International Statistical Review

International Statistical Review (2020), 88, 2, 441–461 doi:10.1111/insr.12399

Statistical Implementations of Agent-Based Demographic Models

Mevin Hooten[1], Christopher Wikle[2] and Michael Schwob[3]

[1] *U.S. Geological Survey, Colorado Cooperative Fish and Wildlife Research Unit, Department of Fish, Wildlife, and Conservation Biology, Department of Statistics, Colorado State University, Fort Collins, 80523-1484, CO, USA*
E-mail: hooten@rams.colostate.edu

[2] *Department of Statistics, University of Missouri, Columbia, 65211-6100, MO, USA*

[3] *Department of Mathematical Sciences, University of Nevada, Las Vegas, Las Vegas, 89154-9900, NV, USA*

Summary

A variety of demographic statistical models exist for studying population dynamics when individuals can be tracked over time. In cases where data are missing due to imperfect detection of individuals, the associated measurement error can be accommodated under certain study designs (e.g. those that involve multiple surveys or replication). However, the interaction of the measurement error and the underlying dynamic process can complicate the implementation of statistical agent-based models (ABMs) for population demography. In a Bayesian setting, traditional computational algorithms for fitting hierarchical demographic models can be prohibitively cumbersome to construct. Thus, we discuss a variety of approaches for fitting statistical ABMs to data and demonstrate how to use multi-stage recursive Bayesian computing and statistical emulators to fit models in such a way that alleviates the need to have analytical knowledge of the ABM likelihood. Using two examples, a demographic model for survival and a compartment model for COVID-19, we illustrate statistical procedures for implementing ABMs. The approaches we describe are intuitive and accessible for practitioners and can be parallelised easily for additional computational efficiency.

Key words: Bayesian; emulator; individual-based model; mechanistic model; MCMC.

1 Introduction

The dynamics of individuals, whether they are organisms, air particles, molecules, businesses, watersheds or political units, are of interest in a variety of fields. Models that describe individual-based dynamics are often referred to as ‘agent-based models’ (ABMs Railsback & Grimm, 2019). Many previous and ongoing studies have relied on ABMs as simulation models (Diggle & Gratton, 1984; Hartig *et al.*, 2011) because it is often more intuitive to consider processes of interest from the bottom up (i.e. through the eyes of the individual) rather than the top down.

By contrast, models for aggregations of individuals are concerned with characterising population-level behaviour. Such models may lack individual-level resolution but facilitate our understanding of large-scale processes that are composed of many individuals (e.g. Williams *et al.*, 2017). While individuals exhibit their own dynamics and characteristics, we may not account for individual-level behaviour in large-scale models due to simplifying assumptions meant to ease the implementation burden (Billari & Prskawetz, 2012). However, we may trade realism for tractability when we build statistical models directly from an aggregated perspective. When considering individual-level dynamics from a forward simulation perspective, we can very easily incorporate small-scale mechanisms that are difficult or impossible to parameterise directly in population-level models (Grimm & Railsback, 2005; Hooten *et al.*, 2010).

Thus, despite the fact that many potentially useful ABMs are trivial to implement from a forward simulation perspective, they are not often utilised directly in statistical models that formally assimilate data to learn about the data generating processes. This notable absence of ABMs in statistics is not due to the lack of usefulness of the model but rather the fact that ABMs bring several statistical challenges that require special consideration during implementation (Wikle & Hooten, 2015).

In what follows, we provide an overview of ABMs and highlight some of these statistical challenges and potential solutions in their implementation. To illustrate the statistical considerations, we provide an example of a simple individual-based demographic model commonly applied in the field of wildlife ecology. This ecological model is not usually referred to as an ABM but is often used to provide population-level inference about demographics while assimilating individual-level longitudinal data. This model can be implemented both from a conventional statistical perspective and using a suite of tools commonly associated with ABMs, and we can compare the inference obtained using both approaches. We then present a more complicated demographic ABM that mimics a human infectious disease process. This ABM is an individual-based version of a susceptible–infected–recovered–deceased (SIRD) model and illuminates additional challenges to implementation. We demonstrate the SIRD ABM with a case study involving COVID-19 on the cruise ship Diamond Princess (DP).

1.1 A Crash Course in Statistical Agent-Based Models

An ABM is an individual-based forward simulation model that takes both known and unknown input variables and provides output variables. Such models are based on interacting autonomous agents (individuals) that are assigned a particular state that varies with time based on simple deterministic or probabilistic rules associated with the agent's state, the states of other agents and the environment. The collective behaviour of the agents through time (and, potentially, space) then describe a complex system. Thus, the ABM itself could be deterministic or stochastic depending on the type of mechanisms we need to account for in the model. In the deterministic case, we need to assume a stochastic model that connects the ABM output to the observed data, and this is usually a choice of convenience (e.g. Hooten *et al.*, 2011; McDermott *et al.*, 2017). In cases where the ABM is inherently stochastic, the data model may be incorporated, or, if not, it could be included in the statistical implementation like it is for deterministic ABMs (e.g. Hartig *et al.*, 2011). Herein, we focus on stochastic ABMs as 'implicit statistical models' (Diggle & Gratton, 1984) and write them generically as $\mathbf{Y} \sim [\mathbf{Y}|\mathbf{X},\boldsymbol{\theta}]$, where $\mathbf{Y}$ are the output, $\mathbf{X}$ are the known inputs and $\boldsymbol{\theta}$ are the unknown inputs on which we seek inference. We note that the ABMs may be hierarchical and decomposed as $[\mathbf{Y}|\mathbf{X},\mathbf{Z},\boldsymbol{\theta}][\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}]$, where $\mathbf{Z}$ represents a latent process, although the conditional structure may not be useful in the statistical implementation.

The distinguishing feature of most ABMs is that the form of $[\mathbf{Y}|\mathbf{X},\boldsymbol{\theta}]$ is often difficult to express analytically, but simple to express computationally. In fact, we may not be able to use traditional implementation methods to fit the model to data. However, because we can simulate from the ABM easily, a variety of approaches have been developed to fit ABMs to data using simulation. Many of these approaches rely on ways to approximate the likelihood. For example:

Monte Carlo (MC) approximation of the ABM likelihood is easy to implement and may work well when the support of the data is discrete and finite and when the data are low dimensional. However, it requires the ability to simulate a large number of realisations from the ABM very quickly. In some cases, further approximation approaches such as pseudo-likelihood (PL) may be useful.

Approximate Bayesian computation (ABC) relies on a Markov chain Monte Carlo (MCMC) algorithm with a carefully selected discrepancy function between the observed and simulated data that is used in place of the true likelihood (which is unknown). The ABC approach requires us to simulate from the likelihood on each iteration of an MCMC algorithm and thus can be computationally intensive when the ABM simulator is slow. Also, the results are often sensitive to the choice of discrepancy function, which is sometimes arbitrarily chosen.

Emulation of an ABM can be helpful if it is slow to simulate from. To construct an emulator, we first design a computer experiment based on a limited number of simulations and record the paired inputs and outputs. Using those inputs and outputs as data, we can fit a phenomenological statistical model that serves as a surrogate (i.e. emulator) for the ABM. The emulator is selected to provide accurate predictions of the outputs given novel inputs, and we can use it in place of the ABM in a variety of ways.

Recursive Bayesian computation proceeds by grouping the data into partitions and computing the posterior in a sequence of stages. Using recursive Bayesian methods, we may be able to economise the computation associated with fitting Bayesian models. In particular, for statistical ABMs that simulate naturally partitioned data (such as time series), we can use recursive computing strategies that leverage distributed computing systems. Recursive Bayesian methods can also be combined with the other three approaches described here.

We demonstrate each of these approaches in what follows and highlight specific strengths and weaknesses as they arise in our two ABM examples. However, more generally, there are several key challenges to expect when fitting statistical ABMs to data, and we describe them here.

In their forward simulation form, ABMs may not be difficult to code in standard programming languages, but specific software packages such as HexSim, Mason, Netlogo and Starlogo may facilitate certain visualisations and automation (Wilensky, 1999; Luke *et al.*, 2003; Schumaker & Brookes, 2018). Some of this software has recently been ported to statistical languages (e.g. Bauduin *et al.*, 2019), but not all of them are easily coupled with statistical software.

Agent-based models explicitly simulate the dynamics of individual agents; thus, even modest amounts of model complexity can lead to slow algorithms as the number of agents and interactions increases. This can be attenuated using compiled programming languages and parallel computing where possible, but even while utilising those techniques, we can expect computational limitations. MC approximation methods can be especially demanding, requiring tens or hundreds of thousands of simulations for a single set of input parameters. When the dimension of the outputs increases, it becomes numerically challenging to approximate likelihoods for correlated data. In these cases, we may be able to use PL methods (e.g. Grazzini *et al.*, 2017) that make independence assumptions about the likelihood, but this induces another layer of uncertainty in the inference.

Many ABMs cannot be fit using traditional statistical methods for the aforementioned reasons. We can use a variety of approximation methods and software to fit statistical ABMs to data (e.g. Gramacy, 2016), but it is difficult to quantify the accuracy of these methods when we cannot compare with the exact inference. Furthermore, there are often a suite of subjective and somewhat arbitrary decisions made when implementing statistical ABMs such as the choice of data model (especially with deterministic ABMs), discrepancy function (ABC) and type of emulator (e.g. neural network, random forest and Gaussian process).

While ABMs are often easy to construct and we can incorporate mechanisms that lead to realistic simulations of data, they can quickly become overparameterised from a statistical perspective. Complications such as lack of identifiability, multimodality and overdispersion may exist. A thorough exploratory empirical analysis of the ABM behaviour can help isolate these issues, and knowledge-based informative priors usually improve ABM inference, but some issues may remain. Furthermore, while the ABM operates on the individual level, the observed data may only be available in aggregated form. Thus, important summary statistics of ABM output that connect directly to the observed data may occur at a different scale than the underlying dynamics (Jiang & Turnbull, 2004).

Finally, we note that conventional statistical model expressions lead to concise written descriptions, but ABMs are challenging to describe in scientific writing even though they are easy to represent in computer algorithms. Moreover, many ABMs are complicated enough that critical elements can be overlooked and undocumented. In these cases, the community of researchers working to develop ABMs often use strict protocols referred to as 'Overview, Design Concepts, and Details' when describing ABMs in such a way that they are reproducible (Grimm *et al.*, 2006,2010).

2 An Agent-Based Model for Wildlife Demography

To provide a realistic, yet tangible, example of an ABM for which we can demonstrate statistical challenges and solutions, we specify a hierarchical Cormack–Jolly–Seber (CJS) model (Cormack, 1964; Jolly, 1965; Seber, 1965). The CJS model is typically used to learn about individual-based survival of wildlife species based on a capture–recapture study design (Amstrup *et al.*, 2010).

In a typical capture–recapture study, we attempt to relocate n individuals during a set of T observation periods. The latent state of an individual is expressed as the binary variable $z_{i,t}$ for individuals $i = 1, \dots, n$ and time periods $t = 1, \dots, T$ (typically days, months or years, depending on the species). In its simplest form, the dynamics of the CJS model are specified using an individual-based binary mixture Markov process

$$z_{i,t} \sim \begin{cases} \text{Bern}(\phi) & , z_{i,t-1} = 1 \\ 0 & , z_{i,t-1} = 0 \end{cases}, \tag{1}$$

where $z_{i,t} = 1$ often implies the individual is alive and $z_{i,t} = 0$ implies the individual is dead. This dynamic framework can be generalised to accommodate epidemiological data with multiple categories such as susceptible, infected, recovered and dead as we illustrate in Section 4.

We recognise (1) as a dynamic zero-inflated Bernoulli model. In the CJS framework, the zero-inflation mechanism arises due to the constraint that an individual must remain dead in the periods after death (i.e. state 2 is an absorbing state). Thus, the parameter ϕ represents the individual-level survival probability between observation periods $t-1$ and t and is often the main focus of inference.

As mentioned previously, ABMs can include a specific observation process. This is the case with CJS models, which deviate from the traditional approaches to infer survival in human biostatistical analyses (e.g. Lin & Wei, 1989) because of the additional complexities in how wildlife capture–recapture data are acquired. In particular, during a survey at time t, a previously captured individual may not be detected. It is convention in these cases to record the data $y_{i,t}$

Table 1. *Probability (i.e. likelihood) under the Cormack–Jolly–Seber model for four individual capture histories (i.e. data $y_{i,t}$, for $t = 1, \ldots, 3$).*

i	$y_{i,1}, y_{i,2}, y_{i,3}$	Probability
1	1,1,1	$\phi \cdot p \cdot \phi \cdot p$
2	1,0,1	$\phi \cdot (1-p) \cdot \phi \cdot p$
3	1,1,0	$\phi \cdot p \cdot (\phi \cdot (1-p) + 1 - \phi)$
4	1,0,0	$(1-\phi) + \phi \cdot (1-p) \cdot (1 - \phi \cdot p)$

such that $y_{i,t} = 1$ indicates the individual was captured (or relocated) alive and $y_{i,t} = 0$ indicates the individual was not detected. Thus, for the cases when $y_{i,t} = 0$ is recorded, we do not know whether the individual was alive and went undetected or the individual was dead. To connect this measurement process with the latent individual-based process, we specify another conditional binary mixture model

$$y_{i,t} \sim \begin{cases} \text{Bern}(p) & , z_{i,t} = 1 \\ 0 & , z_{i,t} = 0 \end{cases}, \tag{2}$$

where p represents the detection probability for an individual i that is alive at time t. Typically, individuals are assumed alive upon first capture, thus $y_{i,1} = z_{i,1} = 1$ for all $i = 1, \ldots, n$. The data model expressed in (2) accounts for a non-ignorable missingness in typical capture–recapture data due to imperfect detectability.

The model expressed in (1)–(2) is a hidden Markov model (HMM Zucchini *et al.*2017) whose likelihood can be calculated using a suitably specified forward algorithm (see Supporting Information). However, the HMM approach to the CJS model was not commonly known, and hence not widely exploited until recently (Johnson *et al.*, 2016). Historically, the integrated likelihood for the CJS model (and all similar capture–recapture models) was computed either by working through the capture history directly or, if the model was implemented using Bayesian methods, an MCMC algorithm was constructed to sample the latent process $z_{i,t}$ from its full-conditional distribution (e.g. Royle & Dorazio, 2008; Hooten & Hefley, 2019). Both conventional approaches to fit the CJS model are cumbersome, especially for newcomers to these models, and thus, a variety of specialised automatic software was developed to make them more accessible to practitioners (White & Burnham, 1999). The need for specialised software may be surprising because, from an ABM perspective, the forward simulation CJS model using (1)–(2) could be programmed in a matter of minutes by someone with minimal coding skills.

To gain an appreciation for the complexity associated with implementing the CJS model, we express the likelihood as a product of probabilities for a set of capture histories for $n = 4$ example individuals based on $T = 3$ survey occasions in Table 1. The small number of individuals and capture occasions in this simple example facilitates the analytical calculation of the integrated likelihood for the CJS model. Although, even in this simple example, not all of the likelihood calculations in Table 1 are immediately apparent given the capture histories. In cases where capture histories are long and with numerous individuals, we must rely on software to work through the time series and check each condition to compute the likelihood correctly.

Expressing the CJS model as an HMM provides a modern alternative to calculate the likelihood that is computationally efficient. By constructing the appropriate state transition and conditional measurement matrices based on the hierarchical model specification in (1)–(2), we can apply the HMM forward algorithm to compute the CJS likelihood. The HMM is much like a Kalman filter that is used to compute likelihoods for Gaussian state-space models resulting in statistical algorithms that are more efficient for certain classes of models (McClintock *et al.*, 2020). However, despite its simplicity, the HMM framework and algorithms present yet another topic of study for the practitioner. Thus, once again, automated software has been developed to facilitate the use of HMM machinery to fit capture–recapture models (e.g. Johnson *et al.*, 2016).

The basic CJS model makes several strong assumptions about independence among individuals and that the conditional survival times are geometrically distributed. Relatively simple generalisations of an HMM may result in a hidden semi-Markov model (King & Langrock, 2016) or other classes of models. In Section 4, we present a more complicated epidemiological ABM that is based on fairly simple extensions of the basic CJS process.

In what follows, we discuss various approaches to treat the CJS model as an ABM and obtain statistical inference using nothing more than forward simulations from the CJS ABM. This example provides a perspective from which to consider more complicated ABMs such as the SIRD ABM, which is not easy to compute the likelihood for directly.

2.1 Implementing the Cormack–Jolly–Seber Model as an Agent-Based Model

Although the likelihood for the CJS model can be calculated directly, we can treat it as an ABM that can only be simulated from, which yields outputs (i.e. the binary response variables) given a set of inputs (i.e. initial conditions and parameter values). We can simulate the latent process and data from our CJS model specification by first setting $z_{i,1} = 1$ (because the individuals are assumed alive at time $t = 1$) and then sample $z_{i,t} \sim \text{Bern}(\phi \cdot z_{i,t-1})$ for $t = 2, \ldots, T$ and all individuals $i = 1, \ldots, n$ as a realisation of the process. We simulate $y_{i,t} \sim \text{Bern}(p \cdot z_{i,t})$ for $t = 2, \ldots, T$ and all individuals $i = 1, \ldots, n$ as a realisation of the data and discard the process $z_{i,t}$. As we will see, the binary support of the observations for a given individual and time can preclude some standard approaches such as Gaussian process emulation but can facilitate numerical approximation of the likelihood.

The first major aspect of statistical ABMs to consider is that we must rely on approximation in some aspect of the model itself, the resulting likelihood or the computational procedure used to fit the model. Moreover, these different types of approximation are not always mutually exclusive. For example, the simplest way to approximate the likelihood given the inputs (i.e. parameters) is to use MC approximation by simulating L independent realisations of $z_{i,t}^{(l)}$ and then $y_{i,t}^{(l)}$ for $l = 1, \ldots, L$ from the ABM using (1)–(2) and then compute

$$[\mathbf{y}_i | p, \phi] \approx \frac{\sum_{l=1}^{L} \mathbb{1}_{\{\mathbf{y}_i = \mathbf{y}_i^{(l)}\}}}{L}, \tag{3}$$

where $\mathbb{1}_{\{\mathbf{y}_i = \mathbf{y}_i^{(l)}\}}$ is an indicator equal to one when the observed and simulated time series for individual i are equal and zero otherwise. Table 2 shows the exact and MC approximated likelihood based on $L = 100\,000$ simulations from the CJS model for each capture history previously introduced. In this simple example, the MC approximation is quite accurate, and statistical inference could be obtained using the ABM simulator in lieu of the CJS likelihood.

As the observed time series grows (T gets large) however, the MC approximation may perform poorly because the exact arrangement of zeros and ones will become rare in the MC simulated sample and the condition $\mathbf{y}_i = \mathbf{y}_i^{(l)}$ in the indicator will not occur in some cases for any l. In these situations, we can resort to one of two additional approximation approaches: PL approximation or ABC.

The PL approximation breaks the dependence structure in the time series in this case and approximates the likelihood as a product over the MC approximated marginal likelihood components (Besag, 1974; Cox & Reid, 2004; Chandler & Bate, 2007). Using the same notation

Table 2. *Exact and approximated likelihood components under the Cormack–Jolly–Seber model with $p = 0.4$ and $\phi = 0.7$ for four individual capture histories (i.e. data $y_{i,t}$, for $t = 1, \ldots, 3$).*

		Likelihood			
i	$y_{i,1}, y_{i,2}, y_{i,3}$	Exact	MC	PL	ABC
1	1,1,1	0.0784	0.0792	0.0552	0.3983
2	1,0,1	0.1176	0.1176	0.1416	0.7986
3	1,1,0	0.2016	0.2014	0.2254	0.8824
4	1,0,0	0.6017	0.6014	0.5778	0.9208

ABC, approximate Bayesian computation; MC, Monte Carlo; PL, pseudo-likelihood.

and ABM simulation strategy as before, we can calculate the PL as

$$[\mathbf{y}_i | p, \phi] \approx \prod_{t=2}^{T} [y_{i,t} | p, \phi], \tag{4}$$

$$\approx \prod_{t=2}^{T} \frac{\sum_{l=1}^{L} \mathbb{1}_{\{y_{i,t} = y_{i,t}^{(l)}\}}}{L}. \tag{5}$$

The condition $y_{i,t} = y_{i,t}^{(l)}$ in the PL calculation is much more likely and leads to a more stable approximation as T grows; however, it introduces an additional layer of approximation. In Table 2, we can see that the PL approximation is only accurate to one decimal place whereas the joint MC approximation is accurate to three decimal places in most cases. Both the MC and PL approximations can be used in place of the likelihood in either a maximum likelihood or Bayesian setting. However, it may be challenging to obtain uncertainty estimates for model parameters using either approximation in the maximum likelihood setting without using a bootstrap or similar approach. Both approximations could be substituted in place of the likelihood in a standard MCMC algorithm in the Bayesian setting without issue. However, the loss of fidelity when using the PL is substantial enough to noticeably affect our inference about the model parameters p and ϕ, especially for longer time series that exhibit more dependence.

By contrast, the ABC approach can remedy the potential issue caused by high dimensionality of the support that leads to very unlikely simulations from the ABM that meet the condition $\mathbf{y}_i = \mathbf{y}_i^{(l)}$. The ABC approach introduces robustness in the likelihood approximation by using a suitably chosen discrepancy measure instead of a likelihood in an MCMC algorithm (e.g. Beaumont *et al.*, 2002; Marjoram *et al.*, 2003). For example, a discrepancy measure to substitute in for the CJS likelihood in the ABM simulation setting could relax the condition $\mathbf{y}_i = \mathbf{y}_i^{(l)}$, such that

$$[\mathbf{y}_i | p, \phi] \approx \frac{\sum_{l=1}^{L} \mathbb{1}_{\left\{d(\mathbf{y}_i, \mathbf{y}_i^{(l)}) \leq d^*\right\}}}{L}, \tag{6}$$

where $d(\mathbf{y}_i, \mathbf{y}_i^{(l)})$ is a distance measure between the data $\mathbf{y}_i$ and the simulation $\mathbf{y}_i^{(l)}$, and d^* is a user-specified threshold. Choi *et al.* (2010) summarised 76 different distance measures for binary data that we could use in (6). Two example discrepancy functions relevant for our CJS model are the simple matching distance and the Jaccard distance. For demonstration, we computed the approximate likelihood components using the ABC approach and simple matching distance using a threshold of $d^* = 0.5$ for our example in Table 2. The simple matching distance simplifies to the squared l^2-norm of $\mathbf{y}_i - \mathbf{y}_i^{(l)}$ divided by $T - 1$ for the CJS ABM. In this case, because T is small, the ABC likelihood approximation is not as accurate as the other methods shown in Table 2, but the ordering is the same despite the difference in magnitude.

There are other ways to use discrepancy measures to represent the likelihood in ABC. For example, when using a deterministic ABM whose output is continuous-valued, it is common to use a Gaussian kernel such as

$$[\mathbf{y}_i | p, \phi] \approx \exp\left(-\frac{1}{2}(\mathbf{y}_i - \mathbf{y}_i^{(l)})'(\sigma^2\mathbf{I})^{-1}(\mathbf{y}_i - \mathbf{y}_i^{(l)})\right), \tag{7}$$

where σ^2 is set by the user as a tuning parameter. However, for discrete-valued data, such discrepancy functions may not be appropriate. We note that the chosen discrepancy function may not integrate (or sum) to one, hence the reason for referring to the approach as 'computational' rather than formally statistical.

2.2 Implementing the Cormack–Jolly–Seber Model with a Likelihood Emulator

In cases where the ABM is computationally demanding to simulate from, we can approximate the ABM itself using a surrogate model to simulate approximate ABM output or the associated ABM likelihood (Gramacy, 2020). In essence, surrogate modeling, also known as statistical emulation (e.g. Kennedy & O'Hagan, 2000, 2001; Higdon *et al.*, 2008; Liu & West, 2009), involves three steps:

Conduct an 'experiment' with the original ABM to explore the output (i.e. data) given a range of input (i.e. parameters).
Fit a tractable statistical model (*f*) to the experimental results that predicts the output given the input.
Use the statistical model *f* (i.e. the surrogate) in place of the ABM to predict the data at new parameter values as needed to optimise the likelihood using one of the methods previously described (e.g. MC, PL or ABC).

Emulators introduce additional uncertainty, but we can account for that uncertainty in the final inference because we usually have a detailed understanding of the emulator model and its prediction uncertainty. In a Bayesian setting, the emulator is nested in a statistical model that integrates over the uncertainty in our emulator predictions and may also account for a calibration offset and a smooth emulator 'inadequancy' (i.e. bias) function (Kennedy & O'Hagan, 2001).

Emulators are most commonly used with models that have continuous-valued responses (e.g. Henderson *et al.*, 2009). In such cases, a large majority of emulators are based on Gaussian processes that predict a univariate surface (i.e. the response) over the multivariate parameter space (Gramacy, 2020). These are akin to geostatistical models often used to predict spatial surfaces over geographic space using a technique known as kriging (Cressie, 1990). Because Gaussian processes are parameterised in terms of second moments, they have sometimes been referred to as second-order emulators. Alternatively, first-order emulators may perform well as predictive models depending on the application (e.g. Hooten *et al.*, 2011; Leeds *et al.*, 2013). Furthermore, any statistical model can be used as an emulator, as long as it has the desired support and is capable of providing accurate predictions rapidly. Thus, machine learning methods such as random forests (Breiman, 2001), neural networks (e.g. Grzeszczuk *et al.*, 1999) and semiparameteric regression models (Wood, 2011) can be used as emulators.

Our CJS ABM, while simple, has several complexities to consider when constructing an emulator. In particular, our data (and ABM output) are binary, multivariate and correlated within individual. Thus, we would seek to use an emulator that is capable of mimicking those features. Because many emulators are developed for continuous-valued univariate data, off-the-shelf emulator models may not be available for even simple demographic models such as the CJS ABM.

As an alternative to conventional emulation strategies, we could use an emulator to predict the CJS likelihood (or some transformation of it such as the log likelihood) directly if it was capable of taking both the data $\mathbf{y}_i$ and the parameters p and ϕ as inputs. For example, suppose that we conducted a computer experiment based on a random sample of size J where $p^{(j)}$ and $\phi^{(j)}$, for $j = 1, \dots, J$, were sampled from their prior distributions and then simulated L MC realisations of the data $\mathbf{y}^{(j,l)}$ for $l = 1, \dots, L$ using the ABM as described in the previous section. Then we could use MC to approximate the associated log likelihood $\lambda^{(j)}$ by matching simulated realisations $\mathbf{y}^{(j,l)}$ with randomly drawn individual capture histories from the data $\mathbf{y}^{(j)}$ using the natural logarithm of (3).

Using the input and output of the computer experiment, we can fit the emulator model f, such that

$$\lambda^{(j)} \sim f(\lambda^{(j)} | \mathbf{y}^{(j)}, p^{(j)}, \phi^{(j)}, \boldsymbol{\gamma}), \tag{8}$$

for $j = 1, \dots, J$ and where $\boldsymbol{\gamma}$ are the unknown emulator parameters that depend on the choice of emulator model. After the emulator model is fit to the data associated with the computer experiment (which only used the ABM to simulate data), we can use it to predict the log likelihood component $\lambda_i^{(*)}$ rapidly given a capture history for individual i and parameter values $p^{(*)}$ and $\phi^{(*)}$.

The key strength of the emulation approach is that, after we have constructed the emulator, we no longer need the ABM. The emulator represents our best statistical understanding of the ABM dynamics and can be used in place of the ABM so that no additional simulation is necessary to fit the ABM to data (approximately) after the computer experiment has been conducted. Also, critically, the computer experiment itself can often be performed completely in parallel (i.e. across the J input/output pairs using the ABM), which reduces the computational time associated with the statistical emulation procedure substantially. Finally, and most importantly, obtaining predictions from the emulator is often much more efficient than simulating from the ABM directly, especially when using the emulator to predict the log likelihood.

To demonstrate the statistical emulation approach using the CJS ABM, we conducted a computer experiment consisting of $J = 1000$ input/output pairs. We sampled $p^{(j)}$ and $\phi^{(j)}$ from a Unif(0,1) distribution and sampled the associated capture history $\mathbf{y}^{(j)}$ randomly with replacement from the original data set (in this case consisting of the $n = 4$ capture histories from our continued example) for $j = 1, \dots, J$. For each element of the computer experiment, we defined the covariate vector $\mathbf{x}^{(j)} \equiv (\mathbf{y}'^{(j)}, p^{(j)}, \phi^{(j)})'$ and used the MC approximated log likelihood $\lambda^{(j)}$.

For comparison, we fit two emulator models to the computer experiment data. First, we fit a linear (LIN) regression $\lambda^{(j)} \sim \mathrm{N}(\mathbf{x}'^{(j)}\boldsymbol{\gamma}, \sigma^2)$ for $j = 1, \dots, J$. Second, we fit a two-layer Bayesian neural network (BNN) model with six neurons, probit activator function and Gaussian error distribution. Table 3 shows the results of approximating the likelihood using these two emulators, as compared with the exact likelihood calculation and the MC approximated likelihood (as previously reported in Table 2). In both cases, we predicted the log likelihood ($\hat{\lambda}^{(j)}$) directly and then exponentiated it ($e^{\hat{\lambda}^{(j)}}$) to obtain the approximation in Table 3 because we would use the log likelihood directly in an MCMC algorithm as the focal quantity. In this simple example, we can see that the LIN emulator is less accurate than the BNN emulator, but as first-order emulators, both perform well at predicting the shape of the likelihood and general order of the

Table 3. *Exact, Monte Carlo and emulator (linear and neural network) approximated likelihood components under the Cormack–Jolly–Seber model with $p = 0.4$ and $\phi = 0.7$ for four individual capture histories (i.e. data $y_{i,t}$, for $t = 1, \ldots, 3$).*

		Likelihood			
i	$y_{i,1}, y_{i,2}, y_{i,3}$	Exact	MC	LIN	BNN
1	1,1,1	0.0784	0.0792	0.0199	0.0803
2	1,0,1	0.1176	0.1176	0.0432	0.1204
3	1,1,0	0.2016	0.2014	0.2586	0.2243
4	1,0,0	0.6017	0.6014	0.5629	0.5911

BNN, Bayesian neural network; LIN, linear; MC, Monte Carlo.

components. Based on the emulator approximation results in Table 3, we would expect the final statistical inference for the ABM to be nearly indistinguishable using the Exact, MC or BNN approximation methods for the case when our time series are of length $T = 3$.

2.3 Recursive Bayesian Methods for Agent-Based Models

The concept of recursive model fitting is very natural from a Bayesian perspective because Bayesian models are designed to assimilate new data to update existing scientific information. For Bayesian implementations of naturally dynamic models such as most ABMs, we can envision fitting a sequence of models in time order such that the posterior based on all data up to time t can be found using the conditional likelihood of data at time t given all previous data (and parameters) and the posterior based on all previous data. This concept implies a natural recursion that facilitates fitting a model sequentially.

In what Hooten, Johnson, & Brost (2019) refer to as 'prior recursive Bayes', we seek a joint posterior distribution $[p,\phi|\mathbf{y}_{1:T}]$, where $\mathbf{y}_{1:T} = (y_1, \ldots, y_T)'$ represents all data (assuming a single individual for now). To obtain the joint posterior distribution associated with the CJS model, we can recursively compute

$$[p, \phi|\mathbf{y}_{1:t}] \propto [y_t|p, \phi, \mathbf{y}_{1:(t-1)}][p, \phi|\mathbf{y}_{1:(t-1)}], \tag{9}$$

$$\propto \left(\prod_{\tau=2}^{t}[y_\tau|p, \phi, \mathbf{y}_{1:(\tau-1)}]\right)[y_1|p, \phi][p, \phi]. \tag{10}$$

This recursive procedure involves the computation of $[p,\phi|y_1]$ and then relies on it as a prior to find the posterior $[p,\phi|\mathbf{y}_{1:2}]$ based on (9), and so on until we arrive at the desired full joint posterior $[p,\phi|\mathbf{y}_{1:T}]$.

Computationally, this procedure can be conducted in many ways including sequential MC (e.g. Chopin, 2002; Chopin *et al.*, 2013) and MCMC algorithms. Using a method called prior-proposal recursive Bayes (PPRB), Hooten, Johnson, & Brost (2019) showed how to obtain an MCMC sample from the full joint posterior using a sequence of computational stages. The procedure involves fitting a Bayesian model to an initial partition of data in a time series and then updating the posterior by recursively resampling from the previous stage MCMC sample.

The key advantage of the recursive Bayes approach for fitting the CJS model using only ABM simulation becomes clear for longer time series. As a first stage in the procedure, we fit a Bayesian model using an MCMC algorithm and approximate the likelihood associated with an initial partition of the capture histories based on only ABM simulations (e.g. $\mathbf{y}_{1:\tilde{t}}$ for $\tilde{t} = 3$). This first stage yields an initial MCMC sample of the parameters $p^{(k)}$ and $\phi^{(k)}$ from the posterior distribution $[p, \phi|\mathbf{y}_{1:\tilde{t}}]$ for $k = 1, \ldots, K$ MCMC iterations.

As a second stage, in parallel, we approximate the conditional data distributions $[y_t|p^{(k)}, \phi^{(k)}, \mathbf{y}_{1:(t-1)}]$ for all $t > \tilde{t}$ using the procedure described in what follows. Then, as a third stage, we use a sequence of MCMC algorithms that update the parameters using the Metropolis–Hastings ratio

$$r_t = \frac{[y_t|p^{(*)}, \phi^{(*)}, \mathbf{y}_{1:(t-1)}]}{[y_t|p^{(k-1)}, \phi^{(k-1)}, \mathbf{y}_{1:(t-1)}]}, \tag{11}$$

which relies on proposals $p^{(*)}$ and $\phi^{(*)}$ that are drawn randomly with replacement from the previous $(t-1)$ MCMC sample. Using this recursive Metropolis–Hastings sampling strategy, we let $p^{(k)} = p^{(*)}$ and $\phi^{(k)} = \phi^{(*)}$ with probability $\min(r_t, 1)$ and retain the previously accepted values otherwise.

There are two important elements to this recursive computing procedure. First, we can choose the initial partition size $\tilde{t}$ small enough to provide a fast and stable approximation of the likelihood. Second, with the initial MCMC sample from the first stage in hand, we can compute the conditional data distributions $[y_t|p^{(k)}, \phi^{(k)}, \mathbf{y}_{1:(t-1)}]$ for the remaining data partitions in parallel for every unique pair of $p^{(k)}$ and $\phi^{(k)}$. If we treat each new time point t, for $t > \tilde{t}$, as a partition, we can use a form of MC approximation to compute the conditional data distributions efficiently.

3 Case Study 1: Fit Cormack–Jolly–Seber Model Using an Agent-Based Model

For the data analysis that follows, we simulated data based on $n = 200$ individuals and $T = 10$ time periods with $p = 0.4$ and $\phi = 0.7$, yielding the individual-based capture histories shown in Figure 1. The number of capture occasions for which the simulated data were collected is large enough ($T = 10$) that a direct approximation of the full likelihood using MC methods is inefficient. Furthermore, PL approximations will be inaccurate due to the presence of temporal dependence in the data. Similarly, for ABC approaches, ecologists often desire more accurate posterior inference than we would obtain using a discrepancy function in the likelihood approximation as described in the previous section.

The degree of missingness in the data due to false negatives (i.e. $p = 0.4$) is substantial enough that a simpler model would not adequately characterise survival. As an ABM, the CJS model is numerically fast enough to simulate from that we do not need to use an emulator in this setting. Additionally, an emulator may not capture the dependence in the data well enough to provide an accurate approximation of the likelihood.

3.1 Computational Strategy

Considering the long list of caveats regarding the use of alternative CJS approximation methods, we used a recursive Bayesian computational strategy to fit the CJS model to the simulated data shown in Figure 1 using ABM simulation only. This procedure follows the aforementioned three computational stages.

For the first stage, we used MC approximation of the likelihood in an MCMC algorithm where, for a proposed parameter set $p^{(*)}$ and $\phi^{(*)}$, we simulated $L = 100\,000$ realisations of the

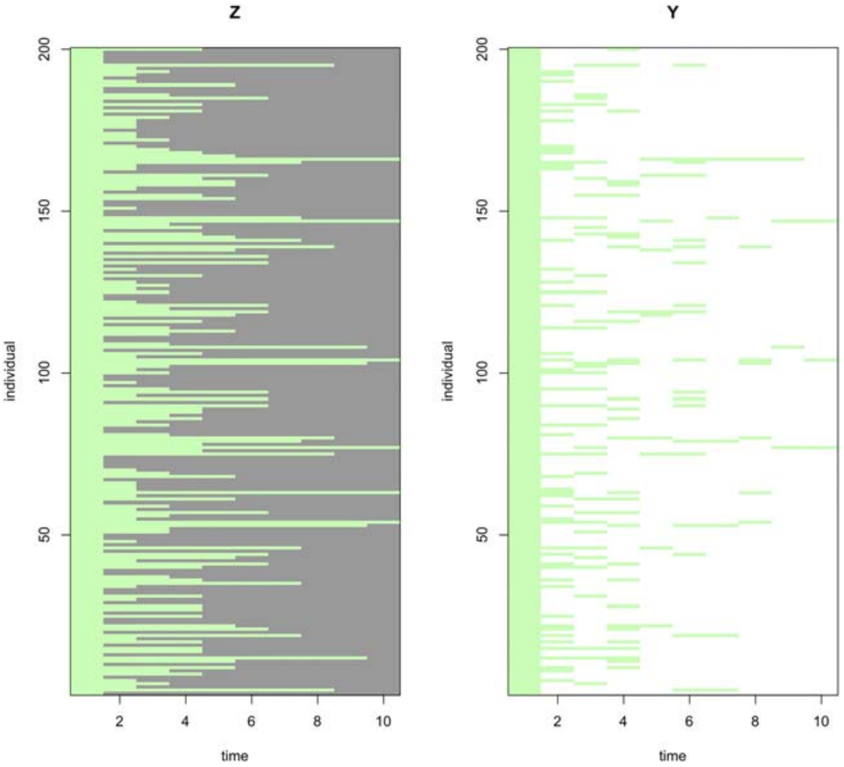

Figure 1. *Simulated individual-based survival processes (left) and data (right) from a Cormack–Jolly–Seber agent-based model. Green represents alive state, grey represents dead state and white represents undetected, based on simulation parameter values set at* $p = 0.4$ *and* $\phi = 0.7$. *[Colour figure can be viewed at* wileyonlinelibrary.com*]*

data resulting in $\mathbf{y}_{i,1:\tilde{t}}^{(*,l)}$ for $l = 1, \ldots, L$, $i = 1, \ldots, n$, and approximated the initial likelihood by

$$[\mathbf{Y}_{1:\tilde{t}}|p^{(*)}, \phi^{(*)}] \approx \prod_{i=1}^{n} \frac{\sum_{l=1}^{L} \mathbb{1}_{\{\mathbf{y}_{i,1:\tilde{t}}^{(*,l)} = \mathbf{y}_{i,1:\tilde{t}}\}}}{L}. \tag{12}$$

This MC approximation relies on a $\tilde{t}$-dimensional match of the data vectors, but as long as $\tilde{t}$ is small, the approximation is stable and accurate as we showed in the previous section. We used Metropolis–Hastings updates based on the likelihood approximation in (12) to acquire an MCMC sample of $K = 200\,000$ parameter values $p^{(k)}$ and $\phi^{(k)}$ for $k = 1, \ldots, K$.

In the second stage of the PPRB computational procedure, we approximated all conditional data distributions in parallel using the following sequence of steps. Because the data are binary, each conditional data distribution is Bernoulli such that $[y_{i,t}|p^{(k)}, \phi^{(k)}, \mathbf{y}_{i,1:(t-1)}] = \text{Bern}(\psi_{i,t}^{(k)})$ with success probability approximated as

$$\psi_{i,t}^{(k)} = \frac{\sum_{l=1}^{L} y_{i,t}^{(k,l)}}{L}, \tag{13}$$

where $y_{i,t}^{(k,l)} \sim [y_{i,t}|p^{(k)}, \phi^{(k)}, \mathbf{y}_{i,1:(t-1)}]$ for $l = 1, \ldots, L$ simulations. To simulate $y_{i,t}^{(k,l)}$, we draw in sequence: $z_{i,t-1}^{(k,l)} \sim [z_{i,t-1}|p^{(k)}, \phi^{(k)}, \mathbf{y}_{i,1:(t-1)}] = \text{Bern}(\omega_{i,t-1}^{(k)})$, then $z_{i,t}^{(k,l)} \sim$

$\text{Bern}(\phi^{(k)} z_{i,t-1}^{(k,l)})$ and finally, $y_{i,t}^{(k,l)} \sim \text{Bern}(p^{(k)} z_{i,t}^{(k,l)})$. This is effectively simulating two steps from the CJS ABM starting at time $t-1$ up to time t for each partition of the data. At each partition, we update probability $\omega_{i,t}^{(k)}$ as

$$\omega_{i,t}^{(k)} = \frac{\sum_{l=1}^{L} z_{i,t}^{k,l} \mathbb{1}_{\{y_{i,t}^{(k,l)} = y_{i,t}\}}}{\sum_{l=1}^{L} \mathbb{1}_{\{y_{i,t}^{(k,l)} = y_{i,t}\}}}. \tag{14}$$

The approximation in (14) is stable because we only have to perform the low-dimensional match $y_{i,t}^{(k,l)} = y_{i,t}$ rather than matching the entire preceding capture history for each individual.

Similarly, for the initial partition of data, we compute $\omega_{i,\tilde{t}}^{(k)}$

$$\omega_{i,\tilde{t}}^{(k)} = \frac{\sum_{l=1}^{L} z_{i,\tilde{t}}^{k,l} \mathbb{1}_{\{\mathbf{y}_{i,1:\tilde{t}}^{(k,l)} = \mathbf{y}_{i,1:\tilde{t}}\}}}{\sum_{l=1}^{L} \mathbb{1}_{\{\mathbf{y}_{i,1:\tilde{t}}^{(k,l)} = \mathbf{y}_{i,1:\tilde{t}}\}}}, \tag{15}$$

based on initial simulations from the CJS ABM that yielded $\mathbf{y}_{i,1:\tilde{t}}^{(k,l)}$ and $z_{i,\tilde{t}}^{k,l}$. This initial approximation in (15) is also relatively stable when $\tilde{t}$ is small.

In the third and final stage of the PPRB procedure, we used a sequence of MCMC algorithms with Metropolis–Hastings updates for the parameter pair (p,ϕ), using the acceptance ratio

$$r_t^{(k)} = \frac{\prod_{i=1}^{n} [y_{i,t} | p^{(*)}, \phi^{(*)}, \mathbf{y}_{i,1:(t-1)}]}{\prod_{i=1}^{n} [y_{i,t} | p^{(k-1)}, \phi^{(k-1)}, \mathbf{y}_{i,1:(t-1)}]}, \tag{16}$$

as previously described. For each new partition of data $\mathbf{y}_t = (y_{1,t}, \ldots, y_{n,t})'$, we use the precomputed conditional data distributions from the previous stage to calculate $r_t^{(k)}$ and update p and ϕ accordingly. This step is very efficient because all the computationally intensive quantities are stored and can be accessed rapidly.

3.2 Analysis of Cormack–Jolly–Seber Data

We applied the PPRB procedure to fit the CJS model to simulated data using only the ABM to simulate realisations of the data. In doing so, we first fit the model to the first $\tilde{t} = 3$ time steps of data using $K = 100\,000$ MCMC iterations and the MC approximated likelihood. This MCMC algorithm resulted in the diffuse marginal posterior distributions for p and ϕ shown in Figure 2.

Using the procedure described in the previous section, we approximated the conditional data models in parallel for $t = 4, \ldots, 10$ and all K MCMC realisations of $p^{(k)}$ and $\phi^{(k)}$. In the final stage of the PPRB procedure, we obtained Metropolis–Hastings proposals by randomly sampling with replacement from the previous MCMC sample in a sequence of simple MCMC algorithms for $t = 4, \ldots, 10$. Each marginal posterior distribution is shown as an overlapping blue density function for parameters p and ϕ in Figure 2. At time T, the resulting MCMC sample arises from the full posterior (dark blue outline), which agrees with posterior resulting from fitting the model jointly (black line) by assuming a knowledge of the CJS model as a statistical model. Thus, while the PPRB inference is approximate because we used MC to approximate the likelihood, it will be nearly indistinguishable from that when fitting the exact model.

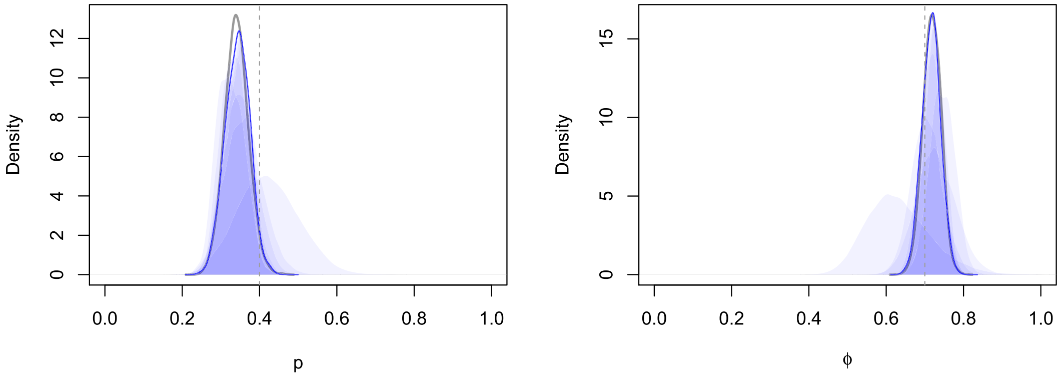

Figure 2. *Marginal posterior distributions for p (left) and ϕ (right). The posterior distributions shown as grey lines were acquired using a standard Markov chain Monte Carlo approach based on a complete understanding of the Cormack–Jolly–Seber model as a statistical model. The sequence of overlaid distributions represent the recursive posterior distributions based only on the Cormack–Jolly–Seber model as an agent-based model and start at time $\tilde{t} = 3$ (most diffuse) and end at time $T = 10$ highlighted by dark blue line. True parameter values used to simulate the data are shown as grey dashed vertical lines. [Colour figure can be viewed at* wileyonlinelibrary.com*]*

4 Case Study 2: An Epidemiological Agent-Based Model

Whereas the CJS model involves a dynamic individual-based process involving two main states (i.e. alive and dead), it is common in mathematical models of infectious disease to characterise the disease state of individuals via compartment labels such as susceptible, exposed, infected, recovered, removed and deceased (Bailey *et al.*, 1975; Hethcote, 2000). Perhaps the simplest of these compartment models is the susceptible–infected–recovered class of models, given that they account for a mechanistic transition among states that can exhibit realistic epidemic dynamics. Although these models are most often considered in a deterministic continuous-time, continuous-state context, they can also be considered as stochastic in the discrete time and discrete state-space to illustrate individual dynamics (e.g. Sattenspiel & Lloyd, 2009). Depending on the specification, the underlying process can be thought of as a discrete-state Markov, or semi-Markov, process with the stochastic state transitions across time based on probabilities or residence periods. In what follows, we describe an individual-based susceptible–infected–recovered model that includes mortality and density-dependent transmission of disease.

4.1 Agent-Based Susceptible–Infected–Recovered–Deceased Model

Let $z_{i,t} \in \{0,1,2,3\}$ correspond to the state of individual $i = 1, \ldots, N$ at time $t = 1, \ldots, T$, where state 0 is 'susceptible', state 1 is 'infected', state 2 is 'recovered' and state 3 is 'deceased'. Define $\pi_{j,k,i,t} \equiv P(z_{i,t} = k | z_{i,t-1} = j)$, and, for this example, assume that $\pi_{0,1,i,t} = \phi_{0,1,i,t}$ (become infected), $\pi_{0,0,i,t} = 1 - \phi_{0,1,i,t}$ (remain susceptible), $\pi_{0,2,i,t} = \pi_{0,3,i,t} = \pi_{1,0,i,t} = \pi_{2,3,i,t} = \pi_{3,1,i,t} = \pi_{3,2,i,t} = 0$, $\pi_{3,3,i,t} = 1$ (remain dead) and $\pi_{1,3,i,t} = \phi_{1,3}$ (die from infection). These transition probabilities correspond to Markovian dynamics and imply geometric residence time in each state. However, we allow for disease spread among individuals by inducing dependence in the transition from susceptible to infected with

$$\phi_{0,1,i,t} = \text{logit}^{-1}\left(\frac{\beta_0}{N}\left(\sum_{j \neq i} \mathbb{1}_{\{z_{j,t-1}=1\}} - N_v\right)^2 + \beta_1\right), \tag{17}$$

where $\mathbb{1}_{\{z_{j,t-1}=1\}}$ is an indicator that takes a value of 1 if the j-th individual at time $t-1$ is infected, and 0 otherwise. This infection probability function in (17) will rise until its peak when the number of infected individuals reaches N_v, after which it will decline again. This model specification assumes an intervention occurs at some point when authorities become aware of it and enforce remedial measures (e.g. a quarantine). The peak of $\phi_{0,1,i,t}$ will coincide with the inflection point in the epidemic.

We rely on a semi-Markov model for recovery where, after an individual becomes infected, they will recover τ_i days later if they do not die first (e.g. King & Langrock, 2016). We allow the individual-based residence time in the infected state to be stochastic such that $\tau_i \sim \text{Pois}(\lambda)$, where λ corresponds to the population-level intensity associated with recovery time. We also constrained the model so that individuals can only undergo a single transition per day.

To initialise the ABM for simulation in our application that follows, we assigned $z_{i,1}=1$ for $i=1,\ldots,10$ and $z_{i,1}=0$ for $i=11,\ldots,N$ to match the situation where a novel pathogen has just been introduced to the population. To account for the fact that public health agencies only report population-level aggregate data, we condensed the output of our SIRD ABM as $n_t = \sum_{\tilde{t}=1}^{t} \mathbb{1}_{\{z_{i,\tilde{t}-1}=0, z_{i,\tilde{t}}=1\}}$, the cumulative new infected individuals out of N total individuals for compartment $j=1$ and day t.

4.2 Computational Strategy

Public health data are not often made available at the individual level for privacy reasons. Thus, given the limited data that are available to fit models like our SIRD ABM and the relatively long time series compared with wildlife survey data, we describe an emulator approach for this case study.

Specifically, we let y_{t_i} represent the observed cumulative new infected individuals in a population on day t_i (vectorised as $\mathbf{y} = (y_{t_1},\ldots,y_{t_n})'$). A simplified Gaussian process emulator model can be written as

$$\mathbf{y} \sim \text{N}(\mathbf{K}\tilde{\mathbf{n}}, \sigma^2\mathbf{I}), \tag{18}$$

where $\tilde{\mathbf{n}} = (\tilde{\text{n}}_1,\ldots,\tilde{\text{n}}_T)'$ represents the cumulative new cases for all days in the study period and $\mathbf{K}$ is an $n \times T$ mapping matrix. Using a combined first-order and second-order emulator framework, we assume that $\tilde{\mathbf{n}}$ can be characterised by a Gaussian process with conditional distribution $\tilde{\mathbf{n}} \sim \text{N}(\tilde{\boldsymbol{\mu}}(\boldsymbol{\theta}), \tilde{\Sigma})$, where the mean and covariance are

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{\theta}) = \sum_{l=1}^{L} w^{(l)}(\boldsymbol{\theta})\mathbf{n}^{(l)}, \tag{19}$$

$$\tilde{\Sigma} = \sigma_n^2 \exp\left(-\frac{\mathbf{D}_n}{\gamma_n}\right), \tag{20}$$

and where $\boldsymbol{\theta}$ is a vector that includes all parameters in the ABM that are unknown and not specified as fixed. We refer to this type of surrogate model as an 'analogue emulator' because the conditional mean (19) of the latent process $\tilde{\mathbf{n}}$ is a weighted average of possible analogues $\mathbf{n}^{(l)}$, for $l=1,\ldots,L$, that arise from an *a priori* computer experiment associated with parameter values $\boldsymbol{\theta}^{(l)}$ (see McDermott & Wikle, 2016, for an overview of analogue methods). The weights are a function of proximity between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^{(l)}$ in parameter space, modulated by range parameters $\boldsymbol{\Gamma}_\theta$, such that

$$w^{(l)} = \frac{\exp(-(\boldsymbol{\theta}-\boldsymbol{\theta}^{(l)})'\Gamma_\theta^{-1}(\boldsymbol{\theta}-\boldsymbol{\theta}^{(l)}))}{\sum_{l=1}^{L}\exp(-(\boldsymbol{\theta}-\boldsymbol{\theta}^{(l)})'\Gamma_\theta^{-1}(\boldsymbol{\theta}-\boldsymbol{\theta}^{(l)}))}. \tag{21}$$

The temporal covariance matrix in (20) relies on pairwise temporal differences in the $T \times T$ matrix $\mathbf{D}_n$ and accounts for dependence in the process $\tilde{\mathbf{n}}$ not accounted for by the analogues.

In the first stage of a sequential emulation framework, we learn about the mean vector and covariance matrices in (19) and (20) by calibrating the emulator using the computer experiment input and output. For a set of input $\boldsymbol{\theta}^{(l)}$, we use the ABM to simulate associated output $n_t^{(l)}$ (for $l = 1, \ldots, L$, and vectorised in (19) as $\mathbf{n}^{(l)}$). Then, to calibrate the emulator, we optimised an aggregated loss function with respect to γ_n and $\boldsymbol{\Gamma}_\theta$. For computational efficiency, we maximised the product of emulator density functions described earlier over all $\mathbf{n}^{(l)}$ while conditioning on $\{\boldsymbol{\theta}^{(-l)}\}$ and $\{\mathbf{n}^{(-l)}\}$ (the sets of parameters and analogues excluding the l-th instance). Effectively, this uses each analogue $\mathbf{n}^{(l)}$ as data that depends on the other analogues to help us learn about the smoothness in the shapes of the analogues in the space of $\boldsymbol{\theta}$. Alternatively, we could write our analogue emulator jointly as a multivariate conditional autoregressive model and maximise the joint likelihood for all analogues simultaneously (e.g. Mardia, 1988; Sain *et al.*, 2011). For this case study and emulator, the multivariate conditional autoregressive approach would require excessive computational resources due to the massive dense covariance/precision matrices involved.

Because the computer experiment can be made as large as feasible, we can assume minimal uncertainty pertaining to γ_n and $\boldsymbol{\Gamma}_\theta$ and thus treat them as fixed when fitting the model in (18) using an MCMC algorithm (Liu *et al.*, 2009). In the second stage of this sequential implementation, we update the ABM parameter vector $\boldsymbol{\theta}$ using Metropolis–Hastings and use Gibbs updates for $\tilde{\mathbf{n}}$ and σ^2.

4.3 Analysis of COVID Data

To demonstrate the emulator implementation of the SIRD model described previously, we analysed data resulting from the COVID-19 outbreak aboard the cruise ship DP in early 2020. In particular, we focus on the confirmed cumulative case data as the response variable y_t (data repository provided in Acknowledgements) for days t in 5–10, 12–13, 15–20 and 26 February shown in Figure 3. The DP incident represents a nearly closed population consisting of $N = 3711$ passengers and crew. Using the confirmed cumulative case data, we focus our inference on the infection process parameters β_0 and β_1 of our SIRD ABM.

Previous studies suggest that the average time until recovery is approximately $\lambda = 13.5$ days (Ling *et al.*, 2020) and the daily death probability for infected individuals is approximately $\phi_{1,3} = 0.002$ (Russell *et al.*, 2020). Thus, we fixed those quantities in the model as well as $N_v = 371$ corresponding to approximately one-tenth of the population and near the inflection point, and let $\boldsymbol{\theta} = (\beta_0, \beta_1)'$ be unknown with prior $\boldsymbol{\theta} \sim \text{N}((-0.08, -3.37)', \text{diag}(0.1^2, 2^2))$ based on a range of parameter values that provide realistic population trajectories. We let the model error variance from (18) have the vague prior $\sigma^2 \sim \text{IG}(0.001, 0.001)$. In our computer experiment with the ABM, we used a combined regular grid of 400 combinations of $\boldsymbol{\theta}$ with a uniform random sample of 200 in the space $-0.3 \leq \beta_0 \leq 0.1$ and $-7 \leq \beta_1 < 0$ to obtain $L = 600$ simulated time series comprising $\mathbf{n}$. We fit the analogue emulator to the L sets of ABM input and output and used the resulting estimates for emulator parameters in the predictive distribution for $\tilde{\mathbf{n}}$ in the statistical emulator model (18). We then fit the statistical emulator model using a standard MCMC algorithm, updating $\tilde{\mathbf{n}}$, $\boldsymbol{\theta}$ and σ^2 for 20 000 iterations.

Pointwise posterior 95% credible intervals of $\tilde{\mathbf{n}}$ are shown in Figure 3 along with 95% credible intervals of population size in each of the four ABM compartments over the study period as derived quantities obtained by simulating from the SIRD ABM given posterior realisations of $\boldsymbol{\theta}$. The joint posterior distribution for $\boldsymbol{\theta}$ is shown in Figure 4. The posterior mean of

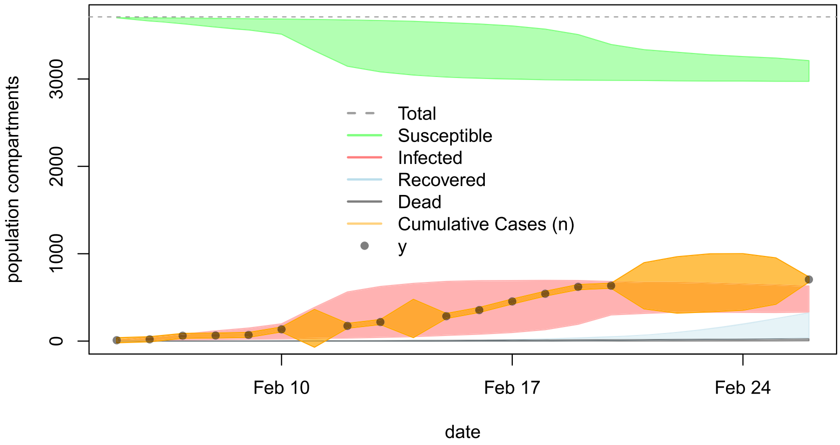

Figure 3. *Posterior pointwise 95% credible intervals for the population compartments and cumulative cases resulting from the COVID-19 outbreak on the cruise ship Diamond Princess. [Colour figure can be viewed at* wileyonlinelibrary.com*]*

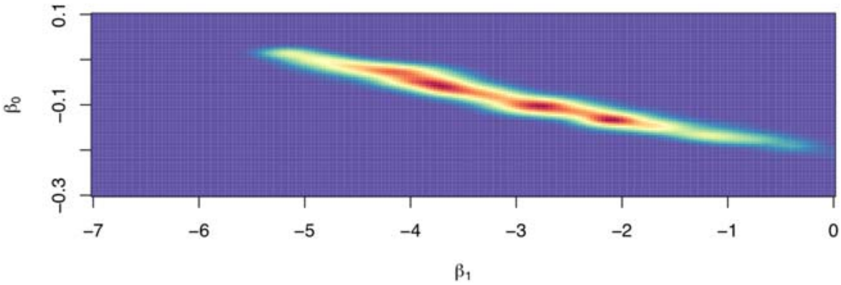

Figure 4. *Joint posterior distribution for the parameters $\boldsymbol{\theta}=(\beta_0,\beta_1)'$ controlling infection probability during the COVID-19 outbreak on the cruise ship Diamond Princess. [Colour figure can be viewed at* wileyonlinelibrary.com*]*

$E(\boldsymbol{\theta} = (\beta_0, \beta_1)'|\tilde{\mathbf{n}}) = (-0.084, -3.075)'$ provides evidence that β_0 induces a bell shape in $\phi_{0,1,i,t}$ with highest values between zero and approximately 700 cases and implies a low infection probability at the beginning of the outbreak and reaching its maximum ($\phi_{0,1,i,t} = 0.075$, the posterior mean of $\text{logit}^{-1}(\beta_1)$) on approximately 16 February 2020.

5 Discussion

We showed that a variety of approaches can provide approximate inference for ABMs when the likelihood may be intractable. Certain approximation methods may be more suitable depending on the characteristics of the ABM. For binary data with temporal structure and non-ignorable missingness such as those that may arise from a capture–recapture study of wildlife to infer survival, we showed that the support of the data presents both challenges and benefits when considered in an ABM context. We also showed that the CJS ABM is intuitive and its natural hierarchical structure lends itself to simulation.

We compared the inference resulting from the traditional computational approach to that using approximate methods and showed that we can achieve equivalent results from a forward simulation perspective that avoids extensive book-keeping and/or additional computational strategies (e.g. the HMM forward algorithm). It is straightforward to modify the CJS ABM to accommodate more realism in the ecological process. Even modest extensions can lead to very unwieldy statistical implementations of the model using conventional methods, as in our case study involving the SIRD ABM.

The SIRD ABM represents a substantial upgrade in complexity and represents a relatively realistic epidemiological process and data source. This increase in complexity creates additional challenges to consider in the implementation. For example, while the underlying dynamics in the SIRD ABM are individual based, the output consists of aggregated counts to match available public health data. These aggregated count data make it challenging to approximate the likelihood using standard methods; thus, we used an emulator approach to provide inference about ABM parameters for which we have less knowledge.

Beyond our SIRD ABM, additional extensions could be considered and would lead to similarly intractable likelihood functions that prohibit the use of conventional statistical methods to obtain inference. In the animal ecological setting, we might consider a spatial-interaction model. In the field of wildlife biology, animal movement modelling has increased in popularity in recent years (Hooten *et al.*, 2017), and some analyses have focused on interactions among moving individuals (e.g. McDermott *et al.*, 2017; Scharf *et al.*, 2018). Despite advances in this field of research, it is not common to consider both survival and animal movement models based on individual trajectories in the same model simultaneously. For example, we could consider the effect of proximity to competitors on individual-level survival by allowing the survival probability to depend on conspecific contact during the period $(t-1,t]$. In fact, finer-scale temporal changes in individual-level physiology (e.g. Hooten, Scharf, & Morales 2019) could also be accommodated.

In the epidemiological setting, movement of agents also plays an important role in infectious disease dynamics. Our SIRD ABM allows for interactions by assuming even mixing among all individuals in the population. That assumption could be generalised by allowing for heterogeneity in movement characteristics such that they vary with demographic groups. Furthermore, a detailed understanding of the environment the agents move through can lead to more realistic simulations of disease transmission (e.g. Pizzitutti *et al.*, 2018). For infectious diseases such as COVID-19, it may be appropriate to consider ABMs with additional compartments (e.g. susceptible–exposed–infected–recovered), and those may also present statistical challenges due to the expanded state-space (e.g. Okhuese, 2020).

Overall, ABMs are a useful way for scientists to express their understanding of the natural world but also present inferential challenges that provide ample opportunities for future statistical innovation. Statistical ABMs represent a solid point of connection between statisticians, computer scientists and application-focused researchers that will lead to productive and useful collaborations as we seek to find new ways to accommodate mechanisms in statistical models.

Acknowledgements

This research was funded by National Science Foundation (NSF) DEB 1927177, NSF DMS 1614392, NSF DMS 1811745 and Division of Social and Economic Sciences NSF SES 1853096. Any use of trade, firm or product names is for descriptive purposes only and does not imply endorsement by the US Government. DP data were acquired from https://github.com/thimotei/cCFRDiamondPrincess/blob/master/data/up_to_date.csv.

References

Amstrup, S.C., McDonald, T.L. & Manly, B.F.J. (2010). *Handbook of Capture-Recapture Analysis*. Princeton University Press: Princeton, New Jersey.

Bailey, N.T.J. et al. (1975). *The Mathematical Theory of Infectious Diseases and Its Applications.* Charles Griffin & Company Ltd: 5a Crendon Street, High Wycombe, Bucks HP13 6LE.

Bauduin, S., McIntire, E.J.B. & Chubaty, A.M. (2019). NetLogoR: A package to build and run spatially explicit agent-based models in R. *Ecography*, **42**(11), 1841–1849.

Beaumont, M.A., Zhang, W. & Balding, D.J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, **162**(4), 2025–2035.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *J. R. Stat. Soc. Ser. B (Methodological)*, **36**(2), 192–225.

Billari, F.C. & Prskawetz, A. (2012). *Agent-Based Computational Demography: Using Simulation to Improve Our Understanding of Demographic Behaviour.* Springer Science & Business Media: Heidelberg.

Breiman, L. (2001). Random forests. *Mach. Learn.*, **45**(1), 5–32.

Chandler, R.E. & Bate, S. (2007). Inference for clustered data using the independence loglikelihood. *Biometrika*, **94**(1), 167–183.

Choi, S.-S., Cha, S.-H. & Tappert, C.C. (2010). A survey of binary similarity and distance measures. *J. Syst. Cybern. Inform.*, **8**(1), 43–48.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, **89**(3), 539–552.

Chopin, N., Jacob, P.E. & Papaspiliopoulos, O. (2013). Smc2: An efficient algorithm for sequential analysis of state space models. *J. R. Stat. Soc. Ser. B*, **75**(3), 397–426.

Cormack, R.M. (1964). Estimates of survival from the sighting of marked animals. *Biometrika*, **51**(3/4), 429–438.

Cox, D.R. & Reid, N. (2004). A note on pseudolikelihood constructed from marginal densities. *Biometrika*, **91**(3), 729–737.

Cressie, N. (1990). The origins of kriging. *Math. Geol.*, **22**(3), 239–252.

Diggle, P.J. & Gratton, R.J. (1984). Monte Carlo methods of inference for implicit statistical models. *J. R. Stat. Soc. Ser. B*, **46**(2), 193–212.

Gramacy, R.B. (2016). laGP: Large-scale spatial modeling via local approximate Gaussian processes in R. *J. Stat. Softw.*, **72**(1), 1–46.

Gramacy, R.B. (2020). *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. CRC Press: Boca Raton, Florida.

Grazzini, J., Richiardi, M.G. & Tsionas, M. (2017). Bayesian estimation of agent-based models. *J. Econ. Dyn. Control*, **77**, 26–47.

Grimm, V., Berger, U., Bastiansen, F. et al. (2006). A standard protocol for describing individual-based and agent-based models. *Ecol. Model.*, **198**(1-2), 115–126.

Grimm, V., Berger, U., DeAngelis, D.L., Polhill, J.G., Giske, J. & Railsback, S.F. (2010). The odd protocol: A review and first update. *Ecol. Model.*, **221**(23), 2760–2768.

Grimm, V. & Railsback, S.F. (2005). *Individual-Based Modeling and Ecology*, Vol. **8.** Princeton University Press: Princeton, New Jersey.

Grzeszczuk, R., Terzopoulos, D. & Hinton, G.E. (1999). Fast neural network emulation of dynamical systems for computer animation. In *Proc. Conf. Neural Inf. Process. Syst.*, pp. 882–888.

Hartig, F., Calabrese, J.M., Reineking, B., Wiegand, T. & Huth, A. (2011). Statistical inference for stochastic simulation models–Theory and application. *Ecol. Lett.*, **14**(8), 816–827.

Henderson, D.A., Boys, R.J., Krishnan, K.J., Lawless, C. & Wilkinson, D.J. (2009). Bayesian emulation and calibration of a stochastic computer model of mitochondrial DNA deletions in substantia nigra neurons. *J. Am. Stat. Assoc.*, **104**(485), 76–87.

Hethcote, H.W. (2000). The mathematics of infectious diseases. *SIAM Rev.*, **42**(4), 599–653.

Higdon, D., Gattiker, J., Williams, B. & Rightley, M. (2008). Computer model calibration using high-dimensional output. *J. Am. Stat. Assoc.*, **103**(482), 570–583.

Hooten, M.B. & Hefley, T.J. (2019). *Bringing Bayesian Models to Life.* CRC Press: Boca Raton, Florida.

Hooten, M.B., Johnson, D.S. & Brost, B.M. (2019). Making recursive Bayesian inference accessible. *Am. Stat.*, 1–10.

Hooten, M.B., Johnson, D.S., Hanks, E.M. & Lowry, J.H. (2010). Agent-based inference for animal movement and selection. *J. Agric. Biol. Environ. Stat.*, **15**(4), 523–538.

Hooten, M.B., Johnson, D.S., McClintock, B.T. & Morales, J.M. (2017). *Animal Movement: Statistical Models for Telemetry Data.* CRC Press: Boca Raton, Florida.

Hooten, M.B., Leeds, W.B., Fiechter, J. & Wikle, C.K. (2011). Assessing first-order emulator inference for physical parameters in nonlinear mechanistic models. *J. Agric. Biol. Environ. Stat.*, **16**(4), 475–494.

Hooten, M.B., Scharf, H.R. & Morales, J.M. (2019). Running on empty: Recharge dynamics from animal movement data. *Ecol. Lett.*, **22**(2), 377–389.

Jiang, W. & Turnbull, B. (2004). The indirect method: Inference based on intermediate statistics–A synthesis and examples. *Stat. Sci.*, **19**(2), 239–263.

Johnson, D.S., Laake, J.L., Melin, S.R. & DeLong, R.L. (2016). Multivariate state hidden Markov models for mark-recapture data. *Stat. Sci.*, **31**, 233–244.

Jolly, G.M. (1965). Explicit estimates from capture-recapture data with both death and immigration-stochastic model. *Biometrika*, **52**(1/2), 225–247.

Kennedy, M.C. & O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, **87**(1), 1–13.

Kennedy, M.C. & O'Hagan, A. (2001). Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B*, **63**(3), 425–464.

King, R. & Langrock, R. (2016). Semi-Markov Arnason–Schwarz models. *Biometrics*, **72**(2), 619–628.

Leeds, W.B., Wikle, C.K., Fiechter, J., Brown, J. & Milliff, R.F. (2013). Modeling 3-D spatio-temporal biogeochemical processes with a forest of 1-D statistical emulators. *Environmetrics*, **24**(1), 1–12.

Lin, D.Y. & Wei, L.-J. (1989). The robust inference for the Cox proportional hazards model. *J. Am. Stat. Assoc.*, **84**(408), 1074–1078.

Ling, Y., Xu, S.-B., Lin, Y.-X. et al. (2020). Persistence and clearance of viral RNA in 2019 novel coronavirus disease rehabilitation patients. *Chin. Med. J.*, **133**, 1039–1043.

Liu, F., Bayarri, M.J. & Berger, J.O. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Anal.*, **4**(1), 119–150.

Liu, F. & West, M. (2009). A dynamic modelling strategy for Bayesian computer model emulation. *Bayesian Anal.*, **4**(2), 393–411.

Luke, S., Balan, G.C., Sullivan, K. & Panait, L. (2003). MASON: A fast discrete-event multiagent simulation library core in Java. https://github.com/eclab/mason/

Mardia, K.V. (1988). Multi-dimensional multivariate Gaussian Markov random fields with application to image processing. *J. Multivar. Anal.*, **24**(2), 265–284.

Marjoram, P., Molitor, J., Plagnol, V. & Tavaré, S. (2003). Markov chain Monte Carlo without likelihoods. *Proc. Natl Acad. Sci.*, **100**(26), 15324–15328.

McClintock, B.T., Langrock, R., Gimenez, O., Cam, E., Borchers, D.L., Glennie, R. & Patterson, T.A. (2020). Uncovering ecological state dynamics with hidden Markov models. arXiv preprint arXiv:2002.10497.

McDermott, P.L. & Wikle, C.K. (2016). A model-based approach for analog spatio-temporal dynamic forecasting. *Environmetrics*, **27**(2), 70–82.

McDermott, P.L., Wikle, C.K. & Millspaugh, J. (2017). Hierarchical nonlinear spatio-temporal agent-based models for collective animal movement. *J. Agri. Biol. Environ. Stat.*, **22**(3), 294–312.

Okhuese, A.V. (2020). Estimation of the probability of reinfection with COVID-19 by the susceptible-exposed-infectious-removed-undetectable-susceptible model. *JMIR Publ Health Surveill.*, **6**(2), e19097.

Pizzitutti, F., Pan, W., Feingold, B., Zaitchik, B., Álvarez, C.A. & Mena, C.F. (2018). Out of the net: An agent-based model to study human movements influence on local-scale malaria transmission. *PloS One*, **13**(3), e0193493.

Railsback, S.F. & Grimm, V. (2019). *Agent-Based and Individual-Based Modeling: A Practical Introduction.* Princeton University Press: Princeton, New Jersey.

Royle, J.A. & Dorazio, R.M. (2008). *Hierarchical Modeling and Inference in Ecology: The Analysis of Data from Populations, Metapopulations and Communities.* Elsevier: London, UK.

Russell, T.W., Hellewell, J., Jarvis, C.I. et al. (2020). Estimating the infection and case fatality ratio for COVID-19 using age-adjusted data from the outbreak on the Diamond Princess cruise ship. medrxiv. 2020 Mar 5. medRxiv DOI 10:05-20031773.

Sain, S.R., Furrer, R. & Cressie, N. (2011). A spatial analysis of multivariate output from regional climate models. *Ann. Appl. Stat.*, **5**(1), 150–175.

Sattenspiel, L. & Lloyd, A. (2009). *The Geographic Spread of Infectious Diseases: Models and Applications.* Princeton University Press: Princeton, New Jersey.

Scharf, H.R., Hooten, M.B., Johnson, D.S. & Durban, J.W. (2018). Process convolution approaches for modeling interacting trajectories. *Environmetrics*, **29**(3), e2487.

Schumaker, N.H. & Brookes, A. (2018). HexSim: A modeling environment for ecology and conservation. *Landsc. Ecol.*, **33**(2), 197–211.

Seber, G.A.F. (1965). A note on the multiple-recapture census. *Biometrika*, **52**(1/2), 249–259.

White, G.C. & Burnham, K.P. (1999). Program mark: Survival estimation from populations of marked animals. *Bird Study*, **46**(sup1), S120–S139.

Wikle, C.K. & Hooten, M.B. (2015). Hierarchical agent-based spatio-temporal dynamic models for discrete valued data. In *Handbook of Discrete-Valued Time Series*, Eds. Davis, R., Holan, S., Lund, R. & Ravishanker, N., Chapman & Hall/CRC Press: Boca Raton, FL. http://www.crcpress.com/product/isbn/9781466577732

Wilensky, U. (1999). A multi-agent programmable modeling environment. http://ccl.northwestern.edu/netlogo/

Williams, P.J., Hooten, M.B., Womble, J.N., Esslinger, G.G., Bower, M.R. & Hefley, T.J. (2017). An integrated data model to estimate spatiotemporal occupancy, abundance, and colonization dynamics. *Ecology*, **98**(2), 328–336.

Wood, S.N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *J. R. Stat. Soc; B*, **73**(1), 3–36.

Zucchini, W., MacDonald, I.L. & Langrock, R. (2017). *Hidden Markov Models for Time Series: An Introduction Using R.* CRC Press: Boca Raton, Florida.

19June2020 30June2020 01July2020

[*Received June 2020, Revised June 2020, Accepted July 2020*]

Supporting Information

Supporting information may be found in the online version of this article.