



ELSEVIER

Contents lists available at ScienceDirect

Epidemics

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

epidmix—An interactive multi-model application for teaching and visualizing infectious disease transmission

Ulrich Muellner^{a,1}, Guillaume Fournié^{b,*,1}, Petra Muellner^a, Christina Ahlstrom^a,
Dirk U. Pfeiffer^{b,c}

^a *Epi-interactive, P.O. Box 15327, Miramar, Wellington, 6243, New Zealand*

^b *Veterinary Epidemiology, Economics and Public Health Group, Pathobiology and Population Sciences Department, Royal Veterinary College, Hatfield, AL9 7TA, UK*

^c *School of Veterinary Medicine, To Yuen Building, 31 To Yuen Street, City University of Hong Kong, Kowloon, Hong Kong*

ARTICLE INFO

Keywords:

Mathematical modelling

Epidemiology

Infectious disease

Application software

Disease dynamics

Teaching

ABSTRACT

Mathematical models of disease transmission are used to improve our understanding of patterns of infection and to identify factors influencing them. During recent public and animal health crises, such as pandemic influenza, Ebola, Zika, foot-and-mouth disease, models have made important contributions in addressing policy questions, especially through the assessment of the trajectory and scale of outbreaks, and the evaluation of control interventions. However, their mathematical formulation means that they may appear as a “black box” to those without the appropriate mathematical background. This may lead to a negative perception of their utility for guiding policy, and generate expectations, which are not in line with what these models can deliver. It is therefore important for policymakers, as well as public health and animal health professionals and researchers who collaborate with modelers and use results generated by these models for policy development or research purpose, to understand the key concepts and assumptions underlying these models.

The software application *epidmix* (<http://shinyapps.rvc.ac.uk>) presented here aims to make mathematical models of disease transmission accessible to a wider audience of users. By developing a visual interface for a suite of eight models, users can develop an understanding of the impact of various modelling assumptions – especially mixing patterns – on the trajectory of an epidemic and the impact of control interventions, without having to directly deal with the complexity of mathematical equations and programming languages. Models are compartmental or individual-based, deterministic or stochastic, and assume homogeneous or heterogeneous-mixing patterns (with the probability of transmission depending on the underlying structure of contact networks, or the spatial distribution of hosts). This application is intended to be used by scientists teaching mathematical modelling short courses to non-specialists – including policy makers, public and animal health professionals and students – and wishing to develop hands-on practicals illustrating key concepts of disease dynamics and control.

1. Introduction

Mathematical models of disease transmission conceptualise the spread of infectious agents within single or multiple host populations using mathematical language (Keeling and Rohani, 2007; Vynnycky and White, 2010). They are used to generate insights into the factors influencing disease dynamics, to identify knowledge gaps and to guide subsequent data collection. They are also used to assess the trajectory and scale of disease outbreaks, the impact of mitigation strategies and the uncertainty surrounding these outputs. By improving our understanding of how host, pathogen and environmental factors drive transmission, they have been transformative in our ability to control human and animal diseases (Heesterbeek et al., 2015), as well as plant

diseases (Cunniffe et al., 2015), and have had, within the last decades, an increasingly important role in guiding the design of policies and outbreak responses in both the public and animal health sectors (Cauchemez et al., 2016; Fraser et al., 2009; Houben et al., 2016; Keeling et al., 2001; WHO Ebola Response Team, 2014).

It has been highlighted that effective communication between epidemiologists and decision-makers is particularly critical during an outbreak, as is transparency to facilitate the acceptance of model outputs (Pfeiffer, 2010). However, as new data and computational resources become available, models are becoming increasingly complex, and require specific mathematical and computational expertise for implementation and interpretation. Their mathematical formulation makes their assumptions explicit, which requires an understanding of

* Corresponding author.

¹ These authors contributed equally to this work.

<https://doi.org/10.1016/j.epidem.2017.12.003>

Received 27 February 2017; Received in revised form 7 December 2017; Accepted 10 December 2017

1755-4365/ © 2017 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

the relevant mathematical methodologies. As a result, a model and its assumptions might be difficult to comprehend by those without the specialist mathematical knowledge. This seriously impedes efforts in communicating model results and their associated uncertainty and limits public and animal health professionals and policymakers' capacity to use such models.

It is therefore important for policymakers, as well as public and animal health professionals and researchers who collaborate with modelers and use results generated by these models for policy development or research purpose, to understand the underlying key concepts and assumptions. Several user-friendly software packages are available to non-specialists for developing mathematical models without requiring users to specify mathematical equations or be proficient in any programming language (e.g. Berkeley Madonna). Mastering these software packages may nonetheless require several hours to become familiar with their interface, which represents an obstacle to the development of short courses. Also, their user-friendly interface is often limited to the development of compartmental models; the development of individual-based models with more complex inter-host mixing patterns generally requires the user to specify mathematical equations. Further, visualization and analytic tools available for infectious disease epidemiology were recently summarized (Carroll et al., 2014), with the authors concluding that interactive graphics with dynamic features facilitate the exploration of complex data and are highly desired by the user.

In this paper, we present a mathematical modelling application, *epidemix*. Our objective was to develop a teaching tool allowing users to learn about key concepts of infectious disease dynamics, and to obtain an understanding of the impact of various modelling assumptions – especially mixing patterns – on the trajectory of an epidemic and the impact of control interventions. Knowledge in mathematics is not a prerequisite, and the step-by-step specification of model structure and parameters do not necessitate learning how to use the application or the reading of a user manual. Graphics and animations capturing changing outputs are produced as input variables and model settings are modified. This application is intended to be used by scientists teaching mathematical modelling short courses to non-specialists – including policy makers, public and animal health professionals and undergraduate students – and wishing to develop hands-on practicals illustrating key concepts of infectious disease dynamics and control.

2. Methods and results

2.1. Software

For easy access, the tool was developed to be available via the internet with a standard browser (<http://shinyapps.rvc.ac.uk>) without the installation of specialist software. RStudio Shiny (Chang et al., 2015) was selected as the base development framework which enabled publishing to the web as well as offline capability by using RStudio. The models were coded as R scripts. Each model code was wrapped into an R function which was called by the server-side script (server.R) of the RStudio Shiny application. This enabled a separation between the interface and the computation of the models. For example, the code of a model could be easily changed without affecting the application. To enable some of the functionality and visual elements of the core RStudio Shiny, functionality was extended by using JavaScript and CSS. In particular, NVD3 line charts were integrated to create dynamic graph outputs for the infection states over time.

The following R packages were utilised: Shiny (Chang et al., 2015), network (Butts, 2008), networkD3 (Allaire et al., 2017), ggplot2 (Wickham, 2009), GGally (Schloerke et al., 2016), sna (Butts, 2010), spatstat (Baddeley et al., 2015), nvd3, deSolve (Soetaert et al., 2010) and adaptivetau (Johnson, 2014).

The following functional elements were utilised:

- **Reactivity:** this core concept of RStudio Shiny enabled a dynamic adjustment of the model parameters triggering a re-calculation of the model data and outputs. Where possible, outputs were cached for faster computation.
- **Isolation:** The graphical outputs for the deterministic models could be displayed instantly after a parameter was changed by the user. As the stochastic models run through several simulations the time to compute the underlying data is significantly longer, therefore an “Update graph” button was introduced which just triggered a re-calculation of the model when a parameter changed. These input values were “isolated” from being reactive in this context. Certain input controls, e.g. time sliders or controls to select a certain simulation, were still reactive as they didn't require a re-calculation of the model.
- **Progress indicators:** For the stochastic models a progress bar was added providing the user with a visual indicator that a calculation is in progress. The progress bar was linked to the simulations to be processed.
- **Download data:** Model outputs were available for download in CSV format.
- **Default values:** On initial launch of the tool default parameter values were applied which represent a starting point for further adjustments.

Model formulation and R scripts were reviewed. The interface was tested by users of different background. This pilot testing was conducted to identify programming errors, inconsistency or application components that were unclear to the user. Feedback was accounted for to create the first version of *epidemix*, described here.

3.1. Model and parameter options

The application was designed such that users can choose from a suite of eight different model types. Additional details about the models and parameters are described in the Supplementary Material. In brief, all models simulate disease spread through a population of hosts. After having selected the type of model to be used, the user can define the characteristics of the host population, the infection states to be taken into account, the control strategies to be tested, and the associated model parameters. Parameters only become available after the appropriate model and model structure have been selected. Information buttons provide explanations of each model option and parameter.

3.2. Models

Models are either compartmental or individual-based, deterministic or stochastic, and assume homogeneous or heterogeneous-mixing patterns (Table 1). In the following, “unit” refers to the unit modelled for the host population of interest. For instance, this unit can be an individual if the host population of interest is a human population, or a cow or a cattle farm if the spread of a disease is simulated within a cattle farm or a group of cattle farms, respectively. In compartmental

Table 1
The eight model types included in the *epidemix* application.

Model	Deterministic/ Stochastic	Compartmental/Individual- based	Mixing
1	Deterministic	Compartmental	Homogeneous
2	Deterministic	Compartmental	Two-group
3	Stochastic	Compartmental	Homogeneous
4	Stochastic	Compartmental	Two-group
5	Stochastic	Individual-based	Homogeneous
6	Stochastic	Individual-based	Two-group
7	Stochastic	Individual-based	Network
8	Stochastic	Individual-based	Spatial

Download English Version:

<https://daneshyari.com/en/article/8644151>

Download Persian Version:

<https://daneshyari.com/article/8644151>

[Daneshyari.com](https://daneshyari.com)