Epidemics 30 (2020) 100383

Contents lists available at ScienceDirect

Epidemics

journal homepage: www.elsevier.com/locate/epidemics

Choices and trade-offs in inference with infectious disease models

Sebastian Funk[a,b,*], Aaron A. King[c,d,e]

[a] Department of Infectious Disease Epidemiology, London School of Hygiene & Tropical Medicine, London WC1E 7HT, UK
[b] Centre for the Mathematical Modelling of Infectious Diseases, London School of Hygiene & Tropical Medicine, London WC1E 7HT, UK
[c] Department of Ecology and Evolutionary Biology, University of Michigan, Ann Arbor, MI, USA
[d] Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI, USA
[e] Department of Mathematics, University of Michigan, Ann Arbor, MI, USA

ARTICLE INFO

Keywords:
Inference
Infectious disease model
Bayesian
Frequentist
Model fitting

ABSTRACT

Inference using mathematical models of infectious disease dynamics can be an invaluable tool for the interpretation and analysis of epidemiological data. However, researchers wishing to use this tool are faced with a choice of models and model types, simulation methods, inference methods and software packages. Given the multitude of options, it can be challenging to decide on the best approach. Here, we delineate the choices and trade-offs involved in deciding on an approach for inference, and discuss aspects that might inform this decision. We provide examples of inference with a dataset of influenza cases using the R packages *pomp* and *rbi*.

1. Introduction

Mechanistic models of infectious disease dynamics (or *infectious disease models*) are built up from first principles to give an accurate reflection of underlying actual or hypothetical biological and social processes. Historically, these were infrequently confronted with particular data sets (Lessler et al., 2016). This could be seen as a contrast to models of statistical association (e.g., linear or logistic regression models) that are used as tools to investigate data, with the choice of models being driven by the data and hypotheses about relationships among variables. The mechanistic and statistical approaches, however, are not mutually exclusive. Through the advent of modern methods of inference, especially so-called plug-and-play or simulation-based approaches, and increasing availability of computing power it has become possible to perform statistically rigorous analysis using mechanistic models. With this development has come an increasing interest in the use of models for epidemiological forecasting and the design of public health policy (Heesterbeek et al., 2015). A wide range of methodologies and software packages are available for performing inference with models of infectious disease dynamics, and there are many active strands of methodological and computational development.

In the light of these developments, selecting a methodology and software can be a challenging task for researchers aiming to fit infectious disease models to data, and it is usually not obvious a priori what the best tool is for a given research question. Here, we aim to delineate the choices that have to be made in this process, the criteria that may be applied to facilitate these choice, as well as the trade-offs that are involved. For reviews of available methods and recent developments see, for example, Fasiolo et al. (2016) and Dattner and Huppert (2018).

2. Choices

2.1. Model

A probabilistic model takes input values (parameters, θ) and returns a probability distribution of output values (observations/data, Y). In the context of infectious disease models, the outputs can, for example, be case counts or deaths at different observation times, but can involve more complex variables such as pathogen genetic sequences or antibody titres (Rasmussen et al., 2014b; Clapham et al., 2016; Smith et al., 2017). One can write the probability of any particular output Y to occur when the model is run with parameters θ as

$$p(Y|\theta) \tag{1}$$

Instead of characterising this probability distribution, models usually prescribe the set of mathematical relationships or computational rules that take a given set of parameters θ and return outputs Y. Inference then consists of learning something about the parameters θ given a data set Y^* and a probabilistic model or set of models. Models are often implicitly or explicitly written as the product of two probability distributions

* Corresponding author at: Department of Infectious Disease Epidemiology, London School of Hygiene & Tropical Medicine, London WC1E 7HT, UK.
E-mail address: sebastian.funk@lshtm.ac.uk (A.A. King).

https://doi.org/10.1016/j.epidem.2019.100383
Received 25 March 2019; Received in revised form 29 September 2019; Accepted 11 December 2019
Available online 20 December 2019

$$p(Y|\theta) = p(Y|X, \theta)p(X|\theta) \quad (2)$$

where the first factor, $p(Y|X, \theta)$ is the *observation model* encoding measurement error, the second factor, $p(X|\theta)$ is the *process model* describing the behaviour of the system as a function of the parameters, and X are unobserved (or *latent*) model states. A model that can be written as in Eq. (2) is also called a *state-space model*.

Deciding on the model for inference involves a choice between *deterministic* and *stochastic* models. In the deterministic case, a given θ will always lead to the same system behaviour X. In other words, deterministic models have all the probability mass of $p(X|\theta)$ concentrated on the same X once θ is fixed. In stochastic models, many different outcomes X are associated with one set of parameters θ, and the probability with which these occur is given by the probability distribution $p(X|\theta)$. This distribution is often not available in closed mathematical form for infectious disease models, for example if they are formulated as a system of differential equations or as rates at which certain processes occur stochastically. In these cases, it is not possible to write down $p(X|\theta)$ explicitly but it is still possible to generate samples from that distribution by simulating the model. It is worth noting that for the purposes of inference even in deterministic models the relationship in the overall model in Eq. (2) is usually probabilistic because of the observation error encoded in $p(Y|X, \theta)$.

The choice between deterministic and stochastic modelling approaches is one aspect of the broader question of *model complexity*. Perhaps the most useful formalism in infectious disease epidemiology is that of the *compartmental model*, in which hosts are grouped according to their infection status and otherwise assumed to be in random contact with each other. Increasing the complexity of such models, one can in a compartmental model represent sub-populations (e.g., age or risk groups etc.) or various levels of biological realism as to how an infection is transmitted, all the way to so-called individual-based models, in which every host is explicitly represented and followed in the model.

2.2. Inference method

Inference with a given infectious disease model often revolves around using Eq. (1) to learn something about θ given a data set Y^*, that is to perform *parameter inference*. In *full information* methods, this is done by defining a function of θ called the *likelihood*,

$$L(\theta) = p(Y^*|\theta) \quad (3)$$

The value of θ that maximises $L(\theta)$ is called the maximum likelihood estimate (MLE). Intuitively, this is the set of parameters that makes the data most likely. Mathematically, the MLE is *asymptotically consistent*, that is, if the data are generated from $p(Y^*|\theta^*)$ the MLE converges to θ^* as the number of observations increases, and efficient, that is, no other estimator of θ^* has lower variance. Maximum likelihood estimation is one of the main methods used in modern *frequentist inference*.

In contrast, *Bayesian* methods perform inference by investigating the *posterior distribution* $p(\theta|Y)$. For a given data set Y^*, the posterior distribution is related to the likelihood via *Bayes' rule*,

$$p(\theta|Y^*) = \frac{p(Y^*|\theta)p(\theta)}{p(Y^*)} \quad (4)$$

where $p(\theta)$ is called the *prior distribution* and interpreted to encode any information that is available about the parameters before confronting the model with data. The denominator $p(Y^*)$ is also called the *evidence*, and reflects the overall compatibility of the model with the data. For parameter inference with a given model, this is constant and can be ignored. In Bayesian inference, the posterior distribution is often interpreted as the probability distribution of a random variable θ, and inference is conducted using *Monte-Carlo methods* that yield random samples from this probability distribution, that is, values of θ that are distributed according to $p(\theta|Y^*)$.

Different frequentist and Bayesian methods are appropriate depending on whether one is conducting inference with a deterministic or stochastic model. For deterministic models, the likelihood $p(Y^*|\theta)$ can be evaluated for any given θ by simulating the model and applying Eq. (2) to the resulting trajectory. In frequentist inference the maximum likelihood can then be estimated by applying a numerical optimisation method, for example the Nelder-Mead or downhill simplex method to find the parameter set θ that maximises $L(\theta)$ (Nelder and Mead, 1965). For Bayesian inference with deterministic models, Markov-chain Monte Carlo (MCMC) can be used, for example, to build a chain of θ values that are samples from $p(\theta|Y^*)$ (Gilks et al., 1995).

Parameter estimation with stochastic models requires alternative methods as the likelihood term $p(Y^*|\theta)$ cannot be simply evaluated by simulating the model with θ to yield a trajectory X and plugging this into the probability distribution for observation error $p(Y|X, \theta)$. Instead, many different trajectories X are possible given θ, each occurring with probability $p(X|\theta)$. To obtain the likelihood, one option is to use *data augmentation*, where the trajectories of unobserved states are estimated as part of the inference routine (e.g., using MCMC), analogously to the inference of parameters (McKinley et al., 2014).

Another option for obtaining the likelihood in stochastic models is to marginalise over the states X, that is

$$p(Y|\theta) = \int \mathrm{dX}\, p(Y|X)p(X|\theta) \quad (5)$$

where the integral is over all possible model trajectories (replaced by a sum if the space of possible model trajectories is discrete).

In many or indeed most cases of interest, a closed expression for $p(X|\theta)$ is not available. Instead, one can simulate the model for a given θ and obtain a sample from $p(X|\theta)$. Methods that only require the ability to perform such simulations are also said to have the *plug-and-play* property or be *simulation-based* (He et al., 2010). A common such method is to replace the integral in Eq. (5) with a Monte Carlo estimate (the so-called *pseudo-marginal* approach), especially by means of sequential Monte Carlo (SMC), also called Particle Filtering (Arulampalam et al., 2002). In a Particle Filter, multiple simulation runs (each a so-called particle) are performed in parallel. The simulations are weighted and resampled at every data point to yield an unbiased estimate of Eq. (5). The likelihood estimate given by the Particle Filter can be used as a basis for maximum-likelihood inference via Iterated Filtering (IF) (Ionides et al., 2006, 2015) or combined with MCMC to give particle Markov-chain Monte Carlo (pMCMC) (Andrieu et al., 2010;Akira Endo et al., 2019).

IF and pMCMC are two examples of so called full-information methods, that is they use all available data via the likelihood $p(Y|\theta)$. Other methods replace the likelihood calculation with summary statistics of the data, either because a decomposition as in Eq. (2) is not available, or to reduce the amount of information used in an attempt to lower the computational burden by requiring fewer simulation runs. The perhaps most prominent of these methods is Approximate Bayesian Computation (ABC) (Lintusaari et al., 2017; McKinley et al., 2018), while others include Synthetic Likelihood (Wood, 2010) and Bayesian History Matching (Andrianakis et al., 2015).

We have focused here on parameter inference for a given model but, more generally, inference can be conducted using a variety of models. In the context of infectious disease models, comparing the ability of different models to reproduce a given data set can give insights into underlying epidemiological mechanisms (King et al., 2008; He et al., 2010; Camacho et al., 2011; Rasmussen et al., 2014a).

2.3. Simulation method

Plug-and-play methods only require the user to be able to simulate a model. This involves a choice of simulation method, with different options available depending on whether one is studying a deterministic or stochastic model. Most deterministic models are formulated as

ordinary differential equations (ODEs), with a variety of ODE solvers of different accuracy and computational efficiency available on most computational platforms to simulate them to high numerical accuracy (Butcher, 2016). A simpler and usually faster way of simulating a deterministic model uses difference equations with a fixed time step, which can be seen as the discrete-time analogue of ODEs and can be used to approximate them (Fulford et al., 1997).

Stochastic models can be formulated as stochastic differential equations (SDEs), which extend ODEs to include random processes. SDEs can be solved using the Euler-Maruyama approximation (Kloeden and Platen, 1992; Milstein and Tretyakov, 2004) or various higher-order approximations. Alternatively, stochastic models can be formulated as discrete events and simulated using Gillespie's algorithm or suitable approximations such as fixed tau-leaping (Gillespie, 1977, 2001). More complex individual-based models may go beyond these methods and implement bespoke simulation schemes.

2.4. Software

Each choice of a method for simulation or inference comes with a choice of software implementation. Methods can be written from scratch in any available programming language or implemented using the range of libraries available in those languages that simplify this task. A more recent development are *probabilistic programming languages* (PPLs, van de Meent et al., 2018) that aim to automate the task of inference, only requiring the user to specify how to sample from the model $p(Y|\theta)$, usually using a particular syntax defined by the PPL. Reviewing the available methods and software libraries is well beyond the scope of this tutorial. Instead, we provide two examples of inference with infectious disease models implemented in the statistical programming language R at the end of this article.

3. Criteria

How does one choose a level of model complexity, a simulation method, an inference method and a programming language or library for conducting inference? These choices are generally not independent and therefore involve trade-offs. For example, the choice of software may restrict the researcher to a certain class of inference methods, simulation methods or model types that have been implemented. The desire to run a complex model may necessitate the use of an approximate method for inference and implementation using a programming language, library or software package that provides efficient simulation techniques. These choices often come down to a trade-off between the necessary (researcher and computer) *time* investment and the level of *confidence* in the obtained result. In the following sections, we set out the criteria that can be used to inform these choices.

3.1. Model adequacy

The adequacy of a model reflects how well it performs the task it has been designed to do, whether to understand the dynamics in a particular system, to test the impact of potential interventions or control strategies, or to make forecasts. These aspects are interrelated as, for example, understanding a system can affect the ability to make accurate forecasts, and the development of reliable control strategies can depend on both a good understanding and forecasting ability.

Model adequacy includes the validity of implicit and explicit assumptions, the ability of a model to make accurate predictions as well as of its parameters to reflect the real-world relationships they are supposed to encode (Lloyd, 2001; Wearing et al., 2005; Huppert and Katriel, 2013). In general, the type and choice of model should be guided both by the underlying research or policy question and by the nature and quantity of the available data. Beyond this, because the relationship of model complexity to model adequacy depends a great deal upon both the nature of the data and the dynamics of the model, little can be said in any degree of generality about the nature of these relationships. Deterministic models are generally inferior to stochastic approaches for scientific purposes as they are unable to appropriately represent uncertainty (King et al., 2015). At the same time, the widespread use of highly compartmentalised or individual-based models reflects both a desire for greater realism and the possibility of running such models in acceptable time with increasing computing power and parallelisation (Willem et al., 2017). Model complexity, however, can mask model misspecification when a model becomes flexible enough to reproduce a wide range of data sets without necessarily reflecting the data-generating process, and rigorously fitting such models to data can be prohibitively slow.

3.2. Computational efficiency

Computational efficiency in the broadest sense determines how many cycles of computation a given inference procedure takes to produce satisfactory results once it has been implemented. Conducting inference with infectious disease models usually consists of simulating from a model multiple times, that is, generating samples from $p(X|\theta)$ and confronting the results with the data via the observation model $p(Y|X, \theta)$.

The overall computational cost in conducting inference consists largely of the time need to run every individual simulation and the number of times that the model must be simulated in order to obtain results. The time cost of a model simulation depends on the level of model complexity, the simulation method and the efficacy of the implementation, including whether code is compiled (when using programming languages such as C/C++) or interpreted (when using programming languages such as R, although this has an interface for using C++ code to generate faster code). The number of times a model must be simulated depends on the inference method and desired or acceptable statistical accuracy.

The time it takes to conduct a given number of simulations can be reduced by performing computation in parallel. One potential advantage of using SMC for likelihood estimation with stochastic models, for example, is that it can be parallelised and therefore, potentially, overall computing time reduced. This includes the use of Graphical Processing units (GPUs), allowing parallelisation at a scale much larger than previously possible (Murray et al., 2016).

3.3. Statistical accuracy

Strictly speaking, all the inference methods presented here only yield approximations of the true quantities of interest, that is, the maximum likelihood estimate of θ given a data set Y^*, or the true posterior distribution $p(\theta|Y^*)$. How good this approximation is and, consequently, how much confidence a researchers can have in the results of the inference, depends on the method used. Most methods further require a certain amount of tuning of, for example, starting values (for maximum-likelihood optimisation and MCMC), proposal distributions (MCMC), cooling schedules (IF) or number of particles (IF and pMCMC), all of which affect statistical accuracy as well as computational efficiency. Full-information methods generally have maximal precision, while feature-based methods sacrifice a level of precision in order to achieve greater computational efficiency, or to make computation possible in the first place (Fasiolo et al., 2016).

3.4. Coding efficiency

There is a researcher time commitment involved in developing and running the inference procedure, generally involving the generation and testing of computer code. The time investment involved in this depends on one's familiarity with a given programming language or inference environment. Learning any new such environment comes with a time investment, often at the promise of future time savings

realised once the particular computing environment is mastered.

The individual time investment benefits from efficient programming practices. In particular, there are benefits to collaborative coding, even if collaboration is only with one's future self (Hogervorst, 2016). Writing efficient code goes well beyond producing the lines of code that performs a given task using the fewest computations possible but involves consistent error checking, testing, legibility, and reproducibility. Efficient computing practices are not generally taught as part of a scientific curriculum, and the adoption of efficient standards of collaborative coding often depends on the interest and background of individual researchers (Wilson et al., 2014, 2017).

4. Discussion

All the choices and criteria laid out here are strongly interrelated. Any decision on using a particular model structure, simulation method, inference method and software or library comes with trade-offs. Fasiolo et al. (2016) recommend to start with an approximate inference method before moving to full-information inference to generate final results. The reverse approach can be equally viable: beginning with a model suited to the questions of interest and testing to determine whether inference with a full-information methods is feasible could be seen to eliminate a potentially costly or even misleading first approximate step, while sitll leaving approximate methods available as a second option. In particular, when expense renders full-information methods infeasible, it is useful to consider where gains in efficiency might be made: It is usually the case that the questions of interest can be formulated in several mathematically different ways, each of which maps adequately onto the questions of interest, but some of which are more computationally efficient than others. Model simplifications can speed up computations, but simplifications that obscure the motivating questions are self-defeating. Selection of an alternative, approximate, inference method is an attractive option, since it avoids any distortion of the question that might arise from modifications of the model. At the same time, approximate inference methods implicitly reject certain features of the data as being non-informative, and it can be difficult to know a priori which features of the data are informative and which are not.

At the beginning of any inference is a data set and a model or set of candidate models. The role of model adequacy depends on the aim of the study and whether one's ultimate purpose is scientific hypothesis testing, forecasting or extrapolation. If one's purpose is purely to scientifically investigate one or more hypotheses about suspected mechanisms and whether there is evidence for them in the data, different model variants might be tested for their ability to reproduce the data. In that case, a negative finding may well be the most useful by making it possible to rule out a given mechanism. A determination of adequacy, on the other hand, does not necessarily support the conclusion that the hypothesised mechanisms are present. Indeed, one cannot guarantee that another model with different mechanisms would not be as much or more adequate. There is no a priori reason that certain mechanisms should or should not be incorporated in the model, and any such decisions must be guided by the hypotheses that are being investigated and prior assumptions about relevant mechanisms.

If the aim of a model is to provide forecasts of future incidence then the ability to make accurate predictions is a prerequisite for those aspects of the enterprise. In that case, it is not a given that mechanistic models are the best tool, and a model completely devoid of mechanisms might perform at least as well (Reich et al., 2019). However, if the aim is extrapolation, for example about the future impact of an intervention or a prediction about a different location, then both predictive ability and representation of relevant mechanisms are required.

Using off-the-shelf software solutions can limit researchers to using the particular model structures and methodologies that have been made available. Designing and developing an approach tailored to a given problem without recourse to existing solutions comes with greater flexibility but at the cost of greater coding time investment. At the same time, using existing libraries or packages as black box can cause avoidable errors due to a misunderstanding of the underlying methodologies.

Having said this, there are particular efficiency gains to be made from using and contributing to collaborative open-source platforms, two examples of which we introduce below. Instead of coding simulation and inference routines from scratch because the particular functionality is not implemented in the available tools for inference, we advocate that investigators engage in collaboratively improving these tools instead. This avoids the waste of researchers' time in redundant implementations of inference methods. It also helps ensure coding correctness and efficiency through mutual review and error checking.

5. Examples

We provide two examples of inference using a mathematical model of influenza transmission and a data set of an outbreak in a British boarding school. We use two different inference methods and software packages to fit the same process model to the data, using the same simulation method. Both examples are implemented in the R statistical programming language (Core Team, 2019) and included as supplementary material with this article. The examples are there to highlight the recommended steps when faced with a new dataset: explore the data, write down one or more candidate models, explore the parameter space, find suitable parameters for the chosen inference before method before running full inference. They also highlight the trade-offs in using different software platforms for inference. Trade-offs between computational and statistical efficiency have been discussed elsewhere (Fasiolo et al., 2016; Chatzilena et al., 2019). The latest version of the examples (including subsequent bug fixes and/or changes due to package updates) will be available at https://github.com/sbfnk/inference_pomp_rbi.

The first example uses Iterated Filtering via the IF2 algorithm as implemented in the *pomp* package, a platform for implementation of inference methods for state-space models (Ionides et al., 2015; King et al., 2016). The *pomp* package provides a range of frequentist and Bayesian simulation-based inference method and is fully integrated with R, that is, the simulation code and related probability densities can be provided as R functions, although in practice they are implemented as C snippets for faster processing. The second example uses pMCMC as implemented in the *rbi* package, an R interface to *LibBi*, a library for Bayesian inference on high-performance hardware and GPUs with a particular focus on parallel computing (Murray, 2015; Jacob and Funk, 2019). *LibBi* defines its own modelling language which is automatically translated into C++ code and compiled, and therefore native R code cannot be used to simulate models with *rbi*.

While both *pomp* and *LibBi* provide a range of inference methods, there is a focus on Iterated Filtering in *pomp* and on Bayesian SMC-based methods such as pMCMC in *LibBi*. Moreover, *LibBi* comes with the ability to parallelise the Particle Filter on GPUs. In our example, when we tested the overall time it took to generate a single pMCMC chain with *LibBi* using GPU computing, this was about 27 times less than the time it took to generate the same chain on a single CPU with *pomp*. While some of these speed gains may have been mitigated by running parallel chains on multiple CPUs, the ability to use GPU hardware is a strength of *LibBi* where the relevant hardware is available. At the same time, however, the implementation of iterated filtering in *LibBi* is rudimentary, and more generally *LibBi* is more limiting than *pomp* in the range of model structure and simulation methods that are available. For example, with *LibBi* it is not possible to use loops or multivariate probability distributions, or to use the Gillespie algorithm for stochastic simulation.

These examples serve to highlight that the use of any particular software package comes with advantages and drawbacks. It is worth emphasising that the speed at which a particular inference algorithm can be performed is only part of the picture. Really the quantity of

interest is time-to-completion of the overall calculation, but that involves a lot more than just the speed of the computational routines. It involves breadth of the prior, information in the data, dimension of the parameter space, number, disposition, and shape of the high-likelihood regions as well as the nature of the inference algorithms used.

Ultimately, the choice of software comes down to the preference of individual researchers, the methods they intend to use and how much they value computational speed over flexibility or ease of use given existing knowledge and expertise. In this context, we reiterate our call for researchers to become contributors to open-source software, be it by reporting errors, raising issues or providing improvements to code or new methodologies. This will result in better tools that are, ultimately, to the benefit of the whole research community in infectious disease dynamics and beyond.

Conflict interest

None declared.

Acknowledgements

SF acknowledges support from a Wellcome Trust Senior Research Fellowship in Biomedical Sciences (210758/Z/18/Z). The authors thank the UK National Institute for Health Research Health Protection Research Unit (NIHR HPRU) in Modelling Methodology at Imperial College London in partnership with Public Health England (PHE) for funding (grant HPRU-2012-10080).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.epidem.2019.100383.

References

Andrianakis, I., Vernon, I.R., McCreesh, N., McKinley, T.J., Oakley, J.E., Nsubuga, R.N., Goldstein, M., White, R.G., 2015. Bayesian history matching of complex infectious disease models using emulation: a tutorial and a case study on HIV in uganda. PLoS Comput. Biol. 11, e1003968.

Andrieu, C., Doucet, A., Holenstein, R., 2010. Particle Markov chain Monte Carlo methods. J. R. Stat. Soc. Se. B Stat. Methodol. 72, 269–342.

Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T., 2002. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Signal Process. 50, 174–188.

Endo, A., Leeuwen, E.V., Baguelin, M., 2019. Introduction to particle markov-chain monte carlo for disease dynamics modellers. Epidemics 29, 100363. https://doi.org/10.1016/j.epidem.2019.100363.

Butcher, J.C., 2016. Numerical Methods for Ordinary Differential Equations. John Wiley & Sons.

Camacho, A., Ballesteros, S., Graham, A.L., Carrat, F., Ratmann, O., Cazelles, B., 2011. Explaining rapid reinfections in multiple-wave influenza outbreaks: Tristan da Cunha 1971 epidemic as a case study. Proc. Biol. Sci. 278, 3635–3643.

Chatzilena, A., van Leeuwen, E., Ratmann, O., Baguelin, M., Demiris, N., 2019. Contemporary Statistical Inference for Infectious Disease Models Using Stan.

Clapham, H.E., Quyen, T.H., Kien, D.T.H., Dorigatti, I., Simmons, C.P., Ferguson, N.M., 2016. Modelling virus and antibody dynamics during dengue virus infection suggests a role for antibody in virus clearance. PLoS Comput. Biol. 12, e1004951.

Dattner, I., Huppert, A., 2018. Modern statistical tools for inference and prediction of infectious diseases using mathematical models. Stat. Methods Med. Res. 27, 1927–1929.

Fasiolo, M., Pya, N., Wood, S.N., 2016. A comparison of inferential methods for highly nonlinear state space models in ecology and epidemiology. Stat. Sci. 31, 96–118.

Fulford, G., Forrester, P., Jones, A., 1997. Modelling with Differential and Difference Equations. Cambridge University Press.

Gilks, W.R., Richardson, S., Spiegelhalter, D., 1995. Markov Chain Monte Carlo in Practice. CRC Press.

Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81, 2340–2361.

Gillespie, D.T., 2001. Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. 115, 1716–1733.

He, D., Ionides, E.L., King, A.A., 2010. Plug-and-play inference for disease dynamics: measles in large and small populations as a case study. J. R. Soc. Interface 7, 271–283.

Heesterbeek, H., Anderson, R.M., Andreasen, V., Bansal, S., De Angelis, D., Dye, C., Eames, K.T.D., Edmunds, W.J., Frost, S.D.W., Funk, S., Hollingsworth, T.D., House, T., Isham, V., Klepac, P., Lessler, J., Lloyd-Smith, J.O., Metcalf, C.J.E., Mollison, D., Pellis, L., Pulliam, J.R.C., Roberts, M.G., Viboud, C., Isaac Newton Institute IDD Collaboration, 2015. Modeling infectious disease dynamics in the complex landscape of global health. Science 347, aaa4339.

Hogervorst, R.M., 2016. Your Most Valuable Collaborator, Future-You. (archived at http://www.webcitation.org/75uNWbAy1) (Accessed 03 February 2019). https://rmhogervorst.nl/cleancode/blog/2016/05/26/content/post/2016-05-26-your-most-valuable-collaborator-future-you/.

Huppert, A., Katriel, G., 2013. Mathematical modelling and prediction in infectious disease epidemiology. Clin. Microbiol. Infect. 19, 999–1005.

Ionides, E.L., Bretó, C., King, A.A., 2006. Inference for nonlinear dynamical systems. Proc. Natl. Acad. Sci. U.S.A. 103, 18438–18443.

Ionides, E.L., Nguyen, D., Atchadé, Y., Stoev, S., King, A.A., 2015. Inference for dynamic and latent variable models via iterated, perturbed Bayes maps. Proc. Natl. Acad. Sci. U.S.A. 112, 719–724.

Jacob, P.E., Funk, S., 2019. rbi: R Interface to LibBi.

King, A.A., Domenech de Cellès, M., Magpantay, F.M.G., Rohani, P., 2015. Avoidable errors in the modelling of outbreaks of emerging pathogens, with special reference to ebola. Proc. Biol. Sci. 282, 20150347.

King, A.A., Ionides, E.L., Pascual, M., Bouma, M.J., 2008. Inapparent infections and cholera dynamics. Nature 454, 877–880.

King, A.A., Nguyen, D., Ionides, E.L., 2016. Statistical inference for partially observed Markov processes via the R Package pomp. J. Stat. Softw. 69.

Kloeden, P.E., Platen, E., 1992. Numerical Solution of Stochastic Differential Equations. Springer.

Lessler, J., Azman, A.S., Kate Grabowski, M., Salje, H., Rodriguez-Barraquer, I., 2016. Trends in the mechanistic and dynamic modeling of infectious diseases. Curr. Epidemiol. Rep. 3, 212–222.

Lintusaari, J., Gutmann, M.U., Dutta, R., Kaski, S., Corander, J., 2017. Fundamentals and recent developments in approximate Bayesian computation. Syst. Biol. 66, e66–e82.

Lloyd, A.L., 2001. The dependence of viral parameter estimates on the assumed viral life cycle: limitations of studies of viral load data. Proc. Biol. Sci. 268, 847–854.

McKinley, T.J., Ross, J.V., Deardon, R., Cook, A.R., 2014. Simulation-based Bayesian inference for epidemic models. Comput. Stat. Data Anal. 71, 434–447. https://doi.org/10.1016/j.csda.2012.12.012.

McKinley, T.J., Vernon, I., Andrianakis, I., McCreesh, N., Oakley, J.E., Nsubuga, R.N., Goldstein, M., White, R.G., 2018. Approximate Bayesian computation and simulation-based inference for complex stochastic epidemic models. Stat. Sci. 33, 4–18.

van de Meent, J.W., Paige, B., Yang, H., Wood, F., 2018. An Introduction to Probabilistic Programming. arXiv:1809.10756.

Milstein, G.N., Tretyakov, M.V., 2004. Stochastic Numerics for Mathematical Physics. Springer.

Murray, L.M., 2015. Bayesian state-space modelling on High-Performance hardware using LibBi. J. Stat. Softw. 67.

Murray, L.M., Lee, A., Jacob, P.E., 2016. Parallel resampling in the particle filter. J. Comput. Graph. Stat. 25, 789–805.

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. Comput. J. 7, 308–313.

R Core Team, 2019. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing.

Rasmussen, D.A., Boni, M.F., Koelle, K., 2014a. Reconciling phylodynamics with epidemiology: the case of dengue virus in southern Vietnam. Mol. Biol. Evol. 31, 258–271.

Rasmussen, D.A., Volz, E.M., Koelle, K., 2014b. Phylodynamic inference for structured epidemiological models. PLoS Comput. Biol. 10, e1003570.

Reich, N.G., Brooks, L.C., Fox, S.J., Kandula, S., McGowan, C.J., Moore, E., Osthus, D., Ray, E.L., Tushar, A., Yamana, T.K., Biggerstaff, M., Johansson, M.A., Rosenfeld, R., Shaman, J., 2019. A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the united states. Proc. Natl. Acad. Sci. U.S.A. 116, 3146–3154.

Smith, R.A., Ionides, E.L., King, A.A., 2017. Infectious disease dynamics inferred from genetic data via sequential Monte Carlo. Mol. Biol. Evol. 34, 2065–2084.

Wearing, H.J., Rohani, P., Keeling, M.J., 2005. Appropriate models for the management of infectious diseases. PLoS Med. 2, e174.

Willem, L., Verelst, F., Bilcke, J., Hens, N., Beutels, P., 2017. Lessons from a decade of individual-based models for infectious disease transmission: a systematic review (2006–2015). BMC Infect. Dis. 17.

Wilson, G., Aruliah, D.A., Brown, C.T., Chue Hong, N.P., Davis, M., Guy, R.T., Haddock, S.H.D., Huff, K.D., Mitchell, I.M., Plumbley, M.D., Waugh, B., White, E.P., Wilson, P., 2014. Best practices for scientific computing. PLoS Biol. 12, e1001745.

Wilson, G., Bryan, J., Cranston, K., Kitzes, J., Nederbragt, L., Teal, T.K., 2017. Good Enough Practices in Scientific Computing.

Wood, S.N., 2010. Statistical inference for noisy nonlinear ecological dynamic systems. Nature 466, 1102–1104.