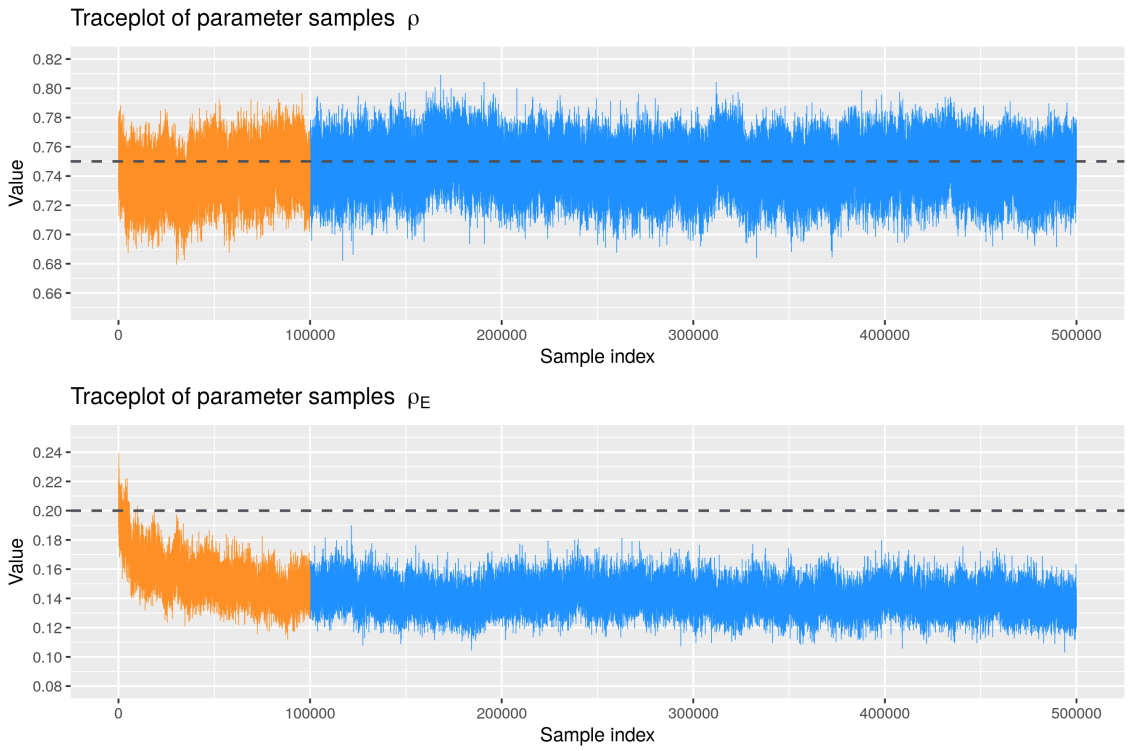


Figure 5.8: Results: Trace plot for ρ and ρ_E. The orange area represents a portion of the burn-in, and the dashed line represents the true value that generated the partially simulated data set.

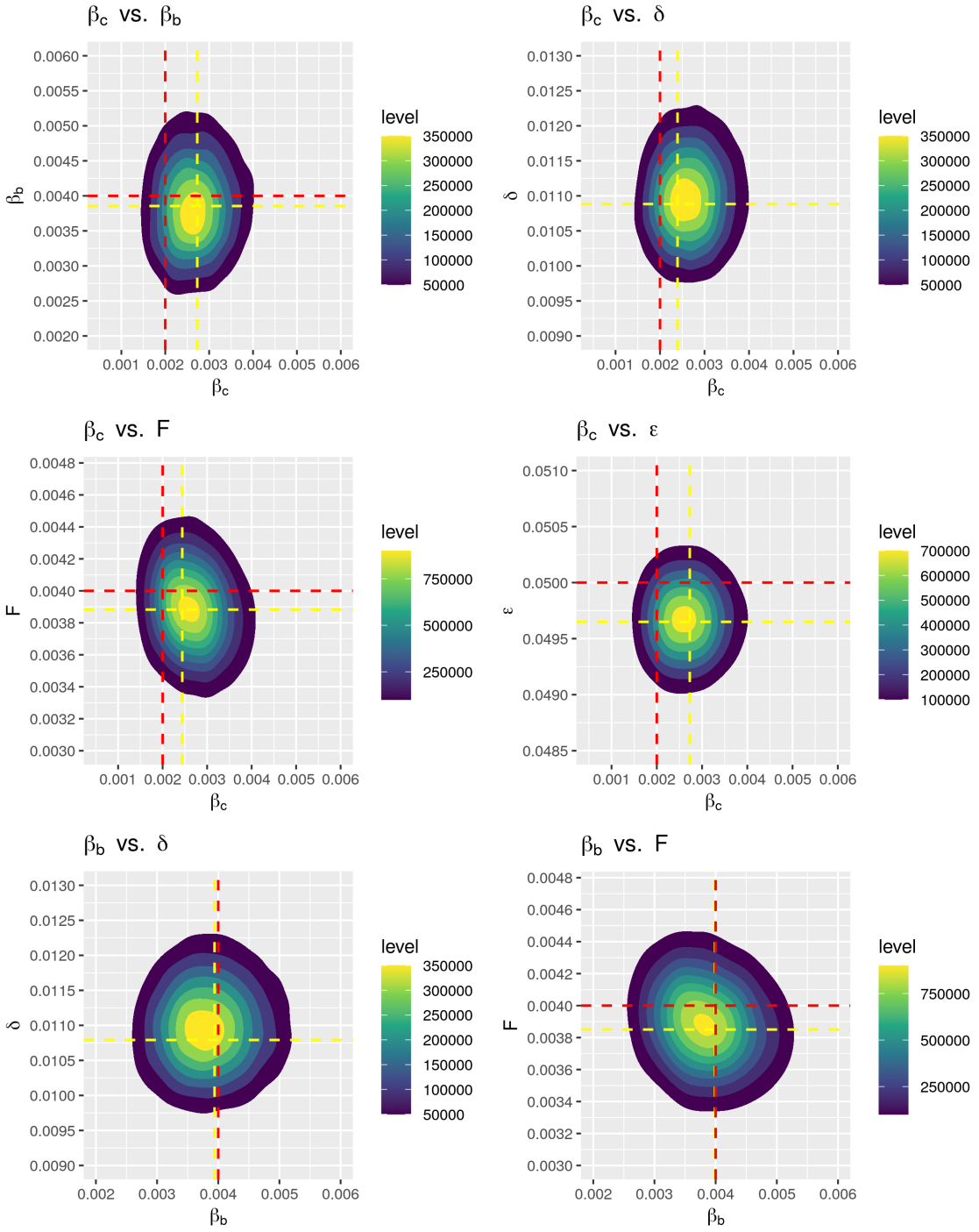


Figure 5.9: Results: Contour plots of the posterior samples for each pair of the infection parameters. Brighter contours represent areas of higher density. The yellow dashed lines show the pair-wise point of highest density on the contour plots, and the red dashed lines represent the true values of the parameters that generated the simulation. From left to right, top to bottom, the plots show β_c *vs* β_b, β_c *vs* δ, β_c *vs* F, β_c *vs* ϵ, β_b *vs* δ, and β_b *vs* F.

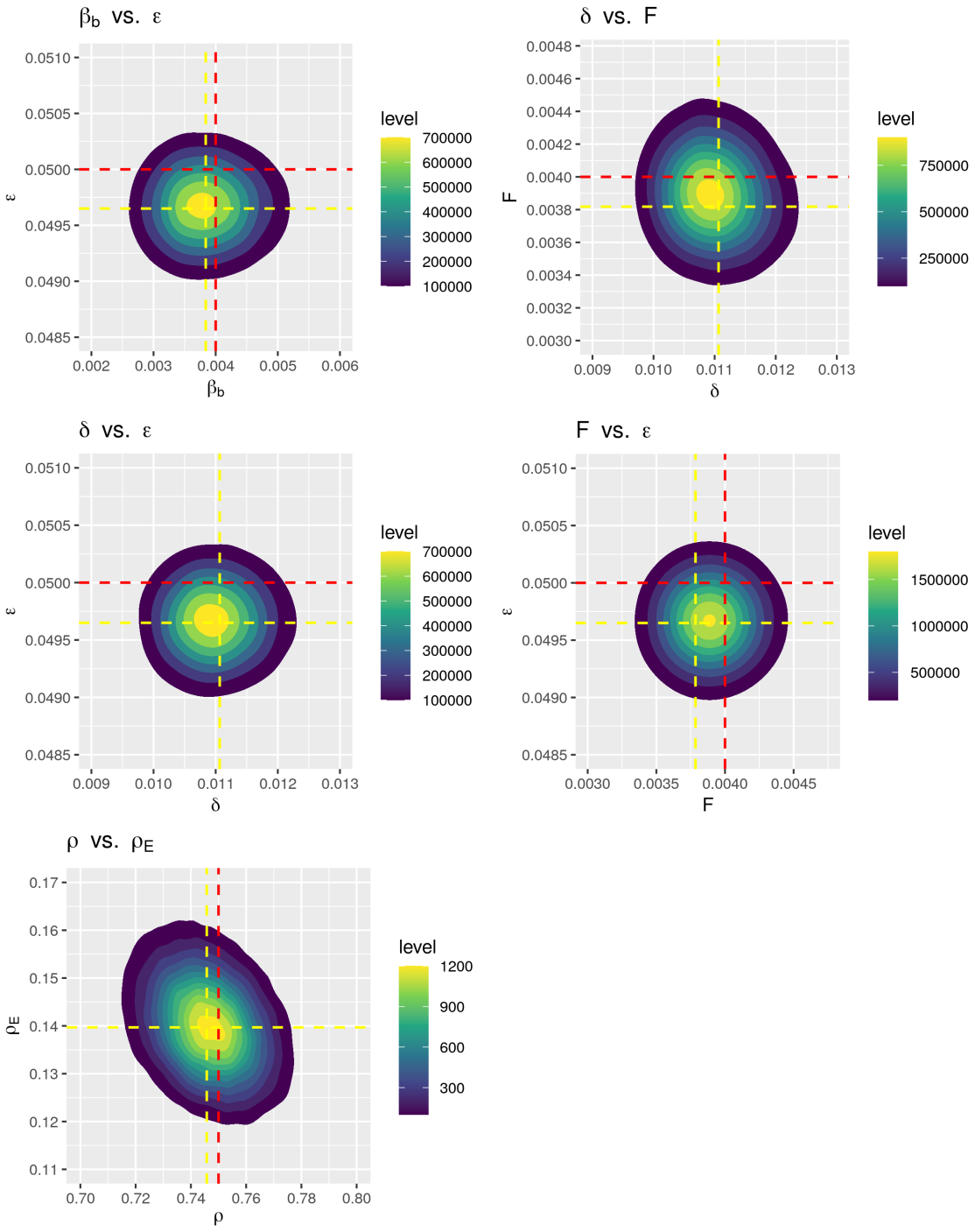


Figure 5.10: Results: Contour plots of the posterior samples for each pair of the infection parameters. Brighter contours represent areas of higher density. The yellow dashed lines show the pair-wise point of highest density on the contour plots, and the red dashed lines represent the true values of the parameters that generated the simulation. From left to right, top to bottom, the plots show β_b *vs* ϵ, δ *vs* F, δ *vs* ϵ, F *vs* ϵ, and ρ *vs* ρ_E.

5.12 Discussion

In this Chapter we simulated a Bovine Tuberculosis epidemic on 307 farms, in 24 parishes, in the county of Cheshire, for the 360 weeks, based on real population and movement data, with the addition of a fully simulated badger population. We developed an MCMC algorithm to make inference on the parameters of the model, except the badger birth and death rates, whilst augmenting the missing data. Overall the algorithm was effective at exploring the posterior distributions, with well mixed trace plots, distinct uni-modal histograms, and recovered the parameters that generated the epidemic, however given an inference of $400,000$ samples, some of the effective sample sizes are rather low. Now that we have validated that an MCMC scheme like this can be effective at making inference on a bTB epidemic of this scale, we can build upon it to make inference for the real data.

There are some areas of improvement to explore. Firstly the code is written in Julia, optimised as much as possible, which over the course of development saw speed ups in hundreds of thousands of times. To do this we both optimised the implementation in line with Julia's design principles, and used methodological improvements such as 'online' co-variance matrices which can update with new observations rather than need full recalculations, and caching likelihood terms that do not needed to be updated between two steps. To run this inference now takes roughly one week per million samples on a virtual machine. This is acceptable for this thesis, but will not scale to the 80,000 farms in the data set. Faster implementations could be adopted through the use of other programming languages and more powerful technology like GPUs. The MCMC scheme is linear, limiting speed ups, but calculations like the likelihood can be parallelised, and further improvements to data storage and access can be explored. We don't explore these ideas in this thesis.

Other than speed the other consideration is the efficiency of the algorithm and the data it relies on. It's true that the algorithm performed well but there are two notable points of discussion. The acceptance rates performed as intended for the

parameters of the model, achieving the desired average of a 30% acceptance rate, however the acceptance rate for the data augmentation is less consistent. Updating this discrete data we have less room for tuning, so guaranteeing the ideal 44% acceptance rate in every case would be challenging, but it should be possible to improve the acceptance rates based on different proposal distributions. With the majority of acceptance rates close to 0 or 100%, and the process being initialised at the true data, there is concern that the data augmentation process did not sufficiently explore the missing data. There are further parameters within the current proposal distributions that could be explored, such as the number of farms or timesteps updated during each proposal, which should hopefully improve the acceptance rate of moving events and exchanging cattle movements by making it lower. Currently an update to only one timestep for one farm is proposed for each data augmentation method between updates of the parameters. A change to one farm can knock onto others, but large changes to the whole population are likely to result in invalidating the epidemic. To improve the acceptance rates of the other proposals in this case it is likely we would need a new proposal function. One issue is that a large proportion of the updates are being rejected because they invalidate the epidemic, or in some cases lead to no change. With such a complex epidemic it is challenging to conceive a computational efficient proposal distribution that guarantees a valid proposal. The more conditions and states that need to checked before proposing a timestep and a farm, such as the infection rate or the number of exposed cattle, the more costly the proposal becomes, but likely the higher the acceptance rate, so there is a trade off. Likewise the no change proposals can happen simply because certain events are rare. If one animal is detected on a farm with 100 infectious cattle and 1 exposed cattle, over 99% of proposals are going to return the same state of an infectious cattle being detected. If we manage to reduce this by making proposals that are distinct more often, then we run the risk of those proposals being invalid or unlikely, and its true that in all cases the vast majority of proposals that aren't rejected because of no change are rejected simply because of the Metropolis-Hastings

acceptance probability.

The second area of improvement concerns the data assumed known. We introduced a simulated badger population and assumed all but the S to E and E to I transition rates, and the states of the animals, to be known. This includes the size of the populations, assumptions that the badgers don't move between farms, a constant birth rate, when births and deaths occur, and the death rate. The first issue is that without a large concentrated effort, and even then, it is likely impossible to have accurate data of this level. The largest bTB in badgers investigation was the Randomised Badger Culling Trial (RBCT) in badger by The Independent Scientific Group on Cattle TB, 2008 which culled badgers and looked at the effect on bTB in local cattle herds, and the majority of investigations into bTB in badgers look primarily at roadkill (Chantrey et al., 2018). This data however gives a solid constraint on which to build the cattle epidemic off of, and as such no doubt improves the algorithms ability to make inference. We investigated an algorithm that also made inference on the badger data and parameters at the same time, and the effect was a markedly worse efficiency and identifiability issues. This could be related to a potential confounding issue between F, ϵ, and V, especially in badgers. On the surface if V is specified to be double its true value, then F can be halved and give the same infectious pressure. If we consider that V_{t+1} is generated based on a random removal from $V(t)$ based on its decay rate ϵ, and a random addition from the proportion of infected animals in the parish, then we can see its a little more robust. The addition is unaffected by the parameters, but if V is initialised too large, then the decay rate ϵ may increase to get it to decay faster. We see a similar trade off relationship with β and γ in the basic SIR model, and the parameters are still identifiable as long as there is sufficient data, but when there isn't, for instance because of the limited badger data, this potential confounding issue can exacerbate the challenges of fitting the model. From this we conclude that if efforts are made to collect rigorous population and epidemic data on badgers then this can greatly help the inference of a model for Bovine Tuberculosis in cattle, but trying to make

inference about both populations at once, including augmenting data, is ineffective. When it comes to making inference on the real cattle data in Chapter 6 we will abstract the consideration of the badger population.

As for the model overall, there were a number of simplifying assumptions and modelling choices that were made that could be explored further. Unlike Brooks-Pollock, Roberts, and Keeling, 2014 we did not consider a within-farm environmental effect that is distinct from the parish one - farm-level effects could be an area of further research. In addition our badger population was very simple - most notably there was only one sett per farm, every farm most likely had badgers, and the birth and death processes were constant rather than seasonal or related to disease status. This means for instance that whilst we propose separate parameters for the birth rate and death rate, unless they are equal the population will most likely either die out naturally or continue to grow in size. The dynamics in reality are likely to be far more complex and nuanced. However there was a purpose to the simplifying assumptions, which was to make the inference methods more efficient. For instance by abstracting the movements of cattle to just be the counts in each state, updates to the infection process are much less likely to invalidate the movement process, and if they do invalidate the movements they will be automatically rejected as the likelihood will be zero.

In the following chapter we will take the learnings from this algorithm and apply them to the real data set with no simulated data assumed known. We do not have data on badgers so we will present a new model that abstracts out this population. In addition there are different considerations for the real data which need to be accounted for, including how we initialise the epidemic.

Chapter 6

Real Data Model

6.1 Introduction

In this chapter we adapt the methodology of Chapter 5 to make full likelihood inference for the true epidemic data including all movements, herd tests, births, deaths, and slaughtered cattle in a subset of farms in Cheshire. The model has been updated in line with the unavailability of badger data and we present these changes in Section 6.2. We made slight changes to how we processed the data in light of this model which is explained in Section 6.3, and adapted the data generating process in Section 6.4. In Section 6.6 we explain a new method of initialising the MCMC with a valid epidemic that is conditioned on all of the available data. With the data initialised we explain how we have adapted the MCMC schema for the real data in Section 6.7, including data augmentation steps of the initial conditions to account for uncertainty. Finally in Section 6.8 we interpret the model results for two runs, one with all of the parameters unknown, and one with the detection parameters fixed. We discuss these results in Section 6.9.

6.2 Model Changes

Our previous model for Bovine Tuberculosis is a spatial discrete-time meta-population compartmental model of disease spread. The populations are the farms which contain a herd of cattle and are potentially associated with other wildlife such as badgers. There are cattle movements between farms, and an environmental reservoir effect at the parish level. This model relies on knowledge of the badger populations associated with each farm, which is not data we have access to, especially at a national scale. The following model is an adaptation of the previous model to account for the missing data by removing the explicit dependence on the knowledge of badgers, and combining their effect with the parish level environmental reservoir.

6.2.1 The Observed Data Bovine Tuberculosis Model

The model is parameterised by 6 parameters which are associated with the infection process or the testing process.

Parameter	Description
β_c	The within-farm infectious contact rate of cattle
γ	The cattle exposed to infectious transition rate
ϵ	The environmental reservoir decay probability
F	The scalar of the environmental reservoir infectious pressure
ρ	The detection probability for a infectious cattle
ρ_E	The scalar of the detection probability for cattle in the exposed state

Table 6.1: The parameters of the full data model.

We subsume the badger dynamics into the background environment. The effect of the previous β_b parameter, which represented the badger-to-badger infectious contact rate, will be absorbed into the environmental parameter F. Note however that these effects are inseparable, no badger population is modelled or data used, nor is there any longer a specific contribution from badgers to the environmental effect. Equally we have removed parameters related to badger birth rates and badger death rates.

Given these parameters the infectious pressure for the cattle on farm i at time t is given by

$$\lambda_{i,t}^{c} = \left\{ -\beta_c \frac{x_{i,t}^{I}}{N_{i,t}^{c}} - F \frac{V_{p(i),t}}{A_{p(i)}} \right\}$$

where $V_{p(i),t}$ is the parish level environmental effect ($i \in p$), which is scaled by the size of the parish, $A_{p(i)}$, for $i \in p$. The parish level environment, $V_{p(i),t}$ no longer depends of the level of badger infection in the parish, and is generated using Algorithm 14.

6.3 Data Pre-processing

Our goal in this chapter is to make inference on as full a data set as possible. In the last chapter, for the partially simulated epidemic, we set the initial conditions based on the data, matched the date of the first testing event, and used all movements that were still valid in our simulated epidemic. However future tests were not from the data, but generated in line with policy. Many movements had to be thrown out meaning that the inference would not be able to identify the true effect of movements in the real epidemic. Finally all test results were randomly generated. We wished to build an epidemic dataset which utilised all of the movements, all of the testing events, all of the testing results, all of the births, and all of the deaths.

First we processed the data into the desired aggregation. Again we aggregated to week timesteps and focused on the initial 360 weeks. Births and deaths were tallied per farm. Movements were aggregated such that each cattle only had at most one movement event per week which detailed the farm where the cattle began the week, and the farm where it ended the week. These individual movements were then aggregated for the total number of movements off of Farm i onto Farm j at time t, for all farms.

In line with Brooks-Pollock, Roberts, and Keeling, 2014 we have elected to only focus on Whole Herd Tests - those in the categories of 6M, 12M, CON, RHT, WHT,

CT, and SI - as less than 1% of reactors are detected from individual level testing. This means that pre-movement and post-movement tests are not included. The testing model implemented assumes all animals on the farm are being tested. We assumed this instead of matching the number of animals tested at each testing event due to the level of aggregation and ordinal structure of events, and the challenges with exact matching this introduced, but the total number of slaughtered animals were matched exactly. We aggregated the total number of slaughtered cattle for each farm each week, and took this as a proxy for positive tests.

Apart from testing results, we have no data on the infectious status of animals, or the infectious process, and so no data regarding infection events is available or utilised, nor did we use confirmation of infection data.

6.4 Data Generating Process

We can now build the epidemic model that we are assuming generated this aggregated epidemic, and derive the likelihood components of each sub-process. The core systems of the model work in much the same way as the previous data generating process in Chapter 5, however we have removed the dependence on the badger data. We are also making the assumption that all observed events are possible, and that all data is utilised. As such we are no longer discarding movement data or assuming simulated testing events.

In this section we define the overall process for simulating from the data generating process in terms of transmission kernels. Transition kernels, K, apply a series of operations to a set of states to produce a new set of states; $K :: \mathbf{X}_t \rightarrow \mathbf{X}_{t+1}$

The kernels are,

- $K_M :: X \rightarrow X'$ (Cattle Movements)
- $K_E :: X \rightarrow X'$ (Cattle Epidemic)
- $K_T :: X \rightarrow X'$ (Cattle Testing)

- $K_D :: X \to X'$ (Cattle Births and Deaths)
- $K_V :: V \to V'$ (Environmental Reservoir)

In addition there is also the supporting function, P, to calculate event probabilities; $P :: [\mathbf{X}_t, \mathbf{V}_t] \to Q_t$. Details of the kernels can be found in Chapter 5 if they remain the same as in the previous model, and in Section 6.4.1 if they have been updated.

We define K_0 to be the kernel for generating a new timestep in the epidemic; $K_0 = K_M \circ K_E \circ K_T \circ K_D \circ K_V :: [\mathbf{X}_t, \mathbf{V}_t] \to [\mathbf{X}_{t+1}, \mathbf{V}_{t+1}]$. The details of K_0 are given in Algorithm 11.

Algorithm 11: Generate states for timestep $t+1$

Input : X_t = Cattle states at beginning of timestep t,
V_t = Environmental reservoir at beginning of timestep t,
θ = Model parameters,
$M_t,\ h_t,\ b_t,\ d_t$ = Data during timestep t.

Output : X_{t+1} = Cattle states at beginning of timestep $t+1$,
V_{t+1} = Cattle states at beginning of timestep $t+1$.

Elements: Q_t = Event probabilities during timestep t,
$X',\ X'',\ X''',\ X'''',\ V'$ = Intermediate states.

$K_0(X_t, V_t)$::
- **Probabilities**
- $Q_t = P(X_t, V_t)$
- **Cattle**
- $X' = K_M(X_t,\ M_t)$
- $X'' = K_E(X',\ Q_t)$
- $X''' = K_T(X'',\ h_t)$
- $X'''' = K_D(X''', b_t, d_t)$
- **Environment**
- $V' = K_V(V_t, X'''', Y'')$

Return X'''', V'

6.4.1 Details of Kernels

In this section we provide the context and details of each of the subroutine and transition kernels necessary for Algorithm 11 where they are different to Chapter 5. We also define the likelihood component for each function.

6.4.1.1 The Probability Function

For each farm, i, we first calculate the exposure and infection probabilities during timestep t given the animal states at beginning of timestep t. We are no longer

calculating probabilities related to events in the badger population.

Algorithm 12: Generate event probabilities for timestep t

Input : X_t = Cattle states at beginning of timestep t,
V_t = Environmental reservoir at beginning of timestep t,
$[\beta_c, \delta, F] \in \theta$ = Model parameters.

Output : Q_t = Event probabilities during timestep t.

Elements: $p^c_{exp}(i,t),\ p_{inf}$ = Event probabilities during timestep t,
ω = The set of farms of interest.

$P(X_t, V_t, \beta_c, \delta, F)$::
  foreach $i \in \omega$ **do**
    $p^c_{exp}(i,t) = 1 - \exp\left\{-\beta_c \frac{x^I_{i,t}}{N^c_{i,t}} - F\frac{V_{p(i),t}}{A_{p(i)}}\right\}$
  end
  $p_{inf} = 1 - \exp\{-\delta\}$
  $Q_t = [p^c_{exp}(i,t),\ p_{inf}]\ \ \forall\ \ i$
  Return Q_t

6.4.1.2 The Cattle Movement Kernel

As in Chapter 5, dividing the movements into 4 distinct sets, we can process the movement data. The distinction with this model is that instead of throwing out movements that didn't work, we assume that all events are possible. There is still the possibility that certain farms need to be processed before others, and of artefacts in the data leading to misalignments, and as such the loops in the process to try and retry farms are still present. This is essentially in lieu of using finer data that maintains the order of movements, and as such may be slower to process as farms need to be tested and retested, but shouldn't affect convergence as it should get to the correct state.

We define M_t to be the full set of movements during time t, with $m_{i,j}(t)$ being the number of animals moved from farm i to farm j during time t. We still assume the states of the animals moved are generated using a Multivariate Hypergeometric,

denoted MHG(N, n) where N is the population vector and n is the number of draws.

The kernel can then be written as:

Algorithm 13: Generate event probabilities for timestep t

Input : X_t = Cattle states at beginning of timestep t,
M_t = Movement data during timestep t.
Output : X' = Post movement cattle states during timestep t.
Elements: X^* = Intermediate cattle states,
R = Set of farms that were unable to be processed during the previous loop,
R' = Set of farms that were unable to be processed during the current loop,
ω = The set of farms of interest.

$K_M(X_t,\ M_t)$::
  Step 1: $(\omega^c \to \omega)$
    foreach $i \in \omega^c,\ j \in \omega$ **do**
      $a_{i,j,t} = \text{Multinomial}\left(m_{i,j,t},\ [0.9,\ 0.05,\ 0.05]\right)$
      $X^* = X_t + a_t$
    end
  Step 2: $(\omega \to \omega)$
    Set $R = \varnothing$, $R' = \varnothing$
    for $i \in \omega$ **do**
      if $\sum_{j \in \omega} m_{i,j,t} > N^*_{i,t}$ **then**
        $R' = \{R' \cup i\}$
        next
      else
        $b_{i,t} = \text{MHG}\left([x^{*S}_{i,t},\ x^{*E}_{i,t},\ x^{*I}_{i,t}],\ \sum_{j \in \omega} m_{i,j,t}\right)$
        $x^*_{i,t} = x^*_{i,t} - b_{i,t}$
        for $j \in \omega$ **do**
          $c_{i,j,t} = \text{MHG}\left(b_{i,t},\ m_{i,j,t}\right)$
          $b_{i,t} = b_{i,t} - c_{i,j,t}$
          $x^*_{j,t} = x^*_{j,t} + c_{i,j,t}$
        end
      end
    end
  Step 3: Retry farms that did not work
    while $R \neq R'$ **do**
      Set $R = R'$, $R' = \varnothing$
      for $i \in R$ **do**
        Do **Step 2:** $(\omega \to \omega)$
        and generate x^{**}
        Note: We make the assumption that eventually $R = R' = \varnothing$.
      end
    end
  Step 4: $(\omega \to \omega^c)$
    for $i \in \omega$ **do**
      $d_{i,t} = \text{MHG}\left([x^{**S}_{i,t},\ x^{**E}_{i,t},\ x^{**I}_{i,t}],\ \sum_{j \in \omega^c} m_{i,j,t}\right)$
      $x^{**}_{i,t} = x^{**}_{i,t} - d_{i,t}$
      for $j \in \omega$ **do**
        $e_{i,j,t} = \text{MHG}\left(d_{i,t},\ m_{i,j,t}\right)$
        $d_{i,t} = d_{i,t} - e_{i,j,t}$
        $x^{**}_{j,t} = x^{**}_{j,t} + e_{i,j,t}$
      end
    end
  $X' = x^{**}$
  Return X'

The number of cattle moved off of each farm i at time t, and their destination farm j, is set by the data. The random variable we are concerned with is the states of the animals moved. We can still calculate the likelihood using,

$$\begin{aligned}\mathbb{P}\left[M_{i,t} \mid X_{i,t}\right] = & \frac{m_{-1,i,t}!}{a^S_{-1,i,t}!\; a^E_{-1,i,t}!\; a^I_{-1,i,t}!} \cdot (0.9)^{a^S_{-1,i,t}} \cdot (0.05)^{a^E_{-1,i,t}} \cdot (0.05)^{a^I_{-1,i,t}} \\ & \times \frac{\binom{x^{*S}_{i,t}}{b^S_{i,t}}\binom{x^{*E}_{i,t}}{b^E_{i,t}}\binom{x^{*I}_{i,t}}{b^I_{i,t}}}{\binom{x^{*S}_{i,t}+x^{*E}_{i,t}+x^{*I}_{i,t}}{\sum_{j\in\omega} m_{i,j,t}}} \times \prod_{j\in\omega} \frac{\binom{b^S_{i,t}}{c^S_{i,t}}\binom{b^E_{i,t}}{c^E_{i,t}}\binom{b^I_{i,t}}{c^I_{i,t}}}{\binom{b^S_{i,t}+b^E_{i,t}+b^I_{i,t}}{m_{i,j,t}}} \\ & \times \frac{\binom{x^{**S}_{i,t}}{d^S_{i,t}}\binom{x^{**E}_{i,t}}{d^E_{i,t}}\binom{x^{**I}_{i,t}}{d^I_{i,t}}}{\binom{x^{**S}_{i,t}+x^{**E}_{i,t}+x^{**I}_{i,t}}{\sum_{j\in\omega^c} m_{i,j,t}}} \times \prod_{j\in\omega^c} \frac{\binom{d^S_{i,t}}{e^S_{i,t}}\binom{d^E_{i,t}}{e^E_{i,t}}\binom{d^I_{i,t}}{e^I_{i,t}}}{\binom{d^S_{i,t}+d^E_{i,t}+d^I_{i,t}}{m_{i,j,t}}}\end{aligned}$$

As in Chapter 5 the assumption for 5% of the external population being in the exposed and infectious state respectively is derived from the average herd breakdowns across the country.

6.4.1.3 The Environmental Kernel

The parish level environment, $V_{p(i),t}$ no longer depends on the level of badger infection in the parish, and is generated using Algorithm 14.

Algorithm 14: Generate the environmental reservoir for timestep $t+1$

Input : X'''' = Post births and deaths cattle states during timestep t,
V_t = Environmental reservoir at beginning of timestep t,
$\epsilon \in \theta$ = Model parameters.

Output : V' = Environmental reservoir at the end of timestep t.

Elements: $R_{p,t}$ = Remaining environmental pressure on farm i at the end of timestep t,
$N_{p,t}$ = New environmental pressure on farm i at the end of timestep t,
ω = The set of farms of interest, with p being the parish groupings.

$K_V(V_t,\ X'''')$::
  foreach $p \in \omega$ **do**
    $R_{p,t} = \text{Bin}\left(V_{p,t},\ 1-\epsilon\right)$
    $N_{p,t} = \text{Po}\left(\sum_p X_t''''\right)$
  end
  $V' = R_t + N_t$
  `Return` V'

The likelihood for this subprocess that generates the infectious pressure in the environmental reservoir for parish p at time t has the form of a Poisson distribution multiplied by a Binomial distribution. The environmental reservoir is generated by choosing a random amount to remain from the previous time step based on the decay rate, and a random amount to be added based on the infectious cattle in the parish. The probability of the environmental reservoir pressure is thus given by

$$\mathbb{P}\left[V_{p(i),(t+1)} \middle|\ V_{p(i),t},\ X''''\right] = \frac{e^{-\left(\sum_{i\in p} X''''_{i,t}\right)} \cdot \left(\sum_{i\in p} X''''_{i,t}\right)^{N_{p(i),t}}}{N_{p(i),t}!} \times \binom{V_{p(i),t}}{R_{p(i),t}} \cdot (1-\epsilon)^{R_{p(i),t}} \cdot (\epsilon)^{V_{p(i),t}-R_{p(i),t}}$$

It is worth noting that for all the farms in the same parish P the value of $V_{p(i),t}$ will be the same.

6.5 Likelihood and Posteriors

6.5.1 Likelihood

As in Chapter 5 the likelihood (previously Function (5.7.1)) can thus be broken down as the product over all timesteps $t \in 1:T$, with the removal of the dependence on the badger population, and assuming that all data is observed:

$$\begin{aligned}
&\mathcal{L}\Big(\mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{D}^c{}_t,\ \mathbf{V}_{t+1}\ | \\
&\qquad \mathbf{X}_t,\ \boldsymbol{\theta},\ \mathbf{V}_t,\ \mathbf{m}_t,\ \mathbf{test}_t,\ \mathbf{d}^c_t\Big) \\
&= \mathbb{P}\left[\mathbf{M}_t \mid \mathbf{X}_t,\ \mathbf{m}_t\right] \\
&\quad \times \mathbb{P}\left[\mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mid \mathbf{M}_t,\ \mathbf{X}_t,\ \boldsymbol{\theta},\ \mathbf{V}_t\right] \\
&\quad \times \mathbb{P}\left[\mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t \mid \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{test}_t,\ \mathbf{X}_t,\ \boldsymbol{\theta}\right] \\
&\quad \times \mathbb{P}\left[\mathbf{D}^c{}_t,\ \mid \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{test}_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{X}_t,\ \mathbf{d}^c_t\right] \\
&\quad \times \mathbb{P}\left[\mathbf{V}_{t+1}|\ \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{D}^c{}_t,\ \mathbf{X}_t,\ \boldsymbol{\theta},\ \mathbf{V}_t\right]
\end{aligned}$$

6.5.2 Posteriors

Using the likelihood derived in Section 6.5 we can derive the posterior distributions used in the MCMC algorithm explored in Section 6.7 by specifying priors for the parameters, and assuming all data is observed.

The full joint conditional posterior is given by

$$\begin{aligned}
\pi\left(\beta_c, \delta, \epsilon, F, \rho, \rho_E | \mathbf{X}, \mathbf{V}\right) = &\prod_{t=1}^{T} \mathcal{L}\Big(\mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{D}^c{}_t,\ \mathbf{V}_{t+1} \mid \mathbf{X}_t,\ \boldsymbol{\theta},\ \mathbf{V}_t,\ \mathbf{m}_t,\ \mathbf{test}_t,\ \mathbf{d}^c_t\Big) \\
&\times \pi\left(\beta_c\right) \times \pi\left(\delta\right) \times \pi\left(\epsilon\right) \times \pi\left(F\right) \times \pi\left(\rho\right) \times \pi\left(\rho_E\right)
\end{aligned}$$

where we are setting the priors to be:

- $\pi(\beta_c) \sim \mathsf{Gamma}(\vartheta_{\beta_c}, \sigma_{\beta_c})$
- $\pi(F) \sim \mathsf{Gamma}(\vartheta_F, \sigma_F)$
- $\pi(\delta) \sim \mathsf{Gamma}(\vartheta_\delta, \sigma_\delta)$
- $\pi(\rho) \sim \mathsf{Beta}(\vartheta_\rho, \sigma_\rho)$
- $\pi(\epsilon) \sim \mathsf{Gamma}(\vartheta_\epsilon, \sigma_\epsilon)$
- $\pi(\rho_E) \sim \mathsf{Beta}(\vartheta_{\rho_E}, \sigma_{\rho_E})$

The form of the Gamma distribution has $\vartheta > 0$ as the shape parameter, and $\sigma > 0$ as the rate parameter. The gamma priors align with the positive real support of the parameters, whilst also being malleable to adapt to weak or strong prior knowledge. The detection parameters, ρ (a probability) and ρ_E (a scalar in [0,1]), both have support on the real line between 0 and 1, so a Beta prior is an appropriate choice, combined with it's malleability.

6.5.2.1 The Infection Process Parameters; $[\beta_c, \delta, F, \epsilon]$

The conditional joint posterior likelihood of the infection process parameters is given by the product for $t \in 1:T$ of:

$$\begin{aligned}\pi(\beta_c, \delta, \epsilon, F | \mathbf{X}_t, \mathbf{V}_t) =& \mathbb{P}\left[\mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ |\ \mathbf{M}_t,\ \mathbf{X}_t,\ \boldsymbol{\theta}\right] \\ & \times \mathbb{P}\left[\mathbf{V}_{t+1} |\ \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{test}_t,\ \mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t,\ \mathbf{D}^c{}_t,\ \mathbf{X}_t,\ \boldsymbol{\theta},\ \mathbf{V}_t\right] \\ & \times \pi(\beta_c) \times \pi(\delta) \times \pi(\epsilon) \times \pi(F)\,.\end{aligned}$$

6.5.2.2 The Detection Process Parameters; $[\rho, \rho_E]$

The conditional joint posterior likelihood of the detection process parameters is given by the product for $t \in 1:T$ of:

$$\begin{aligned}\pi(\rho, \rho_E | \mathbf{X}_t, \mathbf{V}_t) =& \mathbb{P}\left[\mathbf{H}^E{}_t,\ \mathbf{H}^I{}_t\ |\ \mathbf{M}_t,\ \mathbf{dE}^c_t,\ \mathbf{dI}^c_t,\ \mathbf{test}_t,\ \mathbf{X}_t,\ \boldsymbol{\theta}\right] \\ & \times \pi(\rho) \times \pi(\rho_E)\end{aligned}$$

6.5.3 Observed data, latent variables, and parameters

In this chapter the data is that is observed and the data that are latent variables is defined for us by the problem, though the core idea remains the same. We know the number of animals on each farm, the number of movement, test, birth, and death events on each farm, and the number of cattle sent to slaughter (a proxy for detection), but we don't know the states of the animals at any time, including the initial states, or that of the environmental reservoir.

Again we categorise the model components into one of three classes which are relevant to the fitting process; data is observed and used the fit the model, latent variables are unobserved but inferred using data augmentation during the MCMC process, and parameters are inferred through the MCMC process. In this case all of the data is provided by APHA, and none is simulated.

The data that we are fitting the model to include, for each farm i at time t in our set of farms of interest, ω: The number of cattle ($\mathbb{C}_{i,t}$), the number of cattle movements from farm i to j ($m_{i,j,t}$), whether or not it is a test day ($\text{test}_{i,t}$), assuming all cattle are tested, the number of cattle that were sent to slaughter ($H_{i,t}$), the number of cattle births ($b^c_{i,t}$), and the number of cattle deaths ($d^c_{i,t}$).

The latent variables that we infer through data augmentation include, for each farm i at time t in our set of farms of interest, ω: The initial states of the cattle ($\mathbf{X}_{i,0}$), the initial level of environmental infection ($V_{p(i),0}$), the states of the animals that moved to each destination ($M_{i,j,t}$), the number of newly exposed and infectious cattle ($dE^c_{i,t}$, $dI^c_{i,t}$), the number of detected exposed and infectious cattle from testing ($H^E_{i,t}$, $H^I_{i,t}$), the states of the cattle that died ($D^c_{i,t}$), and the additional and remaining environmental pressure ($N_{p(i),t}$, $R_{p(i),t}$).

The model is parameterised by 6 parameters; $[\beta_c, \delta, F, \epsilon]$ relating to the infection process, and $[\rho, \rho_E]$ relating to the testing process.

Thus we can again reformulate the posterior presented in Section 6.5.2 to distinguish between the observed data, the latent variables, and the parameters. Denote the parameters $\boldsymbol{\theta}$. For the product of times t in $1:T$:

$$\begin{aligned}\pi\left(\boldsymbol{\theta} \mid \mathbf{X}_t, \mathbf{V}_t\right) =& \mathcal{L}\Big(\mathbb{C},\ \mathbf{m},\ \mathbf{test},\ \mathbf{H},\ \mathbf{b}^c,\ \mathbf{d}^c,\ \mid \\ & \qquad \mathbf{X}_0,\ \mathbf{V}_0,\ \mathbf{M},\ \mathbf{dE}^c,\ \mathbf{dI}^c,\ \mathbf{H}^E,\ \mathbf{H}^I,\ \mathbf{D}^c,\ \mathbf{N},\ \mathbf{R},\ \boldsymbol{\theta}\Big) \\ & \times \pi\left(\mathbf{X}_0,\ \mathbf{V}_0,\ \mathbf{M},\ \mathbf{dE}^c,\ \mathbf{dI}^c,\ \mathbf{H}^E,\ \mathbf{H}^I,\ \mathbf{D}^c,\ \mathbf{N},\ \mathbf{R} \mid \boldsymbol{\theta}\right) \\ & \times \pi\left(\boldsymbol{\theta}\right)\end{aligned}$$

6.5.3.1 Data Augmentation

For missing event data, as demonstrated in chapter 3, if we augment the data using the 'moving an event in time' method then we will need to calculate all posterior terms between the time the event moved from to the time an event moved to. If we augment the data using the 'adding/removing an event' method, then well will need to calculate all posterior terms from the time the event was added to the end of the process.

6.6 Initialising the MCMC

In this section we discuss an overview of how we generated a valid epidemic data set from the full data to initialise the MCMC. The challenge remains that the discretisation and artefacts in the data leads to inconsistencies and misalignment's, however this time we cannot discard data that doesn't work, and instead have to assume an issue with the initial conditions or randomly generated events or states. There are two processes where this is cause of concern - the movement data and testing data.

Movements can fail when there are less animals on a farm than the data says are moving. This can be due to other movements needing to occur first, or because of issues with the initial conditions. As in chapter 5 we loop through all movements, updating the intermediate states until all moves have been processed. In this case, if there remains a set of movements that are unable to be processed, instead of

discarding them we alter the initial conditions of the epidemic to add the minimum number of additional animals to make these movements valid, and propagate the changes forward. The animals are added to the susceptible class to minimise their effect on the epidemic.

Similarly, detection events fail when there are insufficient exposed and infectious animals on the farm. This can occur for multiple reasons, mostly due to the random additions and removals of infected animals. For instance in reality a susceptible animal was moved, but since we have no data on the states of the animals, we generated that an exposed animal was moved, which resulted in that animal not being present on the farm when testing occurred. There are multiple ways of dealing with this challenge, but we chose a simple method that would take minimal computation to identify a valid epidemic. After we generate the number of newly exposed and infectious animals, if there is a detection event coming up in the near future and there currently are not enough animals to detect, then replace the generated events when valid with another set that guarantees enough animals will be available to detect. This will lead to an epidemic that does not follow the epidemic generating process, but it will have a non-zero likelihood. As such after a potentially longer burn-in period, the data augmentation steps of the MCMC algorithm should be able to move to an areas of higher posterior mass and a more likely epidemic. This method to initialise the epidemic worked, though improvement is possible to reduce the computational burden of the MCMC.

6.7 MCMC Algorithm

The MCMC schema closely matches that of Chapter 5, with some notable changes which can be summarised as removing any dependence on badger data, and introducing methods to augment the initial conditions.

Firstly we naturally change the parameters that are being inferred due to the removal of badger data, but otherwise the methodology stays the same, with two

sets; infection parameters and detection parameters.

For the data augmentation, all of the steps relating to cattle remain, and all of the steps specifically relating to badgers are removed, however an additional set of data augmentations are introduced. In the previous chapter we assumed that the initial conditions of the epidemic were accurate when making inference, and in that case it was true. In fact the inference was initialised with the true data. In the case of this real data however, we are not aware of the true initial conditions, and the inference is initialised with an epidemic that is merely valid in the sense that it has a non-zero likelihood, and is unlikely to have a high posterior likelihood.

The new class of data augmentation step we introduce concerns updating the initial conditions of the epidemic. That is, the states of the cattle on every farm at the beginning of time step $t = 1$, and the level of background infection present in each parish. There are 7 new data augmentation steps introduced, 6 of which concern changing the initial state of animals on a farm, and one relates to update the level of environmental reservoir in a parish.

For the changing of cattle states, the first updater is changing a Susceptible (S) cattle to an Exposed (E) cattle. This change affects all the states of the farm for the entire during of the epidemic. This then requires that the likelihood of every event in the epidemic for that farm be recalculated. The second update step concerns changing a Susceptible (S) cattle to and infected (I) cattle. This change also results in changing all the states and recalculating the full likelihood in that farm. However in addition the new infected cattle will contribute infectious pressure during the epidemic, and will alter the likelihood of the new pressure added to the parish environmental reservoir each timestep, so the likelihood of the parish environmental reservoir will also need to be recalculated. The other augmentation steps are: changing an Exposed (E) to a Susceptible (S) which requires the same updates as the former example, and changing an Exposed (E) to an infected (I), an infected (I) to a Susceptible (S), and an infected (I) to an Exposed (E), all of which require the same changes as the latter example due to changing the number

of infected individuals.

6.8 Results

We made inference on the parameters using the block adaptive MCMC methodology, grouping the parameters into two groups; the infection process parameters $[\beta_c, \delta, F, \epsilon]$ and the detection process parameters $[\rho, \rho_E]$. We focused our inference on 307 farms, in 24 parishes, in the county of Cheshire, for the 360 weeks from the 1st January 2012. The parishes are shown in Figure 6.1.

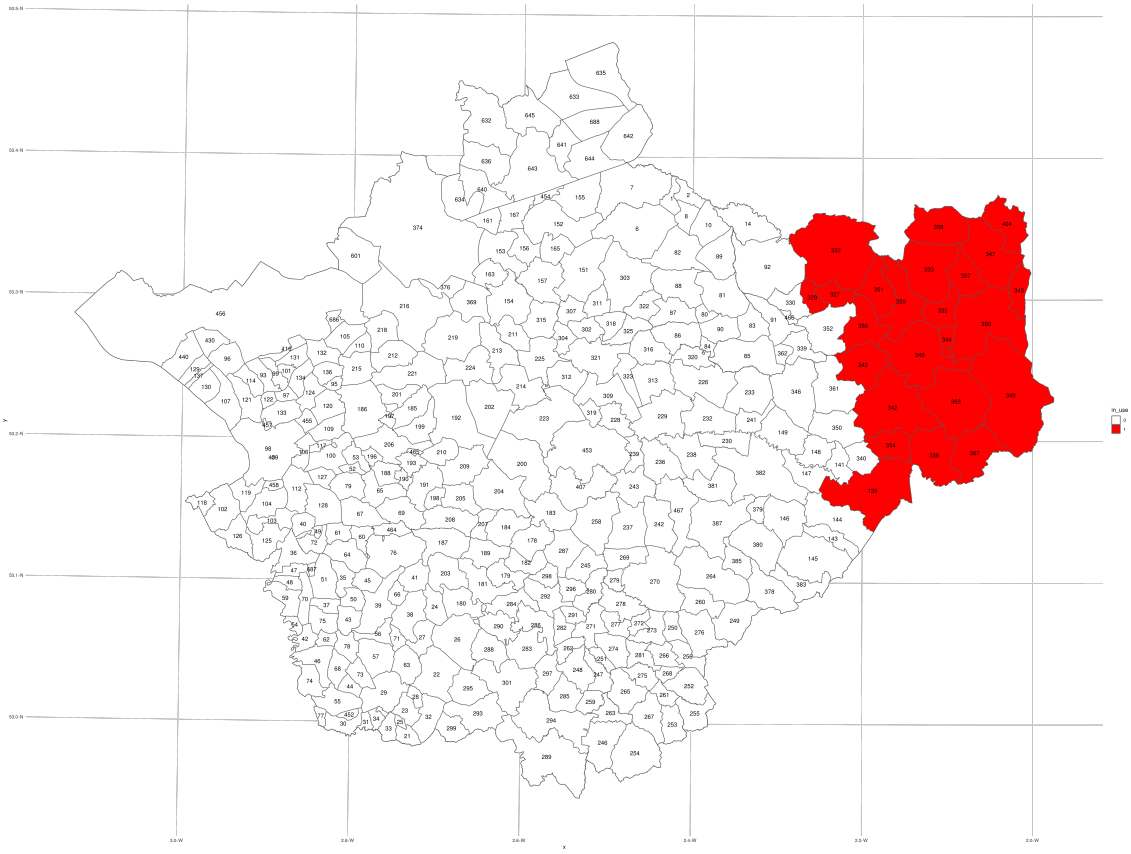


Figure 6.1: A map of the parishes in Cheshire that are used in the inference of the real data parameters.

Below we present two sets of inference. The first is inference on the full set of parameters, however the algorithm clearly struggled to identify the detection parameters. As a result we ran another inference set with the detection parameters fixed at the values identified by Brooks-Pollock, Roberts, and Keeling, 2014; $\rho = 0.72$ and $\rho_E = 0.276$. We discuss why this issue may have occurred in Section 6.9.

The priors were set to be:

- $\beta_c \sim \mathsf{Gamma}(2, 0.001)$
- $F \sim \mathsf{Gamma}(2, 0.002)$
- $\rho \sim \mathsf{Beta}(1.5, 0.5)$
- $\delta \sim \mathsf{Gamma}(3, 0.005)$
- $\epsilon \sim \mathsf{Gamma}(1, 0.05)$
- $\rho_E \sim \mathsf{Beta}(0.4, 1.6)$

6.8.1 Infection and Testing Parameters

The below results are the output of $900,000$ samples after burn-in. The following table presents the summaries of the marginal posterior distributions:

	Mean	95% CI	Std. Dev.	ESS
β_c	0.00302	(0.00232, 0.00362)	0.0003390	28.6
δ	0.0224	(0.0207, 0.0243)	0.0009090	1073.0
F	0.00151	(0.00132, 0.00169)	0.0000979	24.9
ϵ	0.0529	(0.0513, 0.0545)	0.0008220	41.8
ρ	0.935	(0.901, 0.955)	0.0156000	199.7
ρ_E	0.997	(0.991, 1.000)	0.0024900	12258.0

Table 6.2: The summary of the marginal posterior distributions.

Overall the algorithm recovered tight uni-modal and symmetric posteriors around the infection parameters, but struggled to identify the detection parameters, pushing each to their upper limit of 1, as we can see in Figures 6.2 and 6.3. Trace plots of the chains also show a trend towards 1. Given perfect sensitivity for the tests, these results for the epidemic parameters are reasonable, however we know that the tests do not have perfect sensitivity.

As seen in Figure 6.4 the mixing for δ was good, with large jumps and time spent exploring the posterior mass, however, the algorithm struggled with all the other parameters, with thin chains that slowly explored the posterior. The algorithm also took a long time to converge, which we suspect is due to a very low posterior likelihood for the initial conditions, and the number of update steps required to move the chain towards higher posterior mass.

Figures 6.6 and 6.7 show the pairwise contour plots for all parameters in the infection process set, and then the detection process set. The yellow dotted lines represent the position of the pair of parameters with the highest posterior mass.

The correlation between the parameters does not present strongly in these plots, though there was some minor correlation present between the chains of β_c, δ, and F.

The effective sample sizes for each of the parameters is given in Table 6.2. The average acceptance rate of the infection parameters was 29.57%. The average acceptance rate of the detection parameters was 28.65%.

For the data augmentation, the average acceptance rate for: moving S to E exposure events was 89.47%, moving E to I infection events was 77.08%, adding or removing S to E exposure events was 0.42%, adding or removing E to I infection events was 1.12%, exchanging detection events was 0.12%, exchanging cattle death events was 1.54%, adding or removing environmental reservoir pressure was 43.64%, and exchanging movement events was 3.12%.

6.8.2 The Infection Parameters

The below results are the output of $900,000$ samples after burn-in for the inference run with ρ fixed at 0.72 and ρ_E fixed at 0.276. The following table presents the summaries of the marginal posterior distributions:

	Mean	95% CI	Std. Dev.	ESS
β_c	0.00520	(0.00449, 0.00579)	0.000335	154.1
δ	0.00531	(0.00492, 0.00567)	0.000193	1004.8
F	0.00116	(0.00106, 0.00128)	0.000056	91.6
ϵ	0.05258	(0.0514, 0.0540)	0.000707	98.2

Table 6.3: The summary of the marginal posterior distributions.

Overall the algorithm recovered tight uni-modal and symmetric posteriors around the infection parameters, as we can see in Figure 6.8, however the effective sample sizes are very small, and the mixing was poor as can be seen in Figure 6.9, suggesting high levels of autocorrelation. With a lower sensitivity test the detected animals will make up a smaller proportion of the total infected population, and by consequence it implies an epidemic with greater numbers of infected animals. As such there is a

noticeable increase in the estimate of β_c, leading to more infections. In particular δ has a mean 4 times smaller than in the previous inference, leading to much longer waiting times for the onset of infection. Interestingly F did not increase as much as β_c, meaning the cattle to cattle infectious played a greater role than the environment. The variance in the posteriors remains about equal or is slightly lower. These values seem reasonable.

The mixing for δ was still good, and the mixing for β_c showed a small improvement, but overall the mixing of the algorithm was still poor and took a long time to converge. This again could be due to the starting conditions and tuning of the update steps.

Figure 6.10 show the pairwise contour plots for all parameters in the infection process set. The yellow dotted lines represent the position of the pair of parameters with the highest posterior mass. The correlation between the parameters does not present strongly in these plots.

The effective sample sizes for each of the parameters is given in Table 6.3. The average acceptance rate of the infection parameters was 30.1%.

For the data augmentation, the average acceptance rate for: moving S to E exposure events was 91.79%, moving E to I infection events was 78.68%, adding or removing S to E exposure events was 1.48%, adding or removing E to I infection events was 0.75%, exchanging detection events was 0.80%, exchanging cattle death events was 4.41%, adding or removing environmental reservoir pressure was 43.73%, and exchanging movement events was 6.39%.

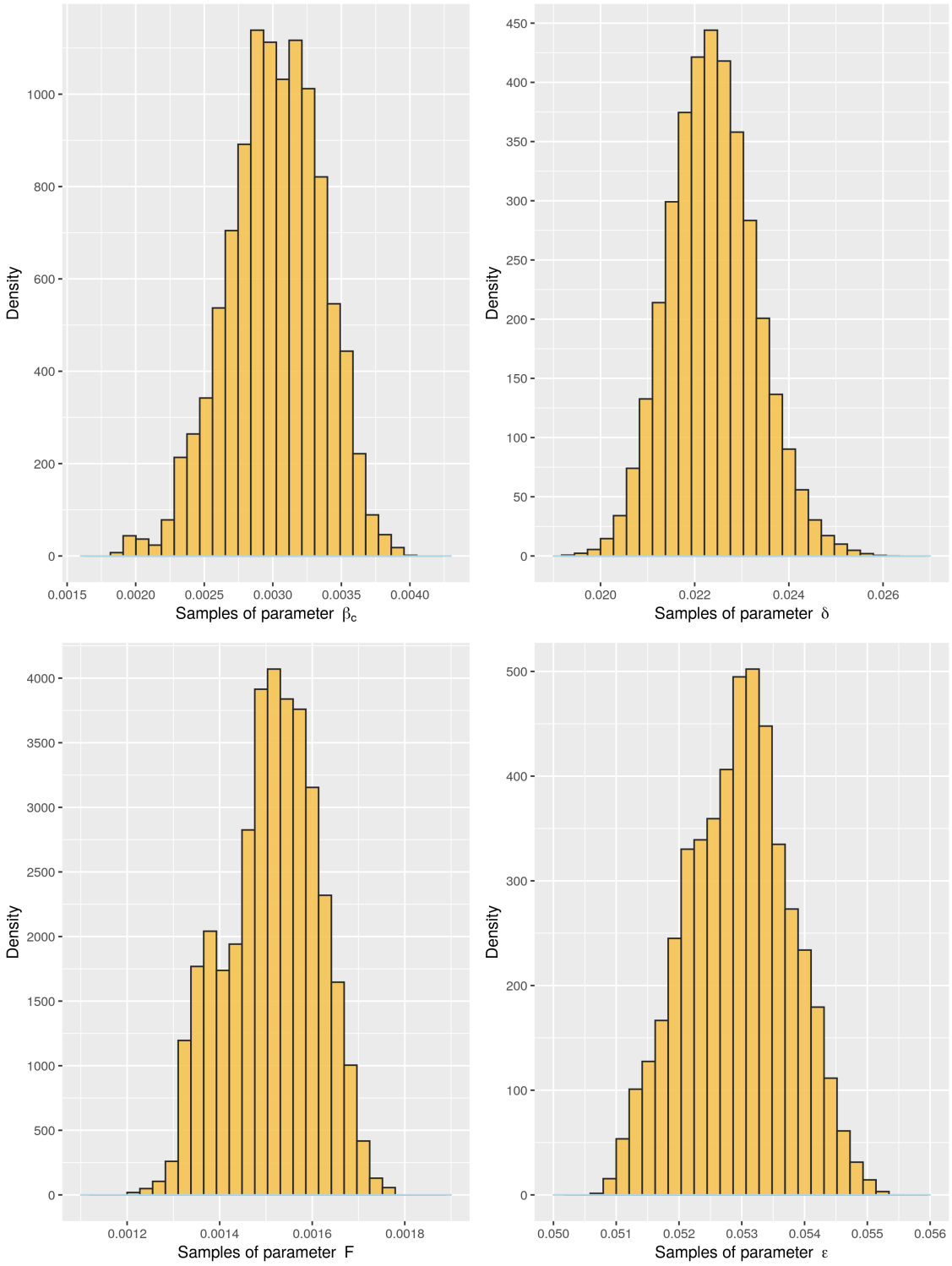


Figure 6.2: Results: The posterior samples of the infection process parameters displayed as their marginal distributions represented in a histogram. The priors are shown in blue.

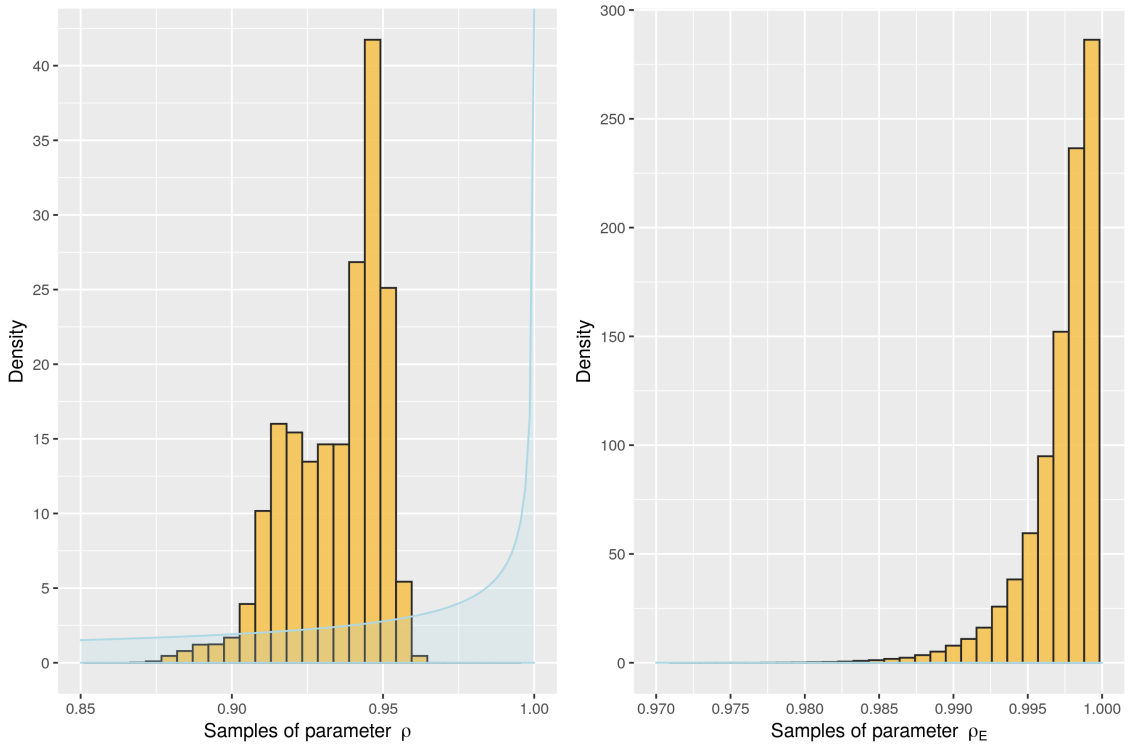


Figure 6.3: Results: The posterior samples of the detection process parameters displayed as their marginal distributions represented in a histogram. The priors are shown in blue.

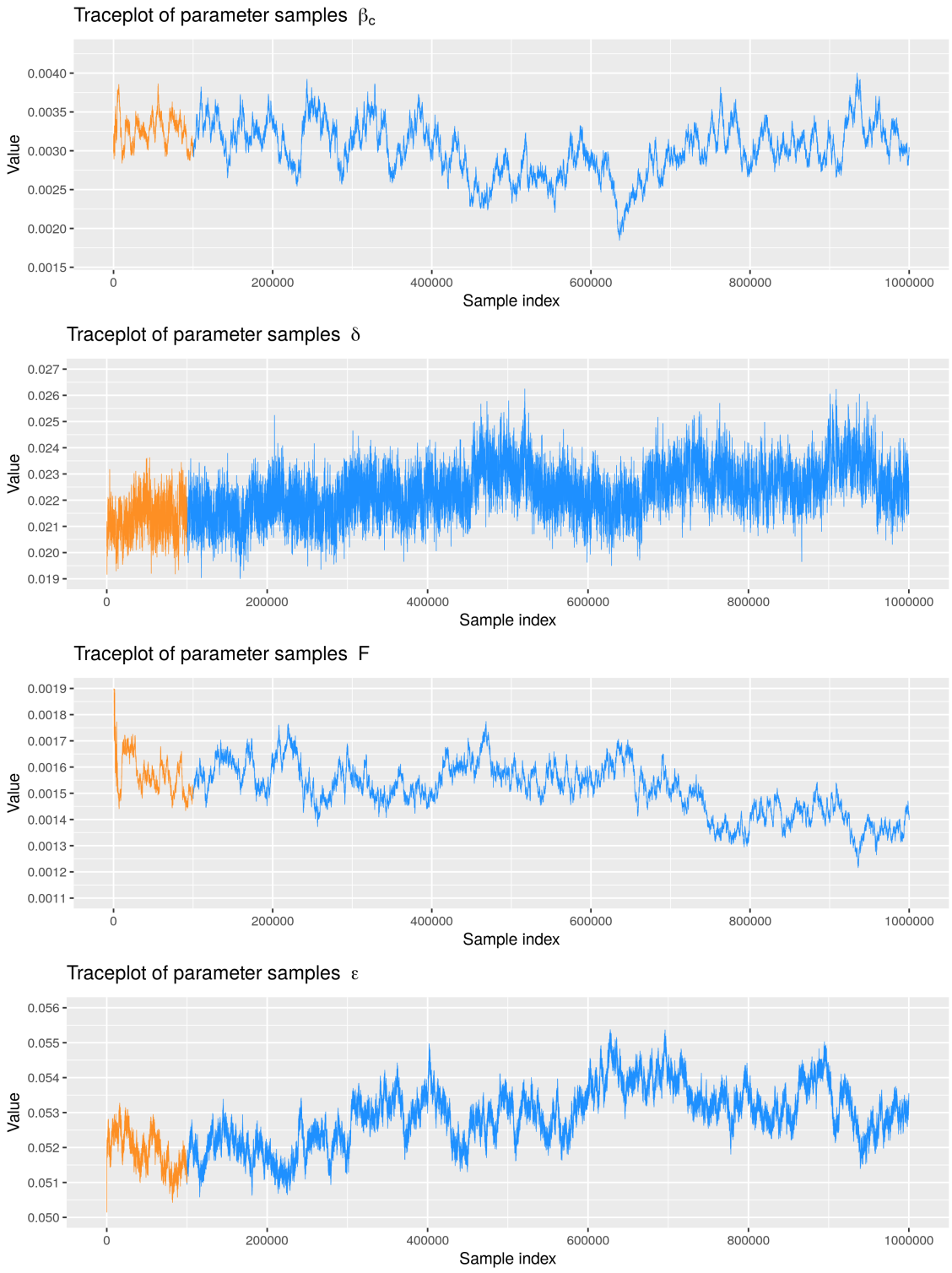


Figure 6.4: Results: Trace plot for β, δ, F, and ϵ. The orange area represents a portion of the burn-in.

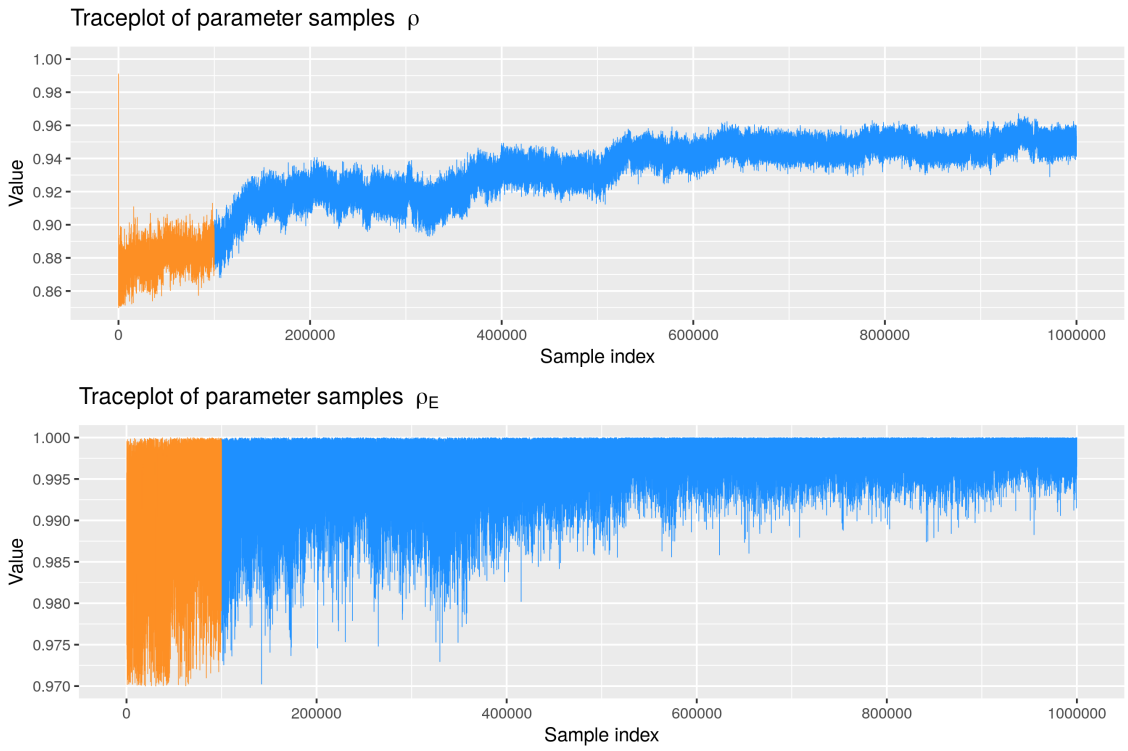


Figure 6.5: Results: Trace plot for ρ and ρ_E. The orange area represents a portion of the burn-in.

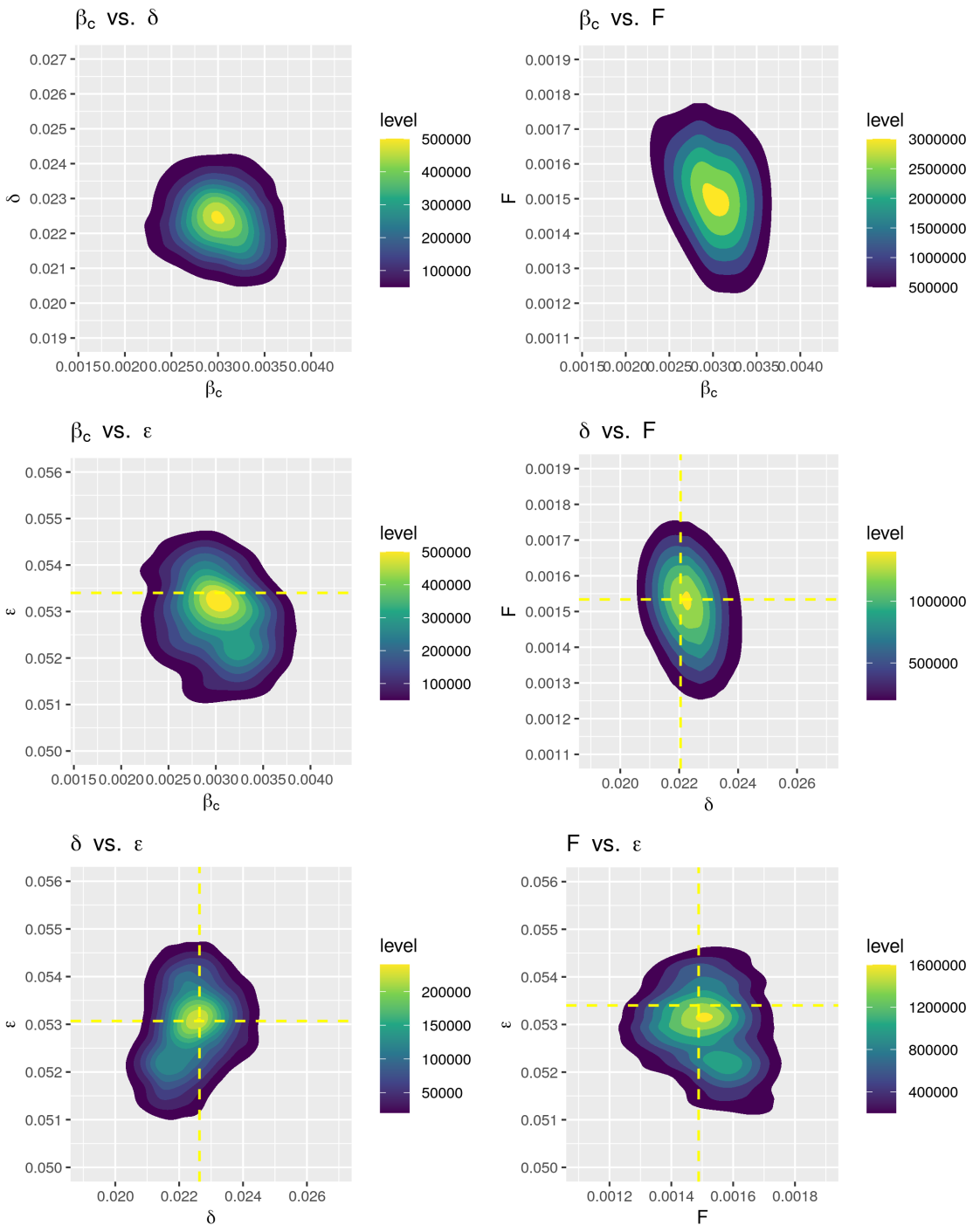


Figure 6.6: Results: Contour plots of the posterior samples for each pair of the infection parameters. Brighter contours represent areas of higher density. The yellow dashed lines show the pair-wise point of highest density on the contour plots. From left to right, top to bottom, the plots show β *vs* δ, β *vs* F, β *vs* ϵ, δ *vs* F, δ *vs* ϵ, and F *vs* ϵ.

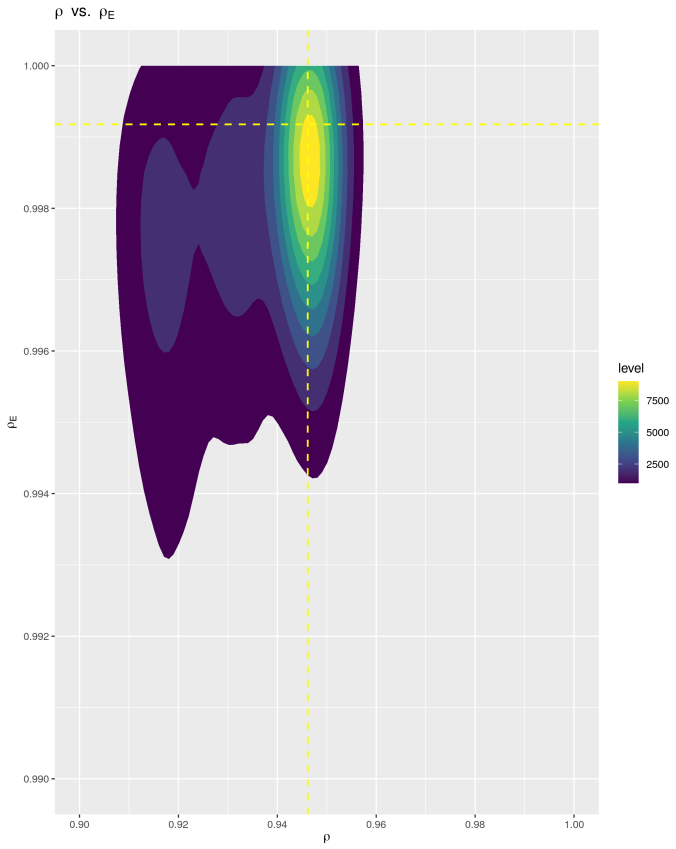


Figure 6.7: Results: Contour plots of the posterior samples for the detection parameters, ρ *vs* ρ_E. Brighter contours represent areas of higher density. The yellow dashed lines show the pair-wise point of highest density on the contour plots.

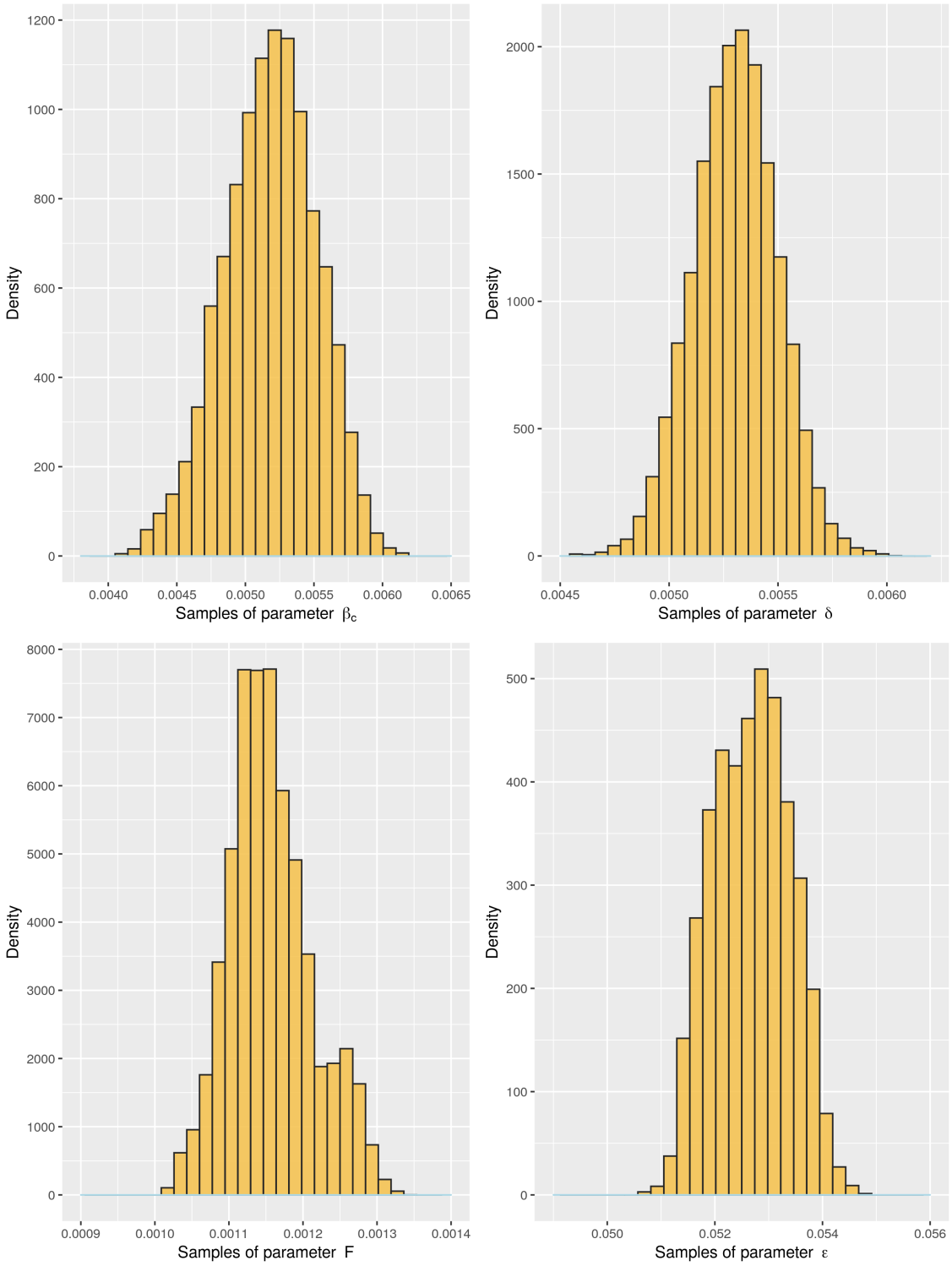


Figure 6.8: Results: The posterior samples of the infection process parameters, assuming the detection parameters known, displayed as their marginal distributions represented in a histogram. The priors are shown in blue.

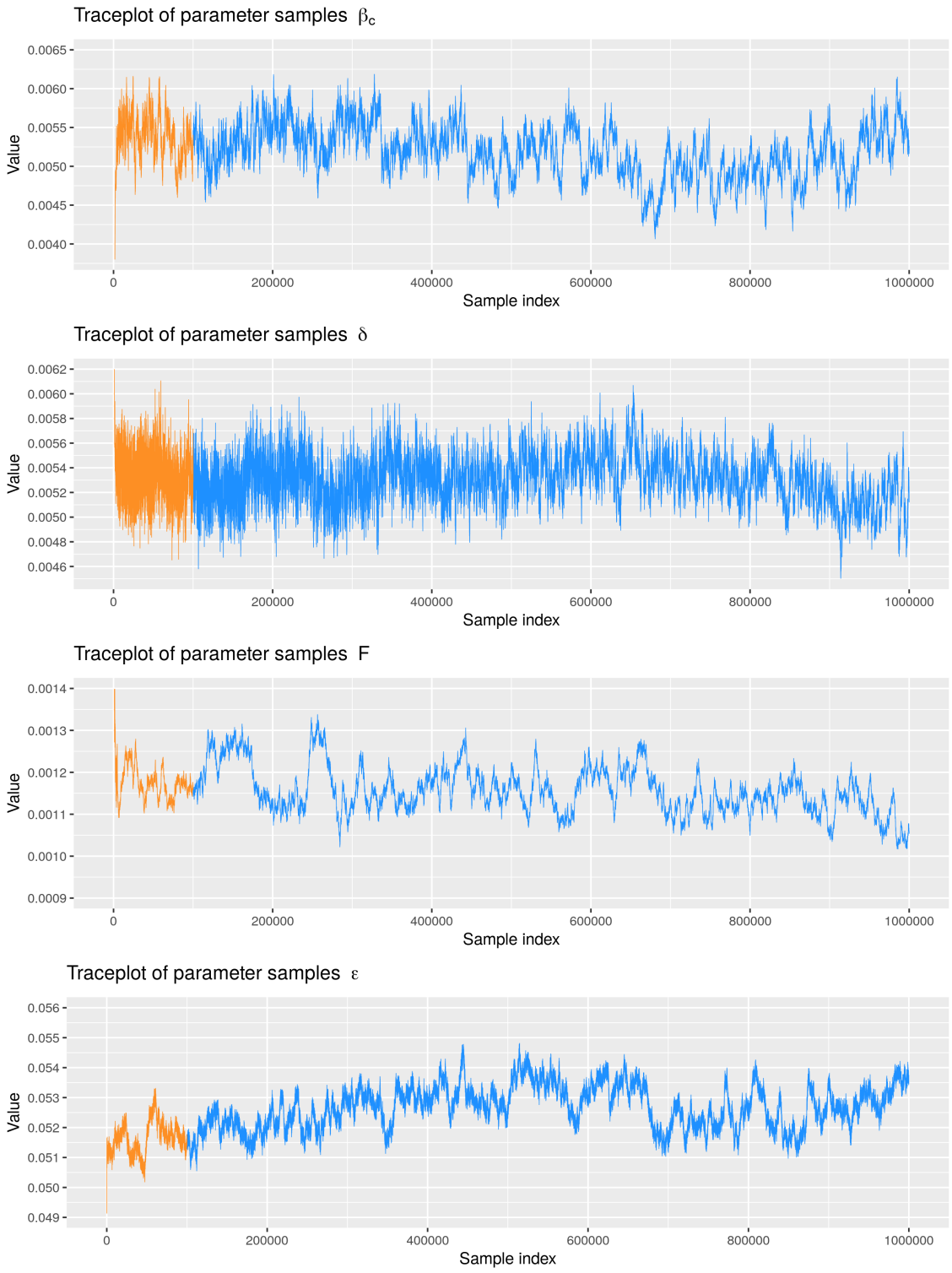


Figure 6.9: Results: Trace plot for β, δ, F, and ϵ, assuming the detection parameters known. The orange area represents a portion of the burn-in.

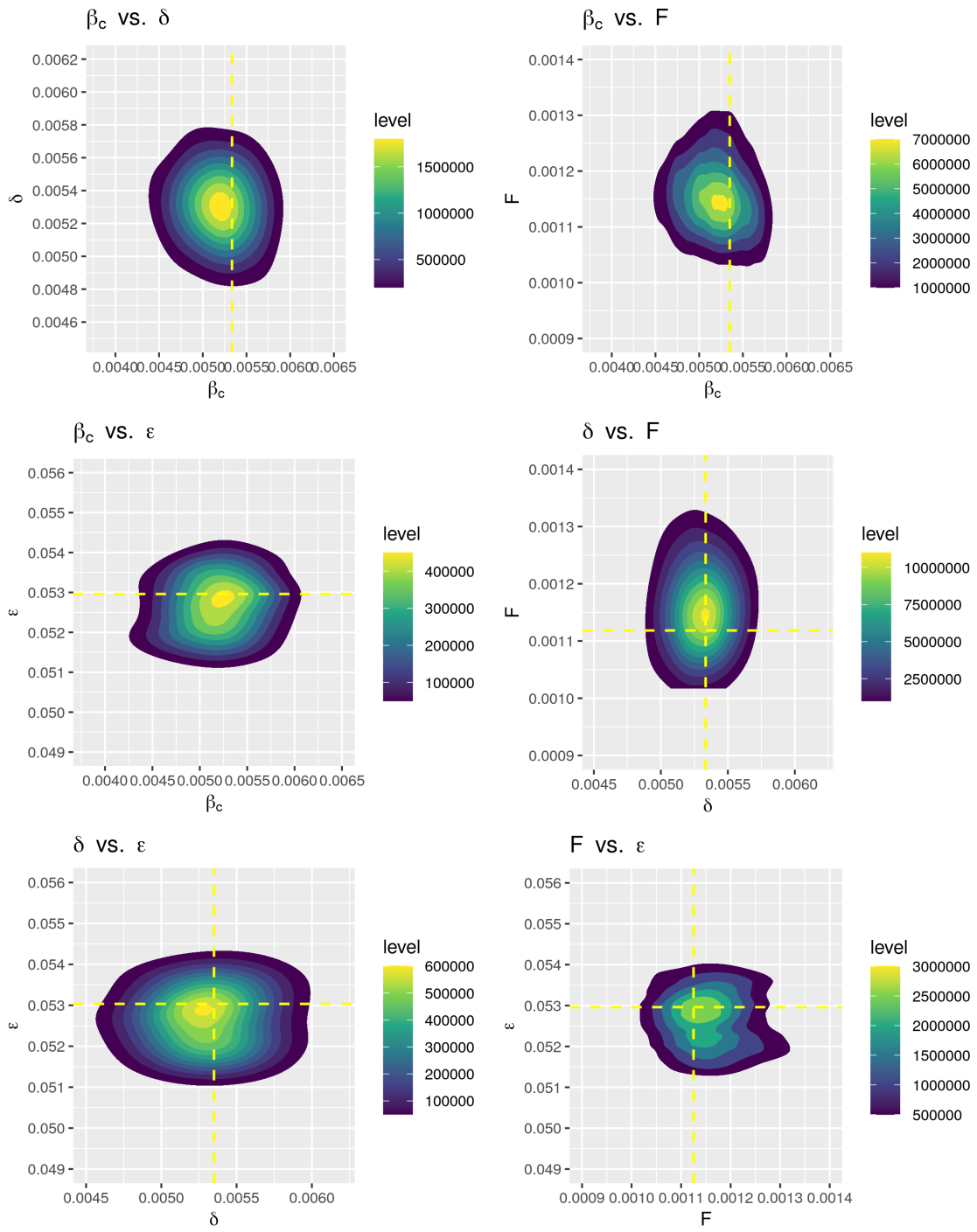


Figure 6.10: Results: Contour plots of the posterior samples for each pair of the infection parameters, assuming detection parameters known. Brighter contours represent areas of higher density. The yellow dashed lines show the pair-wise point of highest density on the contour plots. From left to right, top to bottom, the plots show β *vs* δ, β *vs* F, β *vs* ϵ, δ *vs* F, δ *vs* ϵ, and F *vs* ϵ.

6.9 Discussion

Overall we believe this algorithm performed well, returning reasonable posterior distributions for the infection parameters with, in essence, the detection parameters roughly fixed at particularly values. This lends credence to the feasibility of discrete level models for making computationally efficient inference on epidemic data of this scale. There are some notable aspects for improvement however.

Despite the code being written in a language optimised for speed, Julia, and then being further optimised to the best of our ability, the inference still took roughly 5 to 7 days per million samples to run for 300 farms. There are in actuality over 80,000 farms in England and Wales that housed cattle. Not only would this increase the time taken for a million samples to over 1250 days assuming linear scaling with the number of farms, the current burn-in was 4 million samples to augment the initial conditions. Making likelihood calculations on the full country with the current implementation impossible. A faster calculation of the likelihood and more efficient updating of the data both from an MCMC perspective and implementation perspective would be necessary. In addition parallel updates and calculations would likely be necessary where possible.

Further to this issue, with the current data augmentation methods and tuning, the burn-in would not scale linearly, as a single change at a single farm at a single timestep will have less and less effect on the likelihood as the number of farms increases. This will be the greatest challenge of making inference for data of this scale.

Some of these issues can be addressed by starting with better initial conditions. The method chosen to create the initial conditions worked, but had a noticeable effect on the effectiveness and accuracy of the algorithm. Because we added exposed animals manually to match detection events it means that a large proportion of detected animals were exposed and all animals added were detected. Both of these choices inflated the posterior means of the detection parameters for the initial conditions, which the algorithm found it very difficult to escape from, which we

think is a large part of the reason for the parameters tending to 1. It will be especially challenging given the complexity of the process, but a better method of simulating an epidemic condition on the events in the data that follows the data generating process more accurately will likely be vital for practical applications of these algorithms.

These issues can also be addressed through improvements to the data augmentation process. Taking lessons from the previous chapter, the data augmentation steps were adapted to reduce the number of proposals that resulted in no change, however, the tuning remained the same. This means that for each data augmentation action the minimum change possible was made. This is likely a contributing factor to the poor effective sample size and mixing of the parameters. Better tuning could help for moving events in time, however given the low acceptance rates of many of the other proposal functions it is likely they would need to be replaced. We would need something more intelligent, that considered more data to make proposals that were more likely to be accepted. The challenge derives from the complexity of the data generating process and the interconnectedness of farms and effect of propagating changes. The long time frames of the disease and relatively low level of infection within farms results in very small numbers of valid changes at any given data point. A more effective method may regenerate whole chains of events in a way that is valid. There is a wealth of potential research that can be explored here.

There are still a number of modelling decisions that could be reconsidered in future work. For instance we found it challenging to match the number of tests, the number of cattle detected/sent to slaughter, and the populations of the farms whilst working at this population level. As a result we made the assumption that all animals on the farm were tested when a Whole Herd Test (WHT) occurred, however this is not strictly true in reality. The result would mean that in a well initialised epidemic, we would essentially find the same number of infectives whilst testing more animals, and so the sensitivity of the test may be underestimated.

Finally, though the two can't be directly compared, it is clear with comparisons

to Chapter 5 that the badger data added a significant grounding element to the data inference. The additional data on badgers would be extremely effective in helping to make inference on the cattle parameters, given it were collected and available.

Chapter 7

Conclusion

In this thesis we have demonstrated the possibility of accurate and efficient full likelihood inference for complex big data epidemics.

We began in Chapter 1 by exploring the standard methods for the common form of epidemic data, describing the S-I-R epidemic model, and Markov Chain Monte Carlo for making inference. At the end of this chapter we stated the two main challenges we wish to address in this thesis - 1) The huge computational burden of making full likelihood inference for the Heterogeneous General Stochastic Epidemic, and 2) The reducing efficiency of the algorithm as the epidemic (and thus missing data) increases in scale and complexity.

In Chapter 2 we introduced our first method to potentially address these challenges. We introduced the Near vs Far model, a novel heterogeneous General Stochastic Epidemic for individuals on a plane with x and y coordinates as covariates. The model attempted to reduce the complexity of the epidemic by discretising the distance kernel with relation to the infection rate into two rates; one for individuals that were 'near' an infected individual, and another for individuals that were 'far' from an infected individual. This is a simplification of reality, and as such introduced inaccuracies. Our goal was to assess whether we could still make accurate inference whilst also increasing efficiency and reducing computational burden. Based on inference for a simulated epidemic it was shown that it was possible to make accurate

inference, and furthermore the parameterisation of the model had a noticeable effect of the efficiency of the algorithm. It was noted at the end of the chapter, however, that this model would still be incapable of making efficient and timely inference for epidemic data of the scale of modern epidemics and pandemics such as Bovine Tuberculosis and COVID-19.

To address these challenges, in Chapter 3 we introduced the methodology of the S-E-I-R Chain Binomial epidemic model - a model in discrete time, that concerns itself not with individuals but only with the counts of individuals in each state during each timestep. This model has the potential to vastly reduce the complexity of the likelihood, though there was a question of what size of discrete timestep would be appropriate to maintain accuracy. We also introduced extensions to the MCMC methodology, including an automatic adaptive tuning algorithm, block updates of parameters, and new proposal distributions associated with the discrete Chain Binomial construction. We simulated an epidemic and discretised it at four levels, making inference on each level. We found that with an appropriate level of discretisation, we could vastly reduce the computational burden whilst still generating accurate inference, but over-discretising lead to wild inaccuracies. The inference methods could be improved through tuning and more efficient proposal functions, but they succeeded in showing that accurate inference was possible for larger more complex epidemics. With this in mind we wished to apply these methods to a real big data epidemic dataset.

We chose Bovine Tuberculosis as our case study example of a big data epidemic, and introduced the context in Chapter 4. We reviewed the literature around the dynamics of the disease, and the work of Brooks-Pollock, Roberts, and Keeling, 2014 which inspired the model we developed in Chapters 5 and 6. We received data from the APHA which consisted of over 150 million rows of records, for 20 million cattle across $80,000$ farms. We presented an exploratory analysis of this data, which combined with our review of the literature, we used to guide the development of an epidemic model for Bovine Tuberculosis in England and Wales.

In Chapter 5 we developed the overview and intricacies of our model for Bovine Tuberculosis, assuming data on badger populations in addition to the cattle data provided by APHA. The model is a discrete-time meta-population SEI epidemic model, with a movement subprocess, a test and slaughter subprocess, and a background environmental infection subprocess, in two animal populations (cattle and badgers). We used the cattle data from APHA and a simulated badger population with its own SEIR process to generate a partially-simulated epidemic data set representative of the Bovine Tuberculosis epidemic in England and Wales, using a data generating process that we developed, including a novel method of processing cattle movement data. We then extended the MCMC methodologies introduced in Chapter 3 and made full likelihood inference on this epidemic data set for a subpopulation of Cheshire, assuming the badger data to be known. We showed that given detailed badger data, it was possible to make accurate inference for the epidemic in cattle and badgers for this data set. In reality the badger data is currently unavailable, so with the method validated we wished to apply it to as complete a set of the real data as possible.

Finally, in Chapter 6, we combined the learnings of all the previous chapters, and modified the work of Chapter 5 to make full likelihood inference for a subset of parishes in Cheshire using as much of the real data as possible. We removed from the model the dependence on the badger data, and developed a method of initialising the MCMC with a valid epidemic data set that contained all of the observed data. To the MCMC we introduced new data augmentation steps that updated the initial conditions of the epidemic. We ran two sets of inference. The first made inference on all 6 of the parameters, and struggled to identify the detection probabilities, which can be interpreted as the specificity of the tests. We believe we can attribute this in part to the initial conditions of the epidemic, and that starting the inference with data that produces a higher posterior mass will improve the efficiency of the inference. This said, we can interpret these results as those assuming a test of perfect specificity, and under that assumption the results looked reasonable.

The literature shows that the test does not have perfect specificity however, so we then ran an additional inference with the detection parameters fixed at those inferred by Brooks-Pollock, Roberts, and Keeling, 2014 which aligned well with the literature. This inference also provided reasonable inference. The algorithm has room for improvement through better tuning and more efficient proposal algorithms, and it is unlikely to scale to make efficient inference on the scale of the whole country, and the implementation would take years to run. That said, we have shown that it is possible to make accurate full likelihood inference for a large scale epidemic data, and that these methods are worth further investigation.

Future work should be concerned with increasing both the computational efficiency and methodological efficiency of these inference methods. For the computational efficiency, we implemented this algorithm in Julia, a language designed for its speed and aptitude with complex processes. Without the capabilities of this language, or another high speed language such as C, we would not have been able to achieve the computational efficiency that we did. However, it is clear that these implementations are not capable of scaling to the level of inference we are interested in. The most likely path of investigation is that of Graphical Processing unit (GPU) powered methods, allowing parallelisation that will vastly improve the speed of the algorithm, and possible the efficiency of the inference if implemented in line with inference methods that can take advantage of the parallelisation (Funk and King, 2020). Such methods have been effectively used to make inference on the recent COVID-19 pandemic (Jewell et al., 2023).

Methodological efficiency can be improved on two fronts. The first is by utilising the swathe of sophisticated MCMC methodologies that exist in the research to improve the efficiency of the algorithm. For instance, the use of MCMC algorithms that take into account gradient information such as Hamiltonian MCMC (Duane et al., 1987) can be used to improve the efficiency of sampling the parameters (Chatzilena et al., 2019). The second is by utilising and developing novel methodology to augment the states of a complex discrete time epidemic. The

algorithm struggled most with accepting changes at singular timesteps, when those changes cascaded through the remainder of the epidemic. The minimal changes possible were made, only allowing changes to be proposed in locations where it was locally possible for a different state to occur, and still the acceptance rates for many data augmentation steps were well less than 5%. This challenge will only increase as the scale of epidemic data gets larger, either with accepted changes having less and less effect on the likelihood, leading to poor mixing and an inefficient algorithm, or changes not being accepted because of cascading changes to the 1000s of farms and timesteps that are connected indirectly to this state. New methodologies are needed that can make large changes to the states that are conditioned on the cascading effect of the complex process and are as such valid. This may be a method that intelligently updates the states by conditioning the update on the process, or it may be a method that takes advantage of simulation methods to 're-simulate' a segment of the process, conditional on the remainder of the process, such that it will always be valid. We believe this area of research is rich for exploration, and the key element to unlocking the potential of these methods to make an impact for modern and future epidemic inference.

We believe that this thesis presents novel methodology for the accurate and efficient inference of big data epidemics, and future work is of value to pursue to realise the full potential of these novel methods.

Appendix A

Our Bovine Tuberculosis Model

A.1 MCMC Algorithms

Below we lay out the Adaptive Block MCMC algorithm and it subroutines as detailed in Section 5.9 onward. The details below are given in a general form, which when combined with the details of Chapter 5 can recover the full algorithm used to make inference.

Algorithm 15: Block Adaptive MCMC Algorithm

Input : $\mathsf{N_{its}}$ = Total desired number of iterations eg. 10^6,
Data = All state and event data,
$(\lambda_{inf}, m_{inf}, \Delta_{m_{inf}})$ = Initial values of the infection tuning parameters,
$(\lambda_{det}, m_{det}, \Delta_{m_{det}})$ = Initial values of the detection tuning parameters

Output : Results = Parameter values for each iteration

Elements: $\boldsymbol{\theta}$ = Epidemic parameters, $\mathsf{n_{tune}}$ = Tuning block counter

Set up

  `Initialise` $\boldsymbol{\theta}_{\mathsf{cur}}$

  `Set` it = 1, $\mathsf{n_{tune}} = 1$

Process

  while $it \leqslant \mathsf{N_{its}}$ **do**

    `Blk-Adpt-Metropolis-Hastings-Step()`'s **for** Parameters (Subroutine A.1)**:**

      $[\beta_c, \beta_b, \delta, F, \epsilon] \in \boldsymbol{\theta}^{inf}$

      $[\rho, \rho_E] \in \boldsymbol{\theta}^{det}$

    `Metropolis-Hastings-Step()`'s **for** Data Augmentation (Subroutine A.3)**:**

      Move S→E events in time (Subroutine A.4)

      Move E→I events in time (Subroutine A.5)

      Add/Remove S→E events (Subroutine A.6)

      Add/Remove E→I events (Subroutine A.7)

      Add/Remove Detection events (Subroutine A.8)

      Add/Remove Cattle Death events (Subroutine A.9)

      Add/Remove Parish Environmental Reservoir events (Subroutine A.10)

      Add/Remove Movement events (Subroutine A.11)

    Record the Results

    if $it = 25 \cdot (\mathsf{n_{tune}})$ **then**

      $\mathsf{n_{tune}} = \mathsf{n_{tune}} + 1$

    end

    it = it+1

  end

Subroutine A.1: Block Adaptive Metropolis-Hastings Step for Parameters

Input : $(\lambda_{inf}, \lambda_{det}, m_{inf}, m_{det}, \Delta_{m_{inf}}, \Delta_{m_{inf}})$ = Current tuning parameters, and $\mathsf{N_{its}}$, Data, $\mathsf{n_{tune}}$

Output : $\boldsymbol{\theta}_{\mathsf{cur}}$ = Updated epidemic parameters, $(\lambda_{inf}, \lambda_{det}, m_{inf}, m_{det}, \Delta_{m_{inf}}, \Delta_{m_{inf}})$ = Updated tuning parameters

Elements: $\pi(X|\boldsymbol{\theta}^*)$ = Likelihood of the epidemic given parameters θ^*, $\pi(\boldsymbol{\theta}^*|X)$ = Joint conditional posterior of parameters θ^* given Data X,
$q(\boldsymbol{\theta}'|\boldsymbol{\theta}^*)$ = Prob of proposing parameters θ' given the current parameters θ^*,
d_{inf}, d_{det} = Dimension of $\boldsymbol{\theta}_{\mathsf{cur}}$ for each block,
Σ_{inf}, Σ_{det} = Proposal posterior co-variance matrices

Repeat for each block of parameters:

Propose Update

- **if** *it* $\leqslant$ *min(5000,* $\mathsf{N_{its}}$*/10)* **then**
  - **if** *it* = *25*·$\mathsf{n_{tune}}$ **then**
    - Update λ using `Tune_λ()` (Subroutine A.2)
  - **end**
  - `Draw` $\log(\boldsymbol{\theta}_{\mathsf{prime}}) \sim \mathrm{N}(\log(\boldsymbol{\theta}_{\mathsf{cur}}), \frac{\lambda^2}{d}\mathrm{I}_d)$
- **else**
  - **with** *5% chance* **then**
    - `Set` $\Sigma = \frac{\lambda^2}{d}\mathrm{I}_d$ (1)
  - **else**
    - `Set` $\Sigma = \mathrm{m}^2 \times$[Current empirical Posterior Co-Variance Matrix] (2)
  - **end**
  - `Draw` $\log(\boldsymbol{\theta}_{\mathsf{prime}}) \sim \mathrm{N}(\log(\boldsymbol{\theta}_{\mathsf{cur}}), \Sigma)$
- **end**

Accept/Reject

- `Calculate` $\pi(\boldsymbol{\theta}_{\mathsf{cur}}|X)$, $\pi(\boldsymbol{\theta}_{\mathsf{prime}}|X)$, $q(\boldsymbol{\theta}_{\mathsf{cur}}|\boldsymbol{\theta}_{\mathsf{prime}})$, $q(\boldsymbol{\theta}_{\mathsf{prime}}|\boldsymbol{\theta}_{\mathsf{cur}})$ using `Posterior_fn()`
- `Calculate` the Metroplis-Hastings acceptance probability as $\alpha = \min(1, \frac{\prod_d[\boldsymbol{\theta}_{\mathsf{prime}}]\cdot\pi(\boldsymbol{\theta}_{\mathsf{prime}}|X)\cdot q(\boldsymbol{\theta}_{\mathsf{cur}}|\boldsymbol{\theta}_{\mathsf{prime}})}{\prod_d[\boldsymbol{\theta}_{\mathsf{cur}}]\cdot\pi(\boldsymbol{\theta}_{\mathsf{cur}}|X)\cdot q(\boldsymbol{\theta}_{\mathsf{prime}}|\boldsymbol{\theta}_{\mathsf{cur}})})$
- Accept or reject the proposal
- **if** Σ = *(2)* **then**
  - **if** *update accepted* **then**
    - `Set` $\mathrm{m} = \mathrm{m} + 2.3(\frac{\Delta_m}{\sqrt{it}})$
  - **else**
    - `Set` $\mathrm{m} = \mathrm{m}$ - $(\frac{\Delta_m}{\sqrt{it}})$
  - **end**
- **end**

Subroutine A.2: Function: Tune_λ()

Input : λ_{cur} = The current value of λ for the parameter block of interest,
n_{tune} = The number of tuning blocks so far,
Results = The acceptance (0/1) of the update steps so far
Output : λ_{updated} = The updated value of λ
Elements: it = iterations,
acc_prop = Acceptance proportion for the 25 iterations in the $\mathsf{n}_{\text{tune}}{}^{\text{th}}$ block,
δ = Change in the λ

Function `Tune`$_\lambda$`()`
  `Calculate` acc_prop
  if acc_prop < 0.33 **then**
    `Set` $\delta = -\min(0.05, \frac{1}{\sqrt{\mathsf{n}_{\text{tune}}}})$
  else
    `Set` $\delta = \min(0.05, \frac{1}{\sqrt{\mathsf{n}_{\text{tune}}}})$
  end
  $\log(\lambda_{\text{updated}}) = \log(\lambda_{\text{cur}}) + \delta$
  `Return`$(\lambda_{\text{updated}})$
end

Subroutine A.3: Metropolis-Hastings Step for Data Augmentation

Input : `Proposal_fn` = A function to generate the proposal,
$\boldsymbol{\theta}$ = Current values of the parameters,
and N_{its}, Data
Output : Data = Updated epidemic data
Elements: $\pi(X|\boldsymbol{\theta})$ = Likelihood of the epidemic given parameters $\boldsymbol{\theta}$,
$q(\boldsymbol{X}'|\boldsymbol{X})$ = prob. of proposing data X' given the current data X

Propose Update
  Propose an update to Data, $\boldsymbol{X}_{\text{cur}}$, using `Proposal_fn()`
  `Calculate` $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$ using `Proposal_fn()`
Accept/Reject
  `Calculate` $\pi(X_{\text{cur}}|\boldsymbol{\theta})$, $\pi(X_{\text{prime}}|\boldsymbol{\theta})$ using `Posterior_fn()`
  `Calculate` the Metropolis-Hastings acceptance probability as
  $\alpha = \min(1, \frac{\pi(X_{\text{prime}}|\boldsymbol{\theta}) \cdot q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})}{\pi(X_{\text{cur}}|\boldsymbol{\theta}) \cdot q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})})$
  Accept or reject the proposal

Subroutine A.4: Function: Propose to move an S to E event through time

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: t = A timestep in the data,
Δ = The magnitude and direction the event is moved through time
i = The unique ID of the updated farm

Function `Prop_Move_dE()`
  Generate $\Delta \in \{-1, 1\}$
  if $\Delta > 0$ **then**
    Choose a timestep, $t \in 1 : (T-1)$, and a farm i such that $dE_{i,t} > 0$
  else
    Choose a timestep, $t \in 2 : T$, and a farm i such that $dE_{i,t} > 0$
  end
  Update the Data to create Data′
  `Calculate` the proposal probabilities using
  $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{dE_{i,s}>0\}}$
  $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{dE'_{i,s}>0\}}$
  `Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)
end

Subroutine A.5: Function: Propose to move an E to I event through time

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: t = A timestep in the data,
Δ = The magnitude and direction the event is moved through time
i = The unique ID of the updated farm

Function `Prop_Move_dI()`
  Generate $\Delta \in [-1, 1]$
  if $\Delta > 0$ **then**
    Choose a timestep, $t \in 1 : (T-1)$, and a farm i such that $dI_{i,t} > 0$
  else
    Choose a timestep, $t \in 2 : T$, and a farm i such that $dI_{i,t} > 0$
  end
  Update the Data to create Data′
  `Calculate` the proposal probabilities using
  $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{dI_{i,s}>0\}}$
  $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{dI'_{i,s}>0\}}$
  `Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)
end

Subroutine A.6: Function: Propose to add or remove an S to E event

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: t = A timestep in the data,
$S_{i,t}$ = The number of susceptibles used to generate the exposure events,
$\mathsf{p}_{\mathsf{exp}}(i,t)$ = The probability of exposure at time t,
$dE_{i,t}$ = The number of exposure events at time t on farm i

Function `Prop_AddRem_dI()`

Generate $\Delta \in [-1, 1]$

if $\Delta > 0$ **then**

Choose a timestep, $t \in 1:T$, and a farm i such that $\{S_{i,t} > 0$ **and** $\mathsf{p}_{\mathsf{exp}}(i,t) > 0\}$;

else

Choose a timestep, $t \in 1:T$, and a farm i such that $dE_{i,t} > 0$

end

Update the Data to create Data′

`Calculate` the proposal probabilities using

if $\Delta > 0$ **then**

$q(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{X}_{\mathsf{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{S_{i,s}>0 \text{ and } p_{exp}(i,s)>0\}}$

$q(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{X}_{\mathsf{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{dE'_{i,s}>0\}}$

else

$q(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{X}_{\mathsf{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{dE_{i,s}>0\}}$

$q(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{X}_{\mathsf{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \left\{S'_{i,s}>0 \text{ and } p'_{exp}(i,s)>0\right\}}$

end

`Return`(Data′, $q(\boldsymbol{X}_{\mathsf{cur}}|\boldsymbol{X}_{\mathsf{prime}})$, $q(\boldsymbol{X}_{\mathsf{prime}}|\boldsymbol{X}_{\mathsf{cur}})$)

end

Subroutine A.7: Function: Propose to add or remove an E to I event

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: t = A timestep in the data,
$E_{i,t}$ = The number of exposed used to generate the E to I events,
$dI_{i,t}$ = The number of infection events at time t on farm i

Function `Prop_AddRem_dI()`
  Generate $\Delta \in \{-1, 1\}$
  if $\Delta > 0$ **then**
    Choose a timestep, $t \in 1:T$, and a farm i such that $E_{i,t} > 0$
  else
    Choose a timestep, $t \in 1:T$, and a farm i such that $dI_{i,t} > 0$
  end
  Update the Data to create Data′
  `Calculate` the proposal probabilities using
  if $\Delta > 0$ **then**
    $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{E_{i,s}>0\}}$
    $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{dI'_{i,s}>0\}}$
  else
    $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{dI_{i,t}>0\}}$
    $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_s \{E'_{i,s}>0\}}$
  end
  `Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)
end

Subroutine A.8: Function: Propose to add or remove a detection event

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: t = A timestep in the data,
$E_{i,t}$, $I_{i,t}$ = The number of cattle tested in each state,
$H^E_{i,t}$, $H^I_{i,t}$ = The number of detection events on farm i for each state

Function `Prop_AddRem_Det()`

Choose a timestep, $t \in 1:T$, and a farm i such that $\{E_{i,t} > 0$ **and** $I_{i,t} > 0$ **and** $H^E_{i,t} + H^I_{i,t} > 0\}$;

`Calculate` the probability of each permutation of detections, ζ

Randomly select a new permutation weighted by it's probability

Update the Data to create Data′

`Calculate` the proposal probabilities using

$q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{\zeta'_i}{\sum\zeta}$

$q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{\zeta_i}{\sum\zeta}$

`Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

Subroutine A.9: Function: Propose to add or remove a Death event

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: t = A timestep in the data,
$d_{i,t}$ = The known number of deaths,
$X_{i,t}$ = The states of animals used to generate the deaths,
$D_{i,t}$ = The number of death events on farm i for each state

Function `Prop_AddRem_Deaths()`

Choose a timestep, $t \in 1:T$, and a farm i such that {at least 2 states of $X_{i,t} > 0$ **and** $d_{i,t} > 0$};

`Generate` a new set of Death events using $D'_{i,t} \sim \text{MHG}(X_{i,t},\ d_{i,t})$

Update the Data to create Data′

`Calculate` the proposal probabilities using

$q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = pdf(D'_{i,t}; \text{MHG}(X_{i,t},\ d_{i,t}))$

$q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = pdf(D_{i,t}; \text{MHG}(X_{i,t},\ d_{i,t}))$

`Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

Subroutine A.10: Function: Propose to add or remove Environmental Pressure

Input : Data = The states and events of the epidemic at all timesteps
Output : $V'_{p,t}$ = The current environmental pressure
Elements: t = A timestep in the data,
$V_{p,t}$ = The current environmental pressure,
$N_{p,t}$ = The additional environmental pressure generated,
$R_{p,t}$ = The remaining environmental pressure

Function `Prop_AddRem_Env()`

Choose a timestep, $t \in 1:T$, and a parish p

`Generate` an amount of Environmental pressure remaining using
$R'_{p,t} \sim \text{Bin}\left(V_{p,t},\ 1-\epsilon\right)$

`Generate` an amount of additional Environmental pressure using
$N'_{p,t} = \text{Po}\left(\sum_{i \in p} X_{i,t}\right)$

Update the $V_{i,t}$

`Calculate` the proposal probabilities using

$q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = pdf(R'_{p,t}; \text{Bin}(V_{p,t},\ 1-\epsilon)) \times pdf(N'_{p,t}; \text{Po}\left(\sum_{i \in p} X_{i,t}\right))$

$q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = pdf(R_{p,t}; \text{Bin}(V'_{p,t},\ 1-\epsilon)) \times pdf(N_{p,t}; \text{Po}\left(\sum_{i \in p} X_{i,t}\right))$

`Return`($V'_{p,t}$, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

Subroutine A.11: Function: Propose to add or remove a Movement event

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: t = A timestep in the data,
$m_{i,t}$ = The known number of movments off the farm,
$X_{i,t}$ = The states of cattle used to generate the deaths

Function `Prop_AddRem_Deaths()`

Choose a timestep, $t \in 1:T$, and a farm i such that {at least 2 states of $X_{i,t} > 0$ **and** $m_{i,t} > 0$};

`Generate` a new set of Movement events

Update the Data to create Data′

`Calculate` the proposal probabilities using

$q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = pdf(M'_{i,s})$

$q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = pdf(M_{i,s})$

`Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

Appendix B

Real Data Model

B.1 MCMC Algorithms

The majority of the methods used align with those presented in Chapter 5, making the necessary adjustments for the removal of badger data. Here we present the additional subroutines for augmenting the initial conditions of the epidemic in a generic form, which can be combined with the details of Chapter 6 to recover the algorithm used to make inference.

Algorithm 16: Block Adaptive MCMC Algorithm for Real Data

Input : $\mathsf{N_{its}}$ = Total desired number of iterations eg. 10^6,
Data = All state and event data,
$(\lambda_{inf}, m_{inf}, \Delta_{m_{inf}})$ = Initial values of the infection tuning parameters,
$(\lambda_{det}, m_{det}, \Delta_{m_{det}})$ = Initial values of the detection tuning parameters
Output : $\mathsf{Results}$ = Parameter values for each iteration
Elements: $\boldsymbol{\theta}$ = Epidemic parameters, $\mathsf{n_{tune}}$ = Tuning block counter

Set up
  `Initialise` $\boldsymbol{\theta}_{\mathsf{cur}}$
  `Set` it = 1, $\mathsf{n_{tune}} = 1$
Process
  while *it* $\leqslant \mathsf{N_{its}}$ **do**
    `Blk-Adpt-Metropolis-Hastings-Step()`'s **for** Parameters (Subroutine A.1)**:**
      $[\beta_c, \delta, F, \epsilon] \in \boldsymbol{\theta}^{inf}$
      $[\rho, \rho_E] \in \boldsymbol{\theta}^{det}$
    `Metropolis-Hastings-Step()`'s **for** Data Augmentation (Subroutine A.3)**:**
      Move S→E events in time (Subroutine A.4)
      Move E→I events in time (Subroutine A.5)
      Add/Remove S→E events (Subroutine A.6)
      Add/Remove E→I events (Subroutine A.7)
      Add/Remove Detection events (Subroutine A.8)
      Add/Remove Cattle Death events (Subroutine A.9)
      Add/Remove Parish Environmental Reservoir events (Subroutine A.10)
      Add/Remove Movement events (Subroutine A.11)
      Change Initial S to E (Subroutine B.1
      Change Initial S to I (Subroutine B.1
      Change Initial E to S (Subroutine B.1
      Change Initial E to I (Subroutine B.1
      Change Initial I to S (Subroutine B.1
      Change Initial I to E (Subroutine B.1
      Change Initial Parish Environmental Reservoir (Subroutine B.2
    Record the $\mathsf{Results}$
    if $it = 25 \cdot (\mathsf{n_{tune}})$ **then**
      $\mathsf{n_{tune}} = \mathsf{n_{tune}} + 1$
    end
    it = it+1
  end

Subroutine B.1: Function: Propose to change the initial state of an animal

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: Δ = The magnitude and direction the event is moved through time
i = The unique ID of the updated farm

Function `Change_init_states()`

Choose a state $\kappa \in \{S, E, I\}$
Choose a farm i such that $X^{\kappa}_{i,1} > 0$
Update the Data to create Data′ by changing an
`Calculate` the proposal probabilities using
$q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}}) = \frac{1}{2} \cdot \frac{1}{\sum_i \{X^{\kappa}_{i,1}>0\}}$
$q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}}) = \frac{1}{2} \cdot \frac{1}{\sum_i \{X'^{\kappa}_{i,1}>0\}}$
`Return`(Data′, $q(\boldsymbol{X}_{\text{cur}}|\boldsymbol{X}_{\text{prime}})$, $q(\boldsymbol{X}_{\text{prime}}|\boldsymbol{X}_{\text{cur}})$)

end

Subroutine B.2: Function: Propose a change to the initial parish environmental reservoir

Input : Data = The states and events of the epidemic at all timesteps
Output : Data′ = The states and events of the epidemic at all timesteps after the update
Elements: t = A timestep in the data,
Δ = The magnitude and direction the event is moved through time
i = The unique ID of the updated farm

Function `Prop_Change_penv()()`

Choose a parish p and a farm i,
Generate random variable $\Delta \sim U(-0.001, 0.001)$
Update the Data to create Data′
`Calculate` the proposal probabilities using
$q(\boldsymbol{V}_{\text{prime}}|\boldsymbol{V}_{\text{cur}}) = \frac{1}{0.002}$
$q(\boldsymbol{V}_{\text{cur}}|\boldsymbol{V}_{\text{prime}}) = \frac{1}{0.002}$
`Return`(Data′, $q(\boldsymbol{V}_{\text{cur}}|\boldsymbol{V}_{\text{prime}})$, $q(\boldsymbol{V}_{\text{prime}}|\boldsymbol{V}_{\text{cur}})$)

end

References

Ajelli, Marco et al. (June 2010). "Comparing large-scale computational approaches to epidemic modeling: Agent-based versus structured metapopulation models". In: *BMC Infectious Diseases* 10.1, p. 190.

APHA (2021). *Bovine TB: how to spot and report the disease*. URL: https://www.gov.uk/guidance/bovine-tb.

— (2022). *Bovine TB: get your cattle tested in England*. URL: https://www.gov.uk/guidance/bovine-tb-getting-your-cattle-tested-in-england#tb-breakdown.

— (2023a). *Bovine TB testing intervals 2021*. URL: https://www.gov.uk/guidance/bovine-tb-testing-intervals.

— (2023b). *Dealing with TB in your herd: what to do if bovine TB is detected in your herd in Wales*. URL: https://www.gov.uk/government/publications/what-happens-if-tb-is-identified-in-your-herd/dealing-with-tb-in-your-herd-what-to-do-if-bovine-tb-is-detected-in-your-herd-in-wales.

— (2023c). *Quarterly TB in cattle in Great Britain statistics notice: December 2022*. URL: https://www.gov.uk/government/statistics/incidence-of-tuberculosis-tb-in-cattle-in-great-britain/quarterly-tb-in-cattle-in-great-britain-statistics-notice-december-2022.

Bailey, Norman TJ (1975). *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE.

Balabdaoui, Fadoua and Dirk Mohr (Dec. 2020). "Age-stratified discrete compartment model of the COVID-19 epidemic with application to Switzerland". In: *Scientific Reports* 10. DOI: 10.1038/s41598-020-77420-4.

Bartlett, M. S. (1949). "Some Evolutionary Stochastic Processes". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 11.2, pp. 211–229. ISSN: 00359246. URL: http://www.jstor.org/stable/2984077 (visited on 08/22/2023).

Bhuachalla, Deirdre Ní et al. (2014). "The role of badgers in the epidemiology of Mycobacterium bovis infection (tuberculosis) in cattle in the United Kingdom and the Republic of Ireland: current perspectives on control strategies". In: *Dove Press* 2015.6, pp. 27–38. DOI: https://doi.org/10.2147%2FVMRR.S53643.

Biek, R et al. (2012). "Whole genome sequencing reveals local transmission patterns of Mycobacterium bovis in sympatric cattle and badger populations." In: *PLoS Pathog.* 8. DOI: doi:10.1371/journal.ppat.1003008.

Brooks-Pollock, E, G Roberts, and M Keeling (2014). "A dynamic model of bovine tuberculosis spread and control in Great Britain". In: *Nature* 511, pp. 228–231. DOI: https://doi.org/10.1038/nature13529.

Brooks-Pollock, Ellen, Andrew Conlan, et al. (Oct. 2013). "Age-dependent patterns of bovine tuberculosis in cattle". In: *Veterinary research* 44, p. 97. DOI: 10.1186/1297-9716-44-97.

Brooks-Pollock, Ellen and Matt Keeling (2009). "Herd size and bovine tuberculosis persistence in cattle farms in Great Britain". In: *Preventive Veterinary Medicine* 92.4. Special section: Schwabe Symposium 2008, pp. 360–365. ISSN: 0167-5877. DOI: https://doi.org/10.1016/j.prevetmed.2009.08.022. URL: https://www.sciencedirect.com/science/article/pii/S0167587709002505.

Brooks-Pollock, Ellen and James L. N. Wood (2015). "Eliminating bovine tuberculosis in cattle and badgers: insight from a dynamic model". In: *Royal Society B Bio Sci.* DOI: https://doi.org/10.1098/rspb.2015.0374.

Brunsdon, Chris (2020). "Modelling epidemics: Technical and critical issues in the context of COVID-19". In: *Dialogues in Human Geography* 10.2, pp. 250–254. DOI: 10.1177/2043820620934328. eprint: https://doi.org/10.1177/2043820620934328. URL: https://doi.org/10.1177/2043820620934328.

Cauchemez, S. et al. (2004). "A Bayesian MCMC approach to study transmission of influenza: application to household longitudinal data". In: *Statistics in Medicine* 23.22, pp. 3469–3487. DOI: https://doi.org/10.1002/sim.1912. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.1912. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.1912.

Chantrey, Julian et al. (2018). "A study of tuberculosis in road traffic-killed badgers on the edge of the British bovine TB epidemic area". In: *Scientific Reports* 8.1, pp. 2045–2322. DOI: https://doi.org/10.1038/s41598-018-35652-5.

Chatzilena, Anastasia et al. (2019). "Contemporary statistical inference for infectious disease models using Stan". In: *Epidemics* 29, p. 100367. ISSN: 1755-4365. DOI: https://doi.org/10.1016/j.epidem.2019.100367. URL: https://www.sciencedirect.com/science/article/pii/S1755436519300325.

Conlan, A et al. (2012). "Estimating the Hidden Burden of Bovine Tuberculosis in Great Britain." In: *PLoS Comput Biol* 8.10. DOI: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002730. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002730.

Cox, D et al. (Dec. 2005). "Simple model for tuberculosis in cattle and badgers". In: *Proceedings of the National Academy of Sciences* 102.49, pp. 17588–17593. ISSN: 0027-8424. DOI: 10.1073/pnas.0509003102. URL: http://dx.doi.org/10.1073/pnas.0509003102.

Cox, D.R. and V. Isham (1980). *Point Processes*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis. ISBN: 9780412219108. URL: https://books.google.co.uk/books?id=KWF2xY6s3PoC.

Crispell, Joseph et al. (Dec. 2019). "Combining genomics and epidemiology to analyse bi-directional transmission of *Mycobacterium bovis* in a multi-host system". In: *eLife* 8. Ed. by Marc Lipsitch, Neil M Ferguson, and Christian Gortazar, e45833. ISSN: 2050-084X. DOI: 10.7554/eLife.45833. URL: https://doi.org/10.7554/eLife.45833.

Csilléry, Katalin et al. (July 2010). "Approximate Bayesian Computation (ABC) in practice". In: *Trends in Ecology & Evolution* 25.7, pp. 410–418.

Dawson, Peter M., Marleen Werkman, and Ellen Brooks-Pollock (2015). "Epidemic predictions in an imperfect world: modelling disease spread with partial data". In: *Royal Society B Bio Sci*. DOI: https://doi.org/10.1098/rspb.2015.0205.

de la Rua-Domenech, R. et al. (2006). "Ante mortem diagnosis of tuberculosis in cattle: A review of the tuberculin tests, -interferon assay and other ancillary diagnostic techniques". In: *Research in Veterinary Science* 81.2, pp. 190–210. ISSN: 0034-5288. DOI: https://doi.org/10.1016/j.rvsc.2005.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0034528806000026.

Deardon, R. et al. (2010). "Inference for individual-level models of infectious diseases in large populations". English. In: *Statistica Sinica* 20.1. WOS:000275034100015 Times Cited: 2, pp. 239–261.

DEFRA (2004). *The risk to cattle from wildlife species other than badgers in areas of high herd breakdown risk*. URL: http://www2.defra.gov.uk/research/project_data/More.asp?I=SE3010&M=KWS&V=se3010&SUBMIT1=Search&SCOPE=0.

Delahay, R. J. et al. (2013). "Long-term temporal trends and estimated transmission rates for Mycobacterium bovis infection in an undisturbed high-density badger (Meles meles) population". In: *Epidemiology amp; Infection* 141.7, pp. 1445–1456. DOI: 10.1017/S0950268813000721.

Dellaportas, Petros and Gareth Roberts (2003). "An Introduction to MCMC". In: *Spatial Statistics and Computational Methods*. Ed. by Jesper Møller. New York,

NY: Springer New York, pp. 1–41. ISBN: 978-0-387-21811-3. DOI: 10.1007/978-0-387-21811-3_1. URL: https://doi.org/10.1007/978-0-387-21811-3_1.

Donnelly, C and P Nouvellet (2013). "The contribution of badgers to confirmed tuberculosis in cattle in high-incidence areas in England." In: *PLoS Curr*. DOI: 10.1371/currents.outbreaks.097a904d3f3619db2fe78d24bc776098.

Duane, Simon et al. (1987). "Hybrid Monte Carlo". In: *Physics Letters B* 195.2, pp. 216–222. ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(87)91197-X. URL: https://www.sciencedirect.com/science/article/pii/037026938791197X.

Epstein, Joshua M. (2008). "Why Model?" In: *Journal of Artificial Societies and Social Simulation* 11.4, p. 12. ISSN: 1460-7425. URL: https://www.jasss.org/11/4/12.html.

Frazier, D T et al. (June 2018). "Asymptotic properties of approximate Bayesian computation". In: *Biometrika* 105.3, pp. 593–607. ISSN: 0006-3444. DOI: 10.1093/biomet/asy027. eprint: https://academic.oup.com/biomet/article-pdf/105/3/593/25470059/asy027.pdf. URL: https://doi.org/10.1093/biomet/asy027.

Funk, Sebastian and Aaron A. King (2020). "Choices and trade-offs in inference with infectious disease models". In: *Epidemics* 30, p. 100383. ISSN: 1755-4365. DOI: https://doi.org/10.1016/j.epidem.2019.100383. URL: https://www.sciencedirect.com/science/article/pii/S1755436519300441.

Gelman, A, J B Carlin, et al. (2013). *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis. ISBN: 9781439840955. URL: https://books.google.co.uk/books?id=ZXL6AQAAQBAJ.

Gelman, A, G Roberts, and W R Gilks (1996). "Efficient Metropolis jumping rules". In: *Bayesian Statistics*. Oxford University Press, Oxford, pp. 599–608.

Gillespie, Daniel T. (1977). "Exact stochastic simulation of coupled chemical reactions". In: *The Journal of Physical Chemistry* 81.25, pp. 2340–2361. DOI:

10.1021/j100540a008. eprint: https://doi.org/10.1021/j100540a008. URL: https://doi.org/10.1021/j100540a008.

Goodchild, A. V. et al. (2012). "Geographical association between the genotype of bovine tuberculosis in found dead badgers and in cattle herds". In: *Veterinary Record* 170.10, pp. 259–259. DOI: https://doi.org/10.1136/vr.100193. URL: https://bvajournals.onlinelibrary.wiley.com/doi/abs/10.1136/vr.100193.

Gopal, R. et al. (2006). "Introduction of bovine tuberculosis to north-east England by bought-in cattle". In: *Veterinary Record* 159.9, pp. 265–271. DOI: https://doi.org/10.1136/vr.159.9.265. eprint: https://bvajournals.onlinelibrary.wiley.com/doi/pdf/10.1136/vr.159.9.265.

Gordon, S.B. and P. Barrow (2018). *Bovine Tuberculosis*. CABI. ISBN: 9781786391537. URL: https://books.google.co.uk/books?id=EZmdswEACAAJ.

Green, Darren M et al. (2008). "Estimates for local and movement-based transmission of bovine tuberculosis in British cattle". In: *Royal Society B Bio Sci*. DOI: https://doi.org/10.1098/rspb.2007.1601.

Green, L. and S. Cornell (2005). "Investigations of cattle herd breakdowns with bovine tuberculosis in four counties of England and Wales using VETNET data". In: *Preventive Veterinary Medicine* 70.3, pp. 293–311. ISSN: 0167-5877. DOI: https://doi.org/10.1016/j.prevetmed.2005.05.005. URL: https://www.sciencedirect.com/science/article/pii/S0167587705001480.

Haario, Heikki, Eero Saksman, and Johanna Tamminen (2001). "An Adaptive Metropolis Algorithm". In: *Bernoulli* 7.2, pp. 223–242. ISSN: 13507265. URL: http://www.jstor.org/stable/3318737 (visited on 07/05/2023).

Ham, Cally et al. (2019). "Effect of culling on individual badger Meles meles behaviour: Potential implications for bovine tuberculosis transmission". In: *Journal of Applied Ecology* 56.11, pp. 2390–2399. DOI: https://doi.org/10.1111/1365-2664.13512. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/1365-2664.13512. URL: https://

besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/1365-2664.13512.

Huang, Xiaodong et al. (2016). "Bayesian estimation of the dynamics of pandemic (H1N1) 2009 influenza transmission in Queensland: A space–time SIR-based model". In: *Environmental Research* 146, pp. 308–314. ISSN: 0013-9351. DOI: https://doi.org/10.1016/j.envres.2016.01.013. URL: https://www.sciencedirect.com/science/article/pii/S0013935116300135.

Isham, V. and G. Medley (1996). *Models for Infectious Human Diseases: Their Structure and Relation to Data*. Isaac Newton Institute for Mathematical Sciences Cambridge: Publications of the Newton Institute. Cambridge University Press. ISBN: 9780521453394. URL: https://books.google.co.uk/books?id=U64ariBqV-oC.

Jenkins, H, R Woodroffe, and C Donnelly (Feb. 2010). "The Duration of the Effects of Repeated Widespread Badger Culling on Cattle Tuberculosis Following the Cessation of Culling". In: *PLOS ONE* 5.2, pp. 1–7. DOI: 10.1371/journal.pone.0009090. URL: https://doi.org/10.1371/journal.pone.0009090.

Jewell, C. et al. (2009a). "A novel approach to real-time risk prediction for emerging infectious diseases: A case study in Avian Influenza H5N1". In: *Preventive Veterinary Medicine* 91.1. Special Issue: GisVet 2007, pp. 19–28. ISSN: 0167-5877. DOI: https://doi.org/10.1016/j.prevetmed.2009.05.019. URL: https://www.sciencedirect.com/science/article/pii/S0167587709001408.

Jewell, Chris et al. (2009b). "Bayesian analysis for emerging infectious diseases". In: *Bayesian Analysis* 4.3, pp. 465–496. DOI: 10.1214/09-BA417. URL: https://doi.org/10.1214/09-BA417.

Jewell, Chris et al. (2023). *Bayesian inference for high-dimensional discrete-time epidemic models: spatial dynamics of the UK COVID-19 outbreak*. arXiv: 2306.07987 [physics.soc-ph].

Kamrujjaman, Md. et al. (2022). "Dynamics of SEIR model: A case study of COVID-19 in Italy". In: *Results in Control and Optimization* 7, p. 100119. ISSN: 2666-

7207. DOI: https://doi.org/10.1016/j.rico.2022.100119. URL: https://www.sciencedirect.com/science/article/pii/S2666720722000145.

Kao, R, M Price-Carter, and S Robbe-Austerman (2016). "Use of genomics to track bovine tuberculosis transmission". In: *Rev Sci Tech*. DOI: 10.20506/rst.35.1.2430.

Kao, Rowland R (June 2002). "The role of mathematical modelling in the control of the 2001 FMD epidemic in the UK". In: *Trends in Microbiology* 10.6, pp. 279–286.

Keeling, Matt et al. (2001). "Dynamics of the 2001 UK Foot and Mouth Epidemic: Stochastic Dispersal in a Heterogeneous Landscape". In: *Science* 294.5543, pp. 813–817. DOI: 10.1126/science.1065973. eprint: https://www.science.org/doi/pdf/10.1126/science.1065973. URL: https://www.science.org/doi/abs/10.1126/science.1065973.

Kingman, J.F.C. (1992). *Poisson Processes*. Oxford Studies in Probability. Clarendon Press. ISBN: 9780191591242. URL: https://books.google.co.uk/books?id=VEiM-OtwDHkC.

Kirchhelle, Claas (Nov. 2020). "Angela Cassidy, : British Debates over Bovine Tuberculosis and Badgers". In: *Social History of Medicine* 35.3, pp. 1036–1038. ISSN: 0951-631X. DOI: 10.1093/shm/hkaa075. eprint: https://academic.oup.com/shm/article-pdf/35/3/1036/45627468/hkaa075.pdf.

Krebs, J et al. (1997). *Bovine Tuberculosis in Cattle and Badgers*. English. MAFF.

Lekone, Phenyo E and Bärbel F Finkenstädt (2006). "Statistical Inference in a Stochastic Epidemic SEIR Model with Control Intervention: Ebola as a Case Study". In: *Biometrics* 62.4, pp. 1170–1177. DOI: https://doi.org/10.1111/j.1541-0420.2006.00609.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1541-0420.2006.00609.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1541-0420.2006.00609.x.

Li, Wentao and Paul Fearnhead (Jan. 2018). "On the asymptotic efficiency of approximate Bayesian computation estimators". In: *Biometrika* 105.2, pp. 285–299.

ISSN: 0006-3444. DOI: 10.1093/biomet/asx078. eprint: https://academic.oup.com/biomet/article-pdf/105/2/285/24820949/asx078.pdf. URL: https://doi.org/10.1093/biomet/asx078.

Lloyd, Alun L. and Robert M. May (1996). "Spatial Heterogeneity in Epidemic Models". In: *Journal of Theoretical Biology* 179.1, pp. 1–11. ISSN: 0022-5193. DOI: https://doi.org/10.1006/jtbi.1996.0042. URL: https://www.sciencedirect.com/science/article/pii/S0022519396900429.

Maddock, EC (1933). "Studies on the Survival Time of the Bovine Tubercle Bacillus in Soil, Soil and Dung, in Dung and on Grass, with Experiments on the Preliminary Treatment of Infected Organic Matter and the Cultivation of the Organism." In: *The Journal of hygiene*, pp. 103–17. DOI: 10.1017/s002217240001843x.

Mathews, Fiona et al. (2005). "Bovine tuberculosis (Mycobacterium bovis) in British farmland wildlife: the importance to agriculture". In: *Royal Society* 273.1584. ISSN: 1472-2954. DOI: https://doi.org/10.1098/rspb.2005.3298. URL: https://royalsocietypublishing.org/doi/full/10.1098/rspb.2015.0374#RSPB20150374C9R.

Mbuvha, Rendani and Tshilidzi Marwala (Aug. 2020). "Bayesian inference of COVID-19 spreading rates in South Africa". In: *PLOS ONE* 15.8, pp. 1–16. DOI: 10.1371/journal.pone.0237126. URL: https://doi.org/10.1371/journal.pone.0237126.

McBryde, Emma S. et al. (2020). "Role of modelling in COVID-19 policy development". In: *Paediatric Respiratory Reviews* 35, pp. 57–60. ISSN: 1526-0542. DOI: https://doi.org/10.1016/j.prrv.2020.06.013. URL: https://www.sciencedirect.com/science/article/pii/S1526054220300981.

Monaghan, M.L. et al. (1994). "The tuberculin test". In: *Veterinary Microbiology* 40.1, pp. 111–124. ISSN: 0378-1135. DOI: https://doi.org/10.1016/0378-1135(94)90050-7. URL: https://www.sciencedirect.com/science/article/pii/0378113594900507.

Moustakas, A. and M.R. Evans (2017). "A big-data spatial, temporal and network analysis of bovine tuberculosis between wildlife (badgers) and cattle". In: *Stochastic Environmental Research and Risk Assessment* 31, pp. 315–328. DOI: https://doi.org/10.1007/s00477-016-1311-x.

Natural England (2022). *Summary of 2021 badger control operations*. URL: https://www.gov.uk/government/publications/bovine-tb-summary-of-badger-control-monitoring-during-2021/summary-of-2021-badger-control-operations.

Neal, Peter and Gareth Roberts (Apr. 2004). "Statistical inference and model selection for the 1861 Hagelloch measles epidemic". In: *Biostatistics* 5.2, pp. 249–261. ISSN: 1465-4644. DOI: 10.1093/biostatistics/5.2.249. eprint: https://academic.oup.com/biostatistics/article-pdf/5/2/249/651176/050249.pdf. URL: https://doi.org/10.1093/biostatistics/5.2.249.

O'Neill, P D and G Roberts (1999). "Bayesian inference for partially observed stochastic epidemics". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 162.1, pp. 121–129. DOI: https://doi.org/10.1111/1467-985X.00125. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/1467-985X.00125. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-985X.00125.

O'Neill, Philip D. (2002). "A tutorial introduction to Bayesian inference for stochastic epidemic models using Markov chain Monte Carlo methods". In: *Mathematical Biosciences* 180.1, pp. 103–114. ISSN: 0025-5564. DOI: https://doi.org/10.1016/S0025-5564(02)00109-8. URL: https://www.sciencedirect.com/science/article/pii/S0025556402001098.

Overton, Christopher E. et al. (Sept. 2022). "EpiBeds: Data informed modelling of the COVID-19 hospital burden in England". In: *PLOS Computational Biology* 18.9, pp. 1–20. DOI: 10.1371/journal.pcbi.1010406. URL: https://doi.org/10.1371/journal.pcbi.1010406.

Pollock, J.M. et al. (2001). "Immune responses in bovine tuberculosis". In: *Tuberculosis* 81.1, pp. 103–107. ISSN: 1472-9792. DOI: https://doi.org/10.1054/tube.2000.0258. URL: https://www.sciencedirect.com/science/article/pii/S1472979200902580.

Reynolds, Debby (2006). "A review of tuberculosis science and policy in Great Britain". In: *Veterinary Microbiology* 112.2. 4th International Conference on Mycobacterium bovis, pp. 119–126. ISSN: 0378-1135. DOI: https://doi.org/10.1016/j.vetmic.2005.11.042.

Roberts, G, A Gelman, and W R Gilks (1997). "Weak convergence and optimal scaling of random walk Metropolis algorithms". In: *The Annals of Applied Probability* 7.1, pp. 110–120. DOI: 10.1214/aoap/1034625254. URL: https://doi.org/10.1214/aoap/1034625254.

Roberts, Gareth and Jeffrey Rosenthal (2009). "Examples of Adaptive MCMC". In: *Journal of Computational and Graphical Statistics* 18.2, pp. 349–367. DOI: 10.1198/jcgs.2009.06134. eprint: https://doi.org/10.1198/jcgs.2009.06134. URL: https://doi.org/10.1198/jcgs.2009.06134.

Roberts, Gareth and Richard Tweedie (1996). "Exponential Convergence of Langevin Distributions and Their Discrete Approximations". In: *Bernoulli* 2.4, pp. 341–363. ISSN: 13507265. URL: http://www.jstor.org/stable/3318418 (visited on 11/02/2023).

Rossi, Gianluigi et al. (2022). "Phylodynamic analysis of an emergent Mycobacterium bovis outbreak in an area with no previously known wildlife infections". In: *Journal of Applied Ecology* 59.1, pp. 210–222. DOI: https://doi.org/10.1111/1365-2664.14046. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/1365-2664.14046. URL: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/1365-2664.14046.

Sherlock, Chris, Paul Fearnhead, and Gareth Roberts (2010). "The Random Walk Metropolis: Linking Theory and Practice Through a Case Study". In: *Statistical*

Science 25.2, pp. 172–190. DOI: 10.1214/10-STS327. URL: https://doi.org/10.1214/10-STS327.

Shinde, Gitanjali R et al. (June 2020). "Forecasting Models for Coronavirus Disease (COVID-19): A Survey of the State-of-the-Art". In: *SN Computer Science* 1.4, p. 197.

Smith, G. (2001). "Models of Mycobacterium bovis in wildlife and cattle". In: *Tuberculosis* 81.1, pp. 51–64. ISSN: 1472-9792. DOI: https://doi.org/10.1054/tube.2000.0264. URL: https://www.sciencedirect.com/science/article/pii/S1472979200902646.

Smith, Graham et al. (Nov. 2016). "Model of Selective and Non-Selective Management of Badgers (Meles meles) to Control Bovine Tuberculosis in Badgers and Cattle". In: *PLOS ONE* 11.11, pp. 1–16. DOI: 10.1371/journal.pone.0167206. URL: https://doi.org/10.1371/journal.pone.0167206.

Streftaris, George and Gavin J Gibson (2004). "Bayesian inference for stochastic epidemics in closed populations". In: *Statistical Modelling* 4.1, pp. 63–75. DOI: 10.1191/1471082X04st065oa. eprint: https://doi.org/10.1191/1471082X04st065oa. URL: https://doi.org/10.1191/1471082X04st065oa.

Swift, Benjamin et al. (2021). "Tuberculosis in badgers where the bovine tuberculosis epidemic is expanding in cattle in England". In: *Scientific Reports* 11.1. DOI: https://doi.org/10.1038/s41598-021-00473-6. URL: https://www.nature.com/articles/s41598-021-00473-6#citeas.

Taghizadeh, Leila, Ahmad Karimi, and Clemens Heitzinger (2020). "Uncertainty quantification in epidemiological models for the COVID-19 pandemic". In: *Computers in Biology and Medicine* 125, p. 104011. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2020.104011. URL: https://www.sciencedirect.com/science/article/pii/S0010482520303425.

The Independent Scientific Group on Cattle TB (2008). *Bovine TB: The Scientific Evidence*. URL: https://publications.parliament.uk/pa/cm200708/cmselect/cmenvfru/130/130i.pdf.

The Ministry of Agriculture, Fisheries, and Food (1976). *Bovine Tuberculosis in Badgers*. English. MAFF.

UK Health Security Agency (2023). *Coronavirus (COVID-19) in the UK*. https://coronavirus.data.gov.uk/details/cases?areaType=nation&areaName=England [Accessed: (13-07-23)].

Waddington, Keir (2004). "To Stamp Out "So Terrible a Malady": Bovine Tuberculosis and Tuberculin Testing in Britain, 1890–1939". In: *Medical History* 48.1, pp. 29–48. DOI: https://doi.org/10.1017/S0025727300007043.

Welsh Government (2018). *Testing cattle for bovine TB*. URL: https://www.gov.wales/testing-cattle-bovine-tb.

Wilkinson, D et al. (2004). "A model of bovine tuberculosis in the badger Meles meles: an evaluation of different vaccination strategies". In: *Journal of Applied Ecology* 41.3, pp. 492–501. DOI: https://doi.org/10.1111/j.0021-8901.2004.00898.x. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/j.0021-8901.2004.00898.x. URL: https://besjournals.onlinelibrary.wiley.com/doi/abs/10.1111/j.0021-8901.2004.00898.x.

Wilkinson, Richard David (2013). In: *Statistical Applications in Genetics and Molecular Biology* 12.2, pp. 129–141. DOI: doi:10.1515/sagmb-2013-0010. URL: https://doi.org/10.1515/sagmb-2013-0010.

Woodroffe, R et al. (2006). "Culling and cattle controls influence tuberculosis risk for badgers." In: *Proc. Natl Acad. Sci. USA* 103.14, pp. 713–714. DOI: doi:10.1073/pnas.0606251103.

Woodroffe, Rosie et al. (2016). "Badgers prefer cattle pasture but avoid cattle: implications for bovine tuberculosis control". In: *Ecology Letters* 19.10, pp. 1201–1208. DOI: https://doi.org/10.1111/ele.12654. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/ele.12654.

Woolhouse, M.E.J. (May 2003). "Foot-and-mouth disease in the UK: What should we do next time?" In: *Journal of Applied Microbiology* 94.s1, pp. 126–130. ISSN: 1364-5072. DOI: 10.1046/j.1365-2672.94.s1.15.x. eprint: https://

academic.oup.com/jambio/article-pdf/94/s1/126/47299260/jambio0126.pdf. URL: https://doi.org/10.1046/j.1365-2672.94.s1.15.x.

World Health Organisation (2023). *Influenza (Seasonal)*. URL: https://www.who.int/news-room/fact-sheets/detail/influenza-(seasonal).

Xiang, Yue et al. (2021). "COVID-19 epidemic prediction and the impact of public health interventions: A review of COVID-19 epidemic models". In: *Infectious Disease Modelling* 6, pp. 324–342. ISSN: 2468-0427. DOI: https://doi.org/10.1016/j.idm.2021.01.001. URL: https://www.sciencedirect.com/science/article/pii/S2468042721000038.

Zhou, Yicang, Zhien Ma, and F. Brauer (2004). "A discrete epidemic model for SARS transmission and control in China". In: *Mathematical and Computer Modelling* 40.13, pp. 1491–1506. ISSN: 0895-7177. DOI: https://doi.org/10.1016/j.mcm.2005.01.007. URL: https://www.sciencedirect.com/science/article/pii/S0895717705000099.