



BICI v0.4: Bayesian Individual-based Compartmental Inference

Christopher M. Pooley¹, Grant Henderson¹, Andrea B. Doeschl-Wilson² and Glenn Marion¹

¹ Biomathematics and Statistics Scotland, James Clerk Maxwell Building, The King's Buildings, Peter Guthrie Tait Road, Edinburgh, EH9 3FD, UK.

² The Roslin Institute, The University of Edinburgh, Midlothian, EH25 9RG, UK.

Table of Contents

1) INTRODUCTION	6
1.1 DOWNLOADING	6
1.2 THE USER	6
1.3 GETTING STARTED.....	7
1.3.1 Examples	8
1.4 TUTORIAL.....	8
1.4.1 Setting up a simple SIR model.....	8
1.4.2 Simulation	9
1.4.3 Generating simulated data	10
1.4.4 Inference	11
1.4.5 Posterior simulation.....	12
2) MODEL	14
2.1 COMPARTMENTAL MODEL.....	14
2.1.1 Species.....	14
2.1.2 Classifications.....	14
2.1.3 Compartments	15
2.1.4 Transitions.....	15
2.1.5 Sources and sinks	16
2.1.6 Branching	17
2.1.7 Building up a compartmental model.....	17
2.1.8 Annotations.....	17
2.1.9 Importing	18
2.1.10 Equations	19
2.2 PARAMETERS.....	22
2.2.1 Parameter summary	22
2.2.2 Individual effects	22
2.2.3 Fixed effects	23
2.2.4 Splines	23
2.2.5 Constants	24
2.2.6 Reparameterisations.....	24
2.2.7 Distributions.....	24
2.2.8 Derived.....	25
2.2.9 Factors.....	25
2.3 DESCRIPTION	27
3) SIMULATION.....	27
3.1 INITIAL CONDITIONS	27
3.1.1 The initial population	27
3.1.2 Adding populations to the system.....	28
3.1.3 Removing populations from the system.....	29
3.1.4 Adding individuals.....	29
3.1.5 Removing individuals	29
3.1.6 Moving individuals	30
3.2 PARAMETERS.....	30
3.3 RUN.....	31
3.3.1 Further options.....	32
3.4 RESULTS	32

3.5 GENERATING DATA	33
3.5.1 Simulated initial conditions data	33
3.5.2 Simulated individual-level data.....	34
3.5.3 Simulated population-level data.....	35
3.5.4 Simulated additional data.....	35
4) INFERENCE.....	36
4.1 DATA	36
4.1.1 Initial conditions data	37
4.1.2 Individual-level data.....	37
4.1.3 Population-level data.....	40
4.1.4 Additional data	42
4.1.5 Table editing	42
4.2 PRIOR.....	43
4.2.1 Split	44
4.2.2 Text representation for a prior or distribution	44
4.3 RUN.....	45
4.3.1 Inference algorithms	45
4.3.2 Further options.....	47
4.4 RESULTS	48
4.4.1 Parameters.....	49
4.4.2 Populations	49
4.4.3 Transitions.....	50
4.4.4 Individuals	50
4.4.5 Diagnostics.....	51
4.4.6 Estimating the Bayes factor	52
4.4.7 MCMC diagnostics	52
5) POSTERIOR SIMULATION	54
5.1 POPULATION MODIFICATION.....	54
5.2 PARAMETER MULTIPLIERS.....	54
5.3 RUN.....	54
5.4 RESULTS	54
6) INPUTS AND OUTPUTS.....	55
6.1 BICI ARCHITECTURE	55
6.2 LOADING AND SAVING BICI FILES	55
6.3 EXPORTING	56
6.4 RUNNING ON A LINUX CLUSTER	56
7) BICI-SCRIPT	57
7.1 A SIMPLE EXAMPLE	58
7.2 DIRECTLY RUNNING BICI CORE CODE	58
7.3 BICI-SCRIPT OUTPUTS	59
7.3.1 Parameter samples	59
7.3.2 State samples.....	61
8) EXAMPLES	61
M1 SIMPLE EPIDEMIOLOGICAL MODELS	62
M1.1: SI population-based model (PBM)	62

M1.2: SI individual-based model (IBM)	62
M1.3: SIR model (IBM)	62
M1.4: SIR model with Erlang distribution (PBM)	63
M1.5: SIR model with gamma distribution (IBM)	63
M1.6: SEIR model with exposed period (IBM)	63
M2 ADDITIONAL EPIDEMIOLOGICAL MODELS	64
M2.1: SIRD model with branching using transition rates (PBM).....	64
M2.2: SIRD model with branching probability (IBM)	64
M2.3: SIRD model with branching factors (IBM).....	64
M2.4: SIS model (PBM)	65
M2.5: SIRS model with waning immunity (IBM)	65
M2.6: SIR model with demographic stratification (PBM)	66
M2.7: SIR model with differential infectivity and demographic stratification (PBM)	66
M2.8: SIR model with demographic stratification (IBM).....	66
M3 SPATIAL EPIDEMIOLOGICAL MODELS	67
M3.1: Metapopulation model using geographical regions (PBM).....	67
M3.2: Metapopulation model using geographical points (PBM).....	67
M3.3: Metapopulation model using a distance kernel (PBM)	68
M3.4: Farm-based model using a distance kernel (IBM)	68
M4 ECOLOGICAL MODELS	69
M4.1: Logistic growth population model (IBM)	69
M4.2: A predator-prey model (PBM).....	69
M4.3 A spatial diffusion model (IBM)	70
M4.4: A species presence/absence distribution model (IBM)	70
M5 DISEASE TRANSMISSION EXPERIMENTS.....	70
M5.1: Single contact group, investigating susceptibility (IBM).....	70
M5.2: Multiple contact groups, investigating infectivity (IBM)	71
M5.3: Quantitative genetics model for susceptibility/infectivity (IBM)	71
M5.4: Environmental pathogen accumulation model (IBM/PBM)	72
M6 COVID-19 MODELS	72
M6.1: Simple (PBM)	72
M6.2: Age-structured model (PBM).....	73
A SIMULATION FEATURES.....	73
A1: Multiple simulations (PBM, M1.1)	73
A2: Uncertain initial conditions — single classification (PBM, M1.1)	73
A3: Uncertain initial conditions — multiple classifications — focal selected (PBM, M2.6).....	74
A4: Uncertain initial conditions — multiple classifications — total population selected (PBM, M2.6)	74
A5: Add / remove individuals (PBM, M2.6).....	75
A6: Add / move / remove individuals (IBM)	75
B) POPULATION-LEVEL DATA TYPES	75
B1: Time series population observations (PBM, M1.1)	75
B2: Time series population-level transition observations (PBM, M1.1)	76
B3: Stratified time series population observations (PBM, M2.6)	76
B4: Stratified population observations from multiple compartments (PBM, M2.6)	76
B5: Combined population-based data sources in a COVID-19 model (PBM, M6.1)	77
B6: Time series population observations (IBM, M1.2)	77
C) INDIVIDUAL-LEVEL DATA TYPES	77
C1: Known transition events — infection and recovery (IBM, M1.3)	77
C2: Incomplete transition events - recovery only (IBM, M1.3).....	77

<i>C3: Compartmental observations (IBM, M1.3)</i>	78
<i>C4: Disease diagnostic test results (IBM, M1.3).....</i>	78
<i>C5: A partially observed transition (IBM, M1.3).....</i>	78
<i>C6: A transition observed over a time window (IBM, M1.3)</i>	78
<i>C7: A transition observed in a demographic category (IBM, M2.8).....</i>	79
<i>C8: Uncertain compartmental observations (IBM, M1.3).....</i>	79
D) TIME VARIATION	79
<i>D1: Time variation in transmission rate (PBM, M1.1).....</i>	79
<i>D2: Time variation in transmission rate using a trigonometric function (PBM, M1.1).....</i>	79
<i>D3: Time variation in transmission rate through a covariate (PBM, M1.1).....</i>	80
<i>D4: Time variation in population-level transition observation probability (PBM, M1.1).....</i>	80
<i>D5: Time variation in individual transition observation probability (PBM, M1.2)</i>	80
<i>D6: Time variation in population observation probability (PBM)</i>	80
<i>D7: Time variation in branching probability (PBM, M2.1)</i>	81
<i>D8: Time-varying covariate affecting branching probability (IBM, M2.2)</i>	81
E) INDIVIDUAL-BASED VARIATION	81
<i>E1: Individual fixed effect applied to a transition (IBM, M1.2)</i>	81
<i>E2: Fixed-effect applied to a population (IBM)</i>	82
<i>E3: Individual effect applied to a transition (IBM, M1.2)</i>	82
<i>E4: Correlated individual effect applied to a transition (IBM, M1.2)</i>	82
<i>E5: Correlated individual effect applied to a population (IBM, M5.3)</i>	82
<i>E6: Fixed effect applied to a branching probability (IBM, M2.3)</i>	83
<i>E7: Individual effect applied to a branching probability (IBM, M2.3)</i>	83
F) PARAMETER DEFINITIONS	84
<i>F1: Reparameterisation (IBM, M3.4)</i>	84
<i>F2: Parameter distribution (IBM, M5.2).....</i>	84
<i>F3: Derived quantities (IBM, M1.3)</i>	85
G) INCORPORATING PATHOGEN GENETICS	85
<i>G1: Matrix of genetic differences (IBM, M1.2).....</i>	85
9) LICENSE AND WARRANTY	85
10) CITING BICI	85
ACKNOWLEDGMENTS.....	85
REFERENCES	86
APPENDICES	87
APPENDIX A: BICI-SCRIPT COMMANDS ORDERED BY SECTION.....	87
APPENDIX B: BICI-SCRIPT COMMANDS ALPHABETICALLY ORDERED.....	89
APPENDIX C: PARAMETER VISUALISATIONS.....	107
APPENDIX D: POPULATION VISUALISATIONS.....	112
APPENDIX E: TRANSITION VISUALISATIONS	115
APPENDIX F: INDIVIDUAL VISUALISATIONS	116

1) Introduction

Compartmental models have long been used as a means of understanding the collective dynamics of interacting agents, with notable applications in epidemiology, chemistry and ecology. BICI allows for arbitrary compartmental model specification and performs simulation, inference and posterior simulation¹.

Models can contain multiple interacting species, which can each be treated at a population or individual level. BICI can be run entirely using a point-and-click interface. Alternatively, a scripted language, referred to as “BICI-script”, can be used to construct, store, and export complex models and allow for them to be run on HPC (high performance computing).

Bayesian inference is the process of combining data (which can take a variety of forms and may be inherently noisy) with previous knowledge regarding model parameters (the prior) to generate the best available estimates for model parameters and system dynamics (the posterior).

For inference, BICI accepts a variety of individual and/or population-level data, and priors can be specified from a large range of possibilities. Posterior parameter outputs include trace plots, distributions, correlations, and summary statistics (means and 95% credible intervals) as well as diagnostic information (*e.g.*, measuring MCMC convergence and helping to identify potential model misspecification). State outputs include various visualisations for populations, transitions and individuals.

A detailed description of the statistical model underlying BICI, along with the Bayesian inference methodology, is given in an accompanying paper [1]². The focus of this manual is on the practicalities of running the BICI software tool.

1.1 Downloading

BICI is freely available under the GNU General Public License, and can be downloaded from the GitHub repository github.com/theITeam/BICI.

Depending on the user’s platform, the following instructions explain how BICI is run:

- **Windows** – Download the file “BICI_v0.4_windows.zip” and unzip. BICI is run by clicking on the “BICI.exe” icon (if the error message “Windows protected your PC” appears, click on “More info” and “Run”).
- **Linux** – Download the file “BICI_v0.4_linux.tar.gz”. This can be extracted by using the terminal command “tar -zvxf BICI_v0.4_linux.tar.gz”. The code is executed using “./BICI”.
- **Macintosh** – Download the file “BICI_v0.4_mac.zip” and unzip. BICI is run by clicking on the “BICI.app” icon (if the error message “BICI can’t be opened because it is from an unidentified developer...” appears, right click on “BICI.app” and select “Open” to give the option to run).

1.2 The user

Users of BICI will come from diverse backgrounds and a multitude of disciplines. We imagine that this software will be used by different people in different ways. Here are some examples:

¹ A “simulation” samples system dynamics, otherwise referred to as the “state”, from the model given a set of parameters. “Inference” generates estimates for model parameters and the state based on data (including uncertainty in those estimates). “Posterior simulation” simulates from the model using parameters estimated from inference.

² Note, paper [1] refers to software tool SIRE2.0, which was a predecessor of BICI. A paper which describes the additional methodologies used in BICI is currently in progress.

- **The casual user** – They might work entirely within the interface. They could, for example, create and run simple SIR-type models using the point-and-click interface. Inference could be done using time-series population infection data in ‘.csv’ format. BICI could also be useful as a teaching aid to introduce concepts in epidemiological modelling.
- **The serious user** – Complex models can be loaded using input files or through BICI-script. The interface can generate executable files that can be run in parallel on a Linux cluster, and the outputs can then be read back into the interface for visualisation.
- **The pipeline user** – BICI-script can be created by another program, run by the core code and the output files used for some other purpose, *i.e.* a seamless pipeline which doesn’t require the visual interface at all.

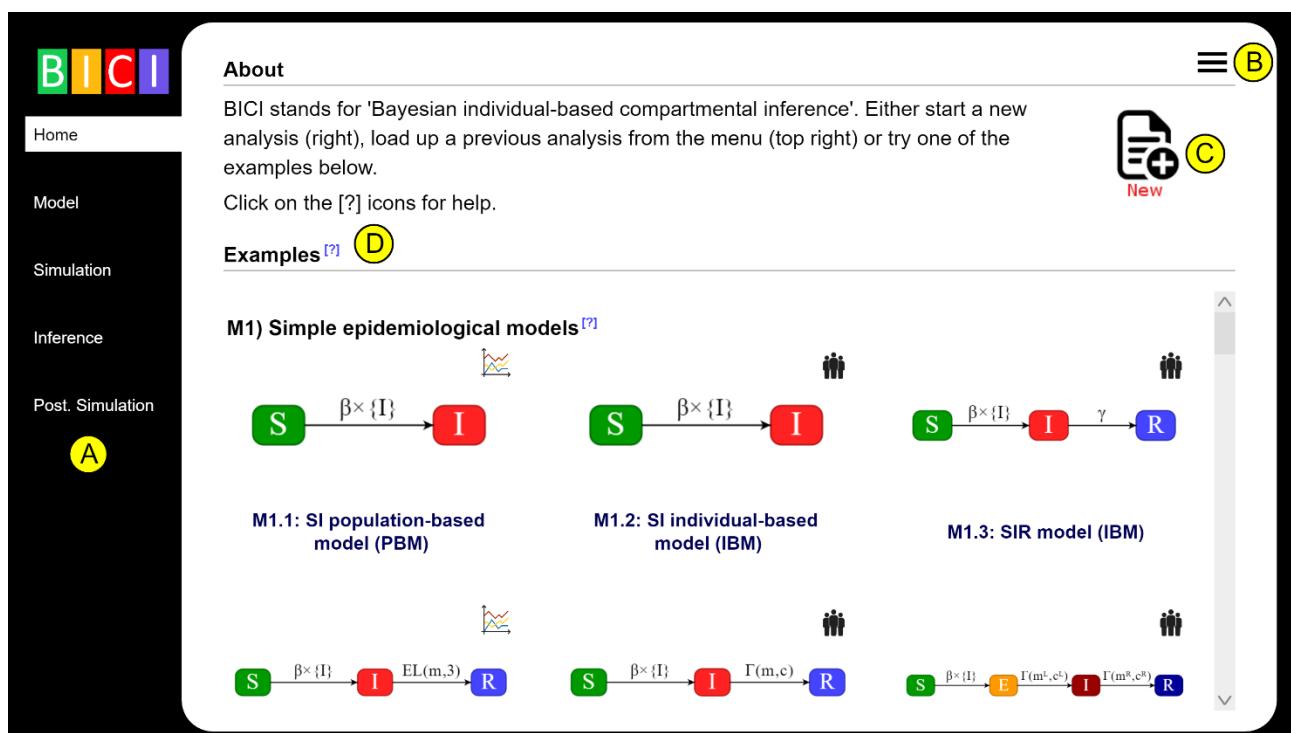


Figure 1 – The home page. A: The main menu (used for navigating between pages), B: The drop-down menu (for loading saving and exporting, see §6.2), C: Start a new model, D: Examples (see §8).

1.3 Getting started

Figure 1 shows the home page displayed when BICI is first loaded. The main menu on the left (Fig. 1A) is used to navigate from page to page. Opening one page may open up further submenus with different options. The structure of this manual broadly follows the tree structure of this menu.

Section 2 describes setting up a compartmental model, §3 deals with simulation from the model, §4 incorporates data to allow for Bayesian inference and, finally, §5 looks at posterior simulation (which can be used for future prediction and counterfactual analysis). Section 6 discusses input and outputs, §7 looks at BICI-script, and, finally, an extensive set of examples is provided in §8.

Three options are available to start using BICI: (1) A previous model/analysis can be loaded from the drop-down menu (Fig. 1B, see §6.2), (2) a new model can be started (Fig. 1C) or (3) one of the example applications can be investigated (Fig. 1D).

1.3.1 Examples

The examples demonstrate a wide variety of different models and data scenarios that illustrate the capabilities of BICI in a variety of different situations (see §8). They are organised into different sections:

- **Sections M1 to M6** – Illustrate different model structures. In each case, initial conditions have been added to the model, so navigating to the “Simulation→Run” page and pressing the “Start” button generates illustrative dynamics.
- **Section A** – Looks at different possibilities when simulating.
- **Sections B to G** – Look at how different data types inform inference. As well as being able to simulate from these models, the user can also go to the “Inference→Run” page and pressing the “Start” button will show illustrative posterior distributions.

The examples can be modified in any way and their default settings will be restored when reloaded from the home page (Fig. 1D). New users are encouraged to spend some minutes exploring the software to get a feel for how BICI’s interface works. Additionally, they can have a go at creating their own model by following the tutorial in the next section.

Whilst using the software, make use of the many [?] help buttons that provide much of the information outlined in this manual.

1.4 Tutorial

In this tutorial the user will learn how to:

- Construct a simple SIR model.
- Simulate from that model.
- Generate a simulated dataset.
- Do inference on that data to generate parameter and state estimates.
- Perform a counterfactual analysis.

1.4.1 Setting up a simple SIR model

Follow these steps:

- A new model can be started by navigating to the “Home” page (on the main menu on the left-hand side) and clicking the “New” button in the top right-hand corner.
- BICI allows for multiple interacting species to be added (for example, predator-prey models, or disease vector models). The example here focuses on just a single species. Let’s add one named “People” and select it to be population-based.
- Next, we add a classification. BICI allows for individuals to have multiple classifications. For example, they may be classified by their disease status, or their age, or their locations. For now, we will set up a single classification called “DS” standing for “disease status” (don’t worry about all the other options).
- The next job is to add the compartments. This can be done by clicking “+ Compartment” on the lower menu bar. Let’s set up three compartments called ‘S’, ‘I’ and ‘R’ and give them different colours. The workspace should look something like this:



- Now let's add some transitions. These allow individuals to change the compartment they are in. Select “ Transition” from the lower menu bar and click first on ‘S’ and then on ‘I’. An arrow going from one to the other should appear. Attached to this is a panel that gives details of the transition. The “distribution” option determines the type of probability distribution that describes the time the individual resides in ‘S’ before moving to ‘I’. The “exponential” option means that there is a certain defined rate with which the transition occurs (*i.e.* a Poisson process).
- Here we set that rate to ‘ $\beta \times \{I\}/N$ ’ (note, Greek characters can be added by typing using Latex notation, *e.g.* ‘\beta’ automatically becomes ‘ β ’, and the ‘*’ character automatically becomes ‘ \times ’). In BICI notation the curly brackets denote “the population of”, so ‘ $\{I\}$ ’ represents the total number of infected individuals.
- Notice that there is no ‘S’ in this expression. This is because BICI uses individual rates of transition, not population-wide rates (as are sometimes shown).
- Add in a recovery rate γ and the model should look something like this:

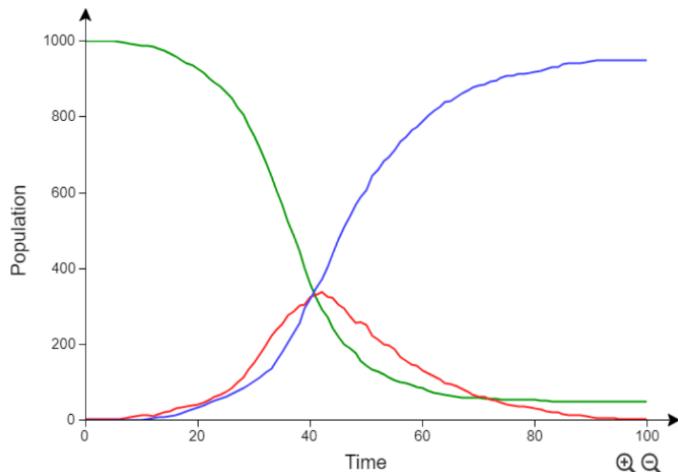


1.4.2 Simulation

- Next, we add some individuals to the model. Click on the “Simulation→Initial Conditions” page and then “ Init. Pop.”. Click “Next” to indicate that we are setting up a fixed initial population with the graphical interface.
- Click on the different compartments to select how many individuals start in each of them. Set them such that S=1000, I=1 and R=0 (*i.e.*, we are going to look at the effect of a single infected individual entering an otherwise susceptible population) and click “Done”.
- On the “Simulation→Parameters” page set N=1001 (*i.e.*, the total population size), $\beta=0.3$ and $\gamma=0.1$ (these correspond to an individual taking on average 10 days to recover from infection and a basic reproduction ratio of $R_0=\beta/\gamma=3$).
- We now set up the simulation on the “Simulation→Run” page. Select a start time of 0, an end time of 100 days and a time-step of 1 (note, longer time-steps speed up simulation/inference but can lead to discretisation errors³).

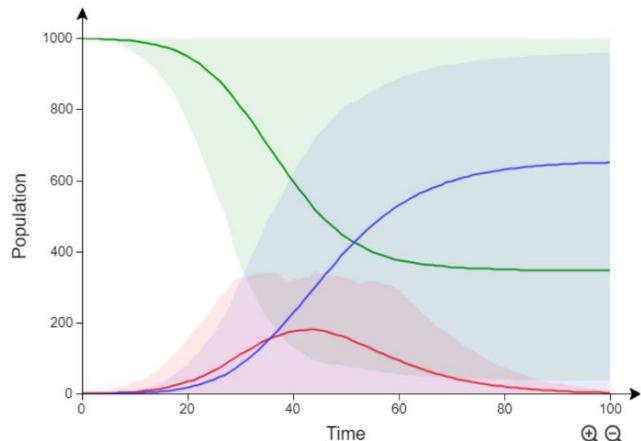
³ As a rough guide, provided the time-step is less than ~20% of the shortest mean transition time, the discretisation error should be relatively minor.

- Click on “Start” to see what happens...



- The output should be something like this, which shows the population variation over time in the three compartments.
- Try looking at different visualisations on the “Simulation→Results” pages. For example, clicking the “View” drop-down menu in the top-right corner, or looking at transitions and parameters on the left menu.

- Go back to “Simulation→Run” and set the simulation number to 100. This produces 100 simulations. The solid lines give the average behaviour, and the shading indicates the region that encompasses 95% of simulations.
- We find huge so-called “stochastic” variability⁴. Sometimes “epidemic extinction” occurs when, despite the fact the basic reproduction number⁵ is greater than one $R_0 > 1$, the epidemic dies out for stochastic reasons.



1.4.3 Generating simulated data

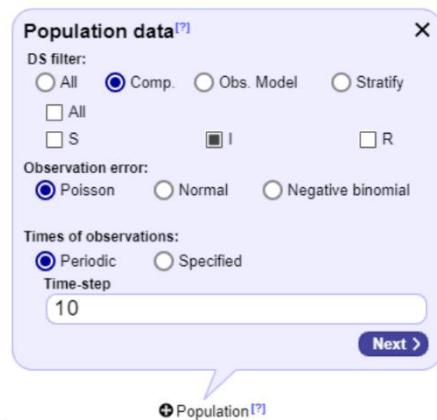
We now generate some hypothetical data from the simulation results in the previous section:

- Go to the “Simulation→Generate Data” page and select “Init. Cond.” from the drop-down menu in the bottom left-hand corner.
- Click on “+ Init. Pop.”. This shows data representing the initial population. Click “Done” twice to add to the list of generated data sources.

⁴ Stochasticity is defined as having a random probability distribution or pattern that may be analysed statistically but not predicted precisely. Epidemics are stochastic because they result from inherently random processes, such as random contacts within a large group of individuals.

⁵ R_0 is defined as the expected number of cases directly generated by one case in a population in which all individuals are susceptible to infection.

- Select “Population” from the drop-down menu in the bottom left-hand corner and click on “ Population”.
- We now tell BICI that we want to generate time series data that estimates the infected population at periodic time intervals. Click on “Comp.” and then select the ‘I’ checkbox (see right). The observation error is set to ‘Poisson’, which means that there is some inherent error in the observed values compared to the real values. We set a time-step of 10 days and click “Next” to generate the data.
- Click “Done” to add this data to our list.
- Finally, click on “Copy” to copy the simulated data into the “Inference→Data” page so we can perform inference on it.



1.4.4 Inference

- The data sources should look like this:

Type	Details	Number	Spec.	Table
Init. Pop.	-	1001	View	Edit
Population	I	10	View	Edit

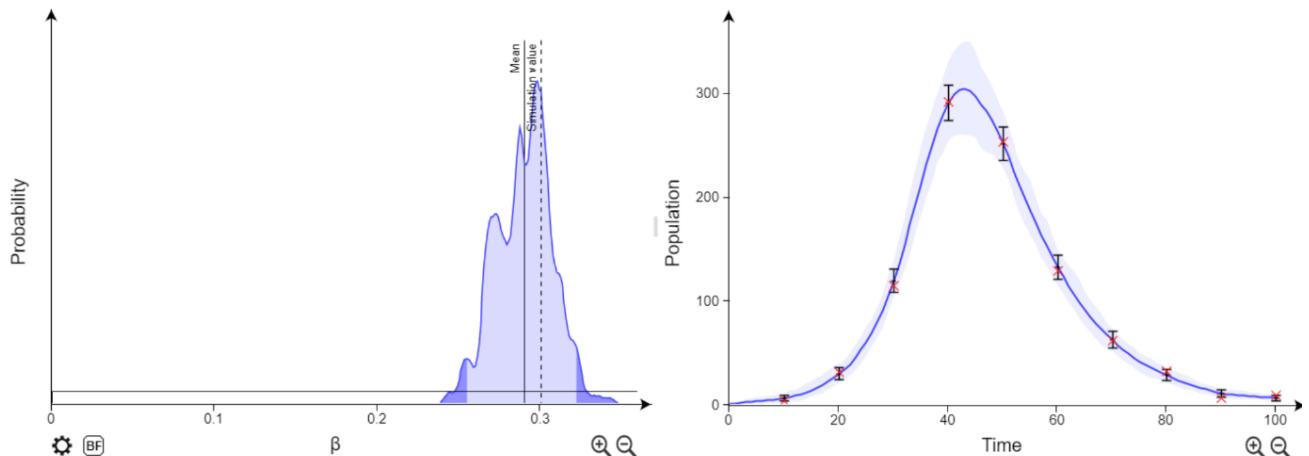
- Now the data has been added, let’s consider the prior. Go to the “Inference→Prior” page.
- The prior captures previous knowledge about the parameter values before the data is considered. For example, if we are very certain about a quantity, we could fix its value. If, on the other hand, we are very uncertain, we could use a wide distribution to represent a plausible range of values.
- Supposing we don’t know much about the disease, the prior is chosen to be wide and relatively uninformative: a uniform prior between zero and one for β and γ .
- N , the population number, is fixed to 1001.
- We start inference by going to the “Inference→Run” page and ensure a start time of 0, an end time of 100 and a timestep 1.
- BICI gives some different possibilities for the algorithm used to perform inference. Here we select the “DA-MCMC” option.
- The number of updates informs how long the algorithm takes to run. Running for longer may produce a better approximation to the posterior, but it can also be computationally wasteful. *A priori* it is difficult to know what this number should be, but the default value of 5000 is usually a good first guess. MCMC diagnostics can be used to determine if inference needs to be rerun using more updates (see §4.4.7 for details).
- MCMC “chains” run in parallel, so helping to speed up computation. Each chain runs on a separate CPU core. Since modern computers typically have 4-8 cores, so the default number of chains is set to 3 (to avoid the computer slowing down too much). An additional advantage of running parallel chains is that it tells us something about the reliability of the results (see §4.4.7)⁶.
- Click on the “Start” button to begin inference.
- This example should take just a few seconds to run...

Parameters [?]

$$\begin{aligned} N &\sim \text{fix}(1001) \\ \beta &\sim \text{uniform}(0,1) \\ \gamma &\sim \text{uniform}(0,1) \end{aligned}$$

⁶ If all chains converge on essentially the same solution, it gives us confident that the solution is correct. However, if different chains end in different solutions, it is indicative of either multimodality or chains becoming stuck in metastable state. In the latter case, strategies such as annealing or particle annealed sampling (PAS) may help (see §4.3.1).

- A variety of different visualisations of the posterior can be found on the “Inference→Results” page (see Appendices C-F).
- Here are just some results that show: (1) the posterior distribution for the parameter β on the left (note, this distribution contains the value used to simulate the data, denoted by the vertical dashed line) and (2) the infected population as a function of time on the right (where the red crosses represent the observed data and the error bars provide uncertainty in those measurements):



1.4.5 Posterior simulation

Posterior simulation can be used for future prediction and counterfactual analysis. Here we consider the effect of an intervention to reduce disease transmission mid-way through the epidemic.

- Go to the “Post. Simulation→Parameter Mult.” page and click on “ \oplus Multiplier”. Select the parameter β .
- “Multipliers” act by multiplying the corresponding parameter by a potentially time-varying factor.

Set the parameter multiplier as follows:

Parameter factor
Knots = <code>start,35,36,end</code>
$f \sim \beta(t) = [1,1,0.3,0.3]$
View X

- This defines a spline that starts with a constant value of one until day $t_{\text{inter}}=35$, at which point it sharply falls to 0.3 for the remaining time (this can be visualised by clicking on the “View” button). This is equivalent to a disease intervention introduced at t_{inter} that reduces the transmission rate β to 30% its previous value (for example, this could be due to the introduction of mask wearing during the COVID-19 epidemic).

- Go to the “Post. Simulation→Run” page and set the start time to 35 and the end time to 100. This corresponds to the epidemic progressing as before until t_{inter} and subsequently predicts the effect of the intervention.
- Click “Start”.
- The graph on the right shows that the intervention drastically reduces the impact of the epidemic.

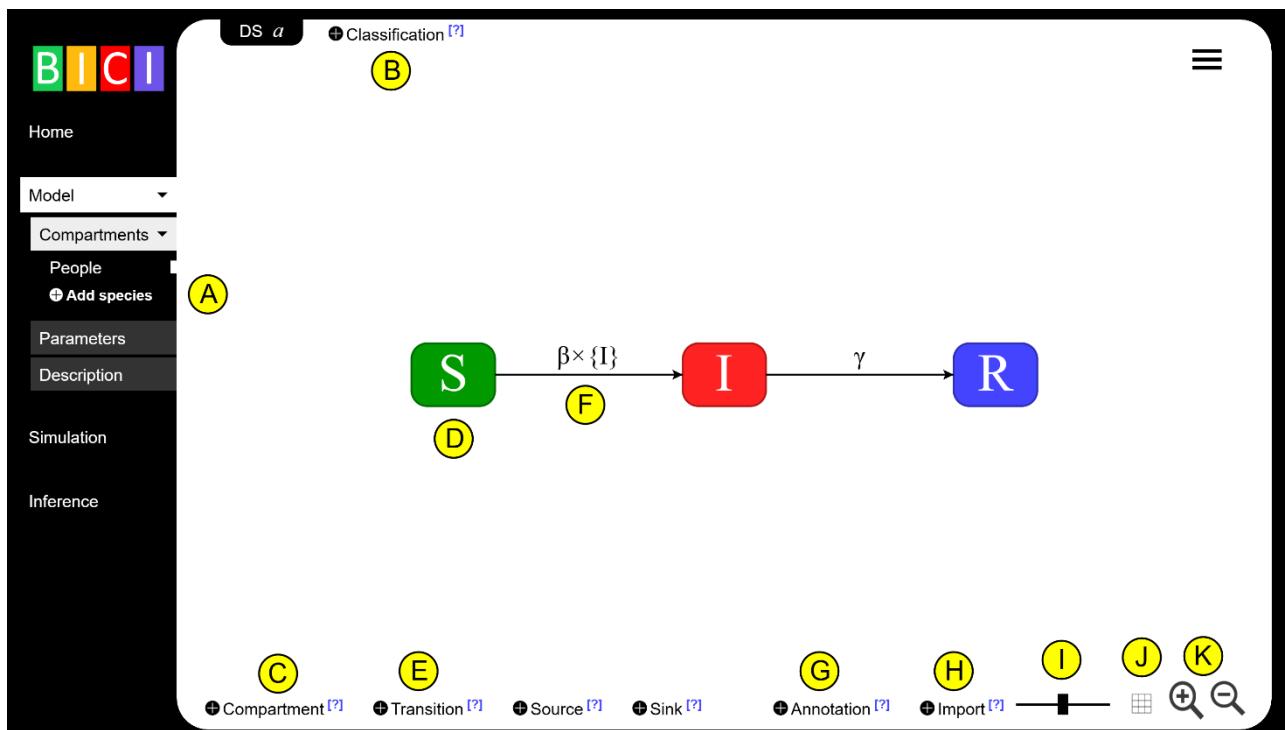
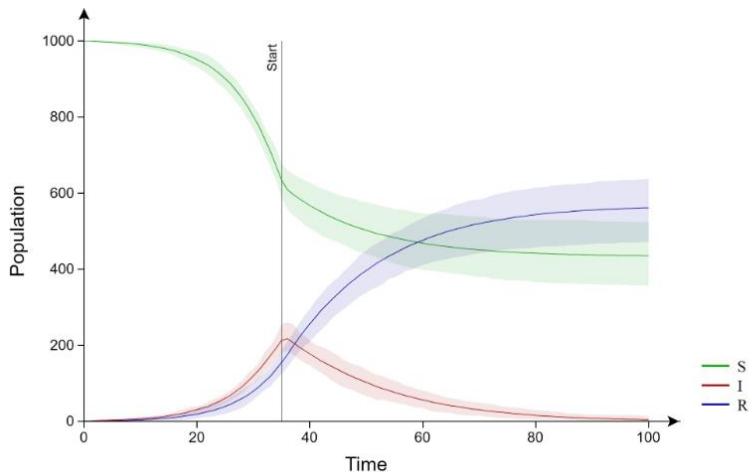


Figure 2 – The compartmental model. This shows an SIR compartmental model (the compartments represent ‘S’ susceptible, ‘I’ infected and ‘R’ recovered from disease). A: Add a species, B: Add a classification, C: Add a compartment, D: A compartment, E: Add a transition, F: A transition, G: Add an annotation, H: Import compartments and transitions, I: Slider to adjust compartment size, J: Switch grid view on or off, K: Zoom in or out (double clicking on the workspace can also zoom in).

2) Model

This section describes model specification, which encompasses both compartmental structure and parameter definitions.

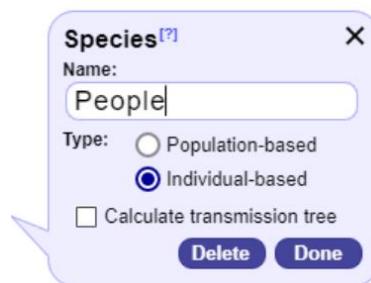
2.1 Compartmental model

The “Model→Compartments” page (Fig. 2) shows how a compartmental model is defined. Three layers of division are used to differentiate individuals within the system: species, classification and compartment. These are each described below:

2.1.1 Species

Species group together individuals of the same type. In many cases only one species will be present, *e.g.* people. In other cases, however, different species can interact, *e.g.* predator-prey models, disease models with vectors, or models that account for the accumulation of pathogen in the environment (here the pathogen, itself, becomes a species).

A species can be added by clicking on “⊕ Add species” (Fig. 2A). See §2.1.11 for naming requirements. Two types of species can be created:



- **Population-based** – The total numbers of individuals in different compartments are tracked as a function of time. Only population-level data can be used and transitions are restricted to have either exponential or Erlang distributed waiting times.
- **Individual-based** – Here each individual in the system has its own timeline. This allows for individual-level data (as well as population-level data) to be utilised in inference, and enables the greatest level of flexibility in defining transition distributions. Furthermore, the dynamics of individuals can vary (beyond their compartmental classification) thanks to so-called “individual” or “fixed” effects. Individual-based models, however, can become computationally slow when they contain many individuals and also memory intensive.

Transmission tree – If the model is individual-based, a further checkbox allows the user to select if the transmission tree is calculated (this keeps track of who is infecting whom). This is of primary use in cases in which genetic information on the pathogen is available (otherwise it is generally turned off to help speed up computations). If selected, it is necessary to choose which classification relates to the infection process and then go on to select “infected” compartments in that classification (see §2.1.3).

2.1.2 Classifications

A classification is a discrete means of differentiating individuals. Examples of typical classifications include disease status, location or sex.

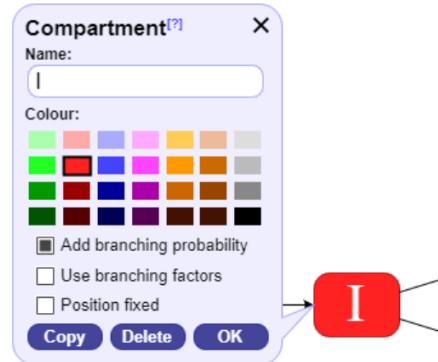
A classification can be added by clicking “⊕ Classification” (Fig. 2B) at the top of the page. It is usually given a short but meaningful name (such as ‘DS’ for “disease status”) and an index. This index is a single letter that can be used in mathematical equations to represent all the possible values the classification can take.

A coordinate system is associated with the classification. Cartesian coordinates are used to represent the usual block style of compartmental models, as shown in Fig. 2 (*e.g.* the SIR model). In spatial models, an alternative is to use geographical coordinates (that use longitude and latitude information). These can represent a map-based distribution of points (*e.g.* farms) or regions (*e.g.* local authorities).

Cloning – “Cloning” a classification is used to copy compartments from one species to another (once cloned, the two classifications must always share the same compartments, but may have different transitions). Cloning is useful when two or more species share the same classification (*e.g.* in disease transmission experiments the individuals and virus share the same set of closed contact groups).

2.1.3 Compartments

Compartments are possible states a classification can take. They are added by clicking “ Compartment” (Fig. 2C), which allows the user to place them onto the workspace (Fig. 2D). A compartment name is usually chosen to be short (conventionally a single uppercase letter), *e.g.* ‘S’ to represent susceptible individuals. The colour of the compartment can be selected (by convention this is green for susceptible ‘S’ and red for infected ‘I’). These colours are used for plotting population graphs when results are generated. The position of the compartment can be fixed by selecting the checkbox, otherwise it can be dragged around the screen with the cursor.



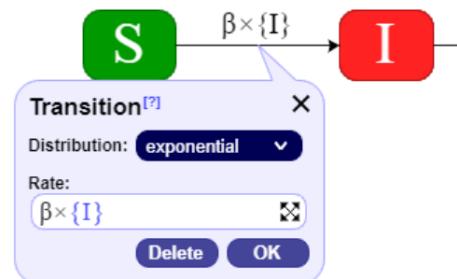
When multiple exponential transitions leave a compartment, the “Add branching probability” checkbox allows the user to select whether or not branching probabilities are implemented. If branching does occur, the “Use branching factors” checkbox specifies whether or not so-called “branching factors” are used (see §2.1.6).

In cases in which the transmission tree is turned on (see §2.1.1), a further checkbox allows the user to select if the compartment corresponds to an “infected” state.

See §2.1.9 for importing compartments into the model from a file (useful if there are many of them).

2.1.4 Transitions

Transitions allow for individuals to move from one compartment (initial) to another (final). Clicking “ Transition” (Fig. 2E) and then selecting the initial and final compartments adds a transition to the model. For example, when an individual becomes infected it moves from an initial susceptible ‘S’ compartment to a final infected ‘I’ compartment (Fig. 2F).



Note, intermediary points can also be added to map out a path instead of just a line, *e.g.* this could be used to add a non-straight arrow going from ‘R’ back to ‘S’ again in Fig. 2 (to incorporate waning immunity).

Different possible probability distributions can be selected for the transition time:

- **exponential** – The exponential distribution is specified by a rate. Such transitions are known as “Markovian”, because they occur independently of when the individual entered the initial compartment. Such an approach is valid for random processes, for example contacts between different members of a population leading to the spread of infection.

- **gamma** – The gamma distribution is specified by a mean and coefficient of variation⁷. An example application of the gamma distribution is to model the incubation period (time duration between becoming infected and then becoming infectious) as used in an SEIR model. In reality such a transition is expected to be non-Markovian, because the time an individual becomes infectious (after becoming infected) depends on complex processes underlying immune system dynamics.
- **erlang** – The Erlang distribution is specified by a mean and shape (where the shape must take a positive fixed integer value). This is a special case of the gamma distribution. This type of distribution can usefully be employed in population-based models (which do not allow for gamma distributions) as a way to incorporate non-Markovian transitions.
- **log-normal** – The log-normal distribution is specified by a mean and coefficient of variation.
- **weibull** – The Weibull distribution is specified by a scale and shape. It is somewhat similar to the gamma distribution, but is used in preference under certain circumstances.
- **period** – Here the event happens after a certain prescribed time period. For example, this could be used to represent the predictable increase in age of an individual. The period must be specified as a fixed positive number (and so cannot be estimated under inference).

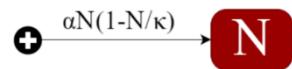
The defining quantities (mean, shape etc...) for these distributions are set via user-defined equations (see §2.1.10). Note, if a species is set to “population-based”, only the “exponential” or “erlang” distributions can be selected from.

See §2.1.9 for importing transitions into the model from a file (useful if there are many of them)

2.1.5 Sources and sinks

A source allows for individuals to enter the system as part of the process (e.g. as a result of births)⁸. The probability per unit time (or rate) at which this occurs is determined by a user-defined equation that can contain model parameters, populations and be dependent on other classifications. Unlike other transitions (which define individual-based transition probabilities), the equation for a source provides the overall rate at which individuals enter the system. Note, if the equation is stratified by a classification, the rates entering each subpopulation are specified separately⁹.

A sink allows for individuals to leave the system as part of the process (e.g. as a result of death)¹⁰. Sinks have the same options as for other transitions.



⁷ The coefficient of variation is equal to the standard deviation divided by the mean.

⁸ This is in contrast to initial conditions, which place individuals in the system as an externally enforced constraint.

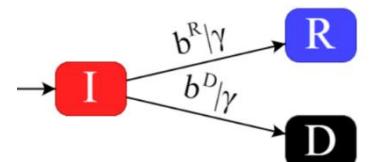
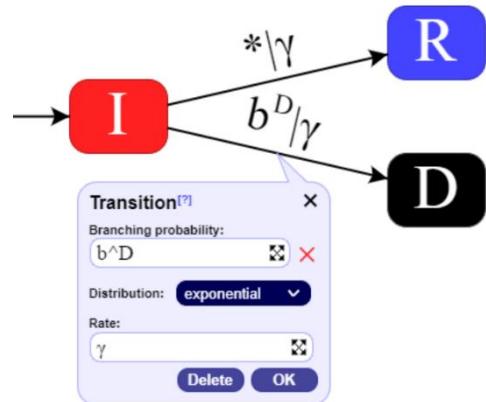
⁹ To give an example, suppose a model has a source entering the S compartment and a further classification “Sex”. If the source rate is defined as “ α ”, this means α individuals enter the system per unit time on average, with 50% male and 50% female, on average. If the source rate is defined as “ α_s ”, then a rate of α_M enter as male and a rate of α_F enter as female, i.e. the different possibilities add up to give the total rate.

¹⁰ This is in contrast to “Remove Ind” data, which can externally specify when individuals are removed from the system.

2.1.6 Branching

In cases in which multiple transitions leave an initial compartment, there are three possibilities:

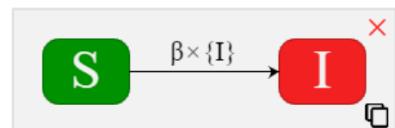
- **No branching probabilities** – In this case, all the transitions must be exponentially distributed with defined rates (note, even here branching probabilities can explicitly be added by clicking on the “Add branching probability” checkbox when the initial compartment is selected).
- **Branching probabilities** – An equation on the transition is used to determine the probability an individual selects a particular branch when they enter the initial compartment. Note, one of the transitions has no probability associated with it because, by definition, the probabilities must add up to one (this is denoted by “*” in the equation definition).
- **Branching factors** – This option is turned on by clicking on the initial compartment and selecting the “Use branching factors” checkbox. The branching factor equation is defined down each transition to denote the *relative* probability of going down that branch. Consequently, the branching probability is given by the branching factor divided by the sum of all branching factors leaving the initial compartment. In the example on the right, the probability of dying is given by $b^D/(b^D + b^R)$. Such a definition is useful because, unlike branching probabilities, branching factors are defined for all positive numbers making them suitable to incorporate fixed and/or individual effects.



2.1.7 Building up a compartmental model

Numerous compartments and transitions can be added to a classification. Compartments can be dragged across the workspace to move their location, or edited to change their colour or name.

Multiple compartments can be selected by dragging a selection box with the right mouse button pressed down. Selected compartments can be moved, copied or deleted.



The size of compartments on the screen can be adjusted using the slider at Fig. 2I (this can add clarity by avoiding compartments from overlapping each other).

A spatial grid guide can be added by clicking on Fig. 2J. This can be of use if the distance between compartments is used in the transition equations, e.g. in a spatial disease transmission kernel (see §2.1.10 for how distances are added to equations).

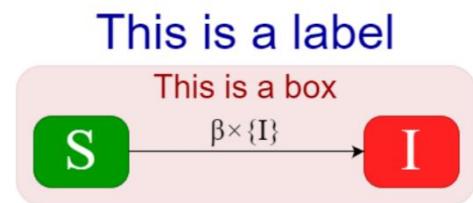
For large compartmental models (e.g. spatial models with many farms), it may be necessary to zoom in and out of the workspace. This can be done by clicking on the zoom buttons (Fig. 2K). Zooming in can, alternatively, be done by double-clicking on the workspace.

2.1.8 Annotations

Although annotations play no part in how the model works, they can be used to clarify the meaning behind different elements of the model. They can be added by clicking “⊕ Annotation” (Fig. 2G).

Different types of annotations can be added:

- **Label** — These are user-defined text that can be positioned anywhere on the workspace.
- **Bounding box** — These are labelled boxes which bound selected compartments.
- **Map** — A map can be loaded in GeoJSON format (note, this is only available when geographical coordinates are used), e.g. maps could be added as a visual reference for farm locations.



2.1.9 Importing

Creating large numbers of compartments and transitions would become laborious using the point-and-click interface. For this reason, BICI allows them to be imported using “ Import” (Fig. 2H).

The following import options are available:

- **Compartments** – Loads multiple compartments from a file.
REQUIRES – A data table containing the column “Name”, which gives the compartment names to be added. Optionally, position data can be also included by columns “x” and “y” giving the *x* and *y* position in the workspace (in Cartesian coordinates) or “Lat” latitude and “Long” longitude (in geographical coordinates). If positions are not specified, they are automatically set. Also a column “Color” can be used to allocate colours to compartments (either in hexadecimal format, e.g. “#ff0000”, or RGB format, e.g. “rgb(255,0,0)”).
- **Comp. Map** – Loads multiple compartments represented by geographical boundary data.
REQUIRES – A GeoJSON file from which regions can be selected.
- **Transitions** – Loads multiple transitions from a file.
REQUIRES – A data table containing at least the following columns: “From” gives the compartment from which the transition originates (‘+’ is used to represent a source), “To” gives the compartment to which the transition ends up (‘-’ is used to represent a sink) and “Value” specifies the transition. The format for this value depends on the transition distribution:
 - *exp(rate:[eq])* – An exponential distribution, where “[eq]” is replaced with an equation that determines the rate at which transitions occurs, e.g. ‘r’ or ‘ $\beta \times \{I\}$ ’.
 - *gamma(mean:[eq1],cv:[eq2])* – A gamma distribution with equations for mean and coefficient of variation.
 - *erlang(mean:[eq],shape:[integer])* – An Erlang distribution with equation for mean and a fixed positive integer shape.
 - *log-normal(mean:[eq1],cv:[eq2])* – A log-normal distribution with equations for mean and coefficient of variation.
 - *weibull(scale:[eq1],shape:[eq2])* – A Weibull distribution with equations for scale and shape.
 - *period(time:[eq])* – For when the event happens after a certain prescribed time period.

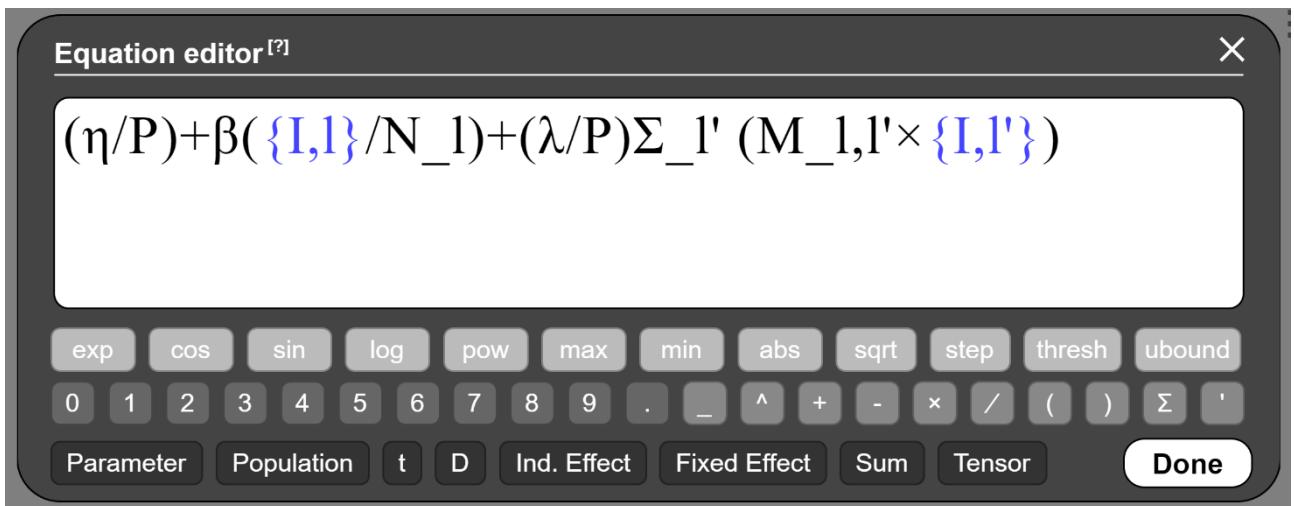


Figure 3 – The equation editor. Different options in this editor are described in §2.1.10.

2.1.10 Equations

Equations are used to represent many different quantities in BICI. Examples include the rate at which individuals pass down a transition, coefficients of variation for distributions, branching probabilities, and parameters related to the observation model.

A textbox is provided to enter equations, and for simple expressions this is more than adequate.

Sometimes, however, equations become complicated, and for this reason an editor is incorporated into the software. This can be opened by clicking on the ‘’ icon.

Equations are potentially made up of the following elements:

- **Numbers** – Fixed numerical quantities.
- **Parameters** – These are usually represented by Greek letters (e.g. β) or a single letter (e.g. a), but can, in principle, be any text (see below for exceptions). Greek can be typed in Latex format which automatically gets converted to the equivalent character (e.g. ‘\alpha’ changes to ‘ α ’).

Parameters can depend upon classification indices by using an underscore (which makes a subscript when formatted). For example, ‘ β_a ’ becomes “ β_a ” when formatted, which means that β is a vector that depends on a compartment indexed by a (where a could, e.g., represent different locations).

Superscripts can also be added, e.g. ‘ β^{Label} ’ gets converted to β^{Label} . Note, superscript do not imply the parameter is being raised to a given power¹¹. They are, however, a useful way to differentiate parameters, e.g. such that β^A is treated differently to β^B . Indices and superscripts can both be used, but the superscript must always go first, e.g. ‘ β^{Label}_a ’ is fine but ‘ β_a^{Label} ’ is incorrect.

The values for parameters must be set under model simulation, but can be estimated from data under inference (in which case they require a prior definition, see §4.2).

- **Populations** – These are denoted by curly brackets and represent the number of individuals within a sub-population (e.g. ‘{I}’ may represent the total number of infected individuals or ‘{S,M}’ may represent the number of susceptible males). Indexes can also be used in populations, e.g. “{I,a}” could represent the local infected population in location indexed by a .

¹¹ This is implemented through the `pow(a|b)` function.

- **Time** – The quantity ‘t’ is reserved to represent time. This can either be explicitly used within an equation (e.g. ‘ $a+b\cos(\text{ctx})$ ’, which is a trigonometric function involving time t), or to indicate a time-varying parameter represented by a spline (e.g. $\eta(t)$).
- **Distance** – The parameter ‘D’ is reserved to represent distance. ‘ $D_{a,a'}$ ’ when formatted becomes $D_{a,a'}$, which is the distance between compartments a and a' . In geographical coordinates this is given in kms and in Cartesian coordinates it represents the Euclidean distance between the coordinates of the two compartments.
- **Operators** – Standard operators are used to combine together quantities: ‘*’ for multiply (which gets converted to ‘ \times ’), ‘/’ for divide, ‘+’ for addition and ‘-’ for subtraction.
- **Sums** – These are used to add up over compartments within a classification. For this purpose, a “primed” index is used. For example the text input “ $\backslash\text{sum}_a' (M_{a,a'} * \{I, a'\})$ ” is interpreted as the mathematical expression $\sum_{a'} (M_{a,a'} \times \{I, a'\})$. Here the matrix $M_{a,a'}$ mediates interactions between different areas and the primed location index a' is summed over.

Sums have the format “ $\backslash\text{sum}_\text{index1}',\text{index2}' (\dots)$ ”, i.e. they can be over multiple indices, but the content must always be encapsulated by round brackets. If the indices are the same, double primes can be used to differentiate, e.g. “ $\backslash\text{sum}_\text{index}',\text{index}'' (\dots)$ ”.

It is possible to restrict sums such that they only act over a specified distance (e.g., this can be used to speed up computation on spatial models, because interactions over a long distance may be negligible). For example, “ $\backslash\text{sum}^\text{max:50}_a' (M_{a,a'} * \{I, a'\})$ ”, which gets converted to $\sum_{a'}^{\text{max:50}} M_{a,a'} \times \{I, a'\}$, only sums over terms where location a' is less than 50km from location a . This example assumes geographical coordinates. In Cartesian coordinates “max” restricts the Euclidean distance between compartments.

- **Individual effects** – These are denoted by square brackets (e.g. $[g]$) and allow individual-based variation. Individual effects are log-normally distributed with a mean of one and a variance matrix that can be set or estimated. Additionally, they can be correlated between individuals (through some loadable \mathbf{A} matrix), which allows for quantitative genetics models to be incorporated, or correlated with other individual effects. See §2.2.2 for details.
- **Fixed effects** – These are denoted by triangular brackets (e.g. $\langle w \rangle$) and allow for an individual property (e.g. weight) to be related to model transitions (e.g. it could be that lighter individuals become infected more quickly). To implement a fixed effect, the covariate vector \mathbf{X} , which gives the property values for each individual in the system, must be loaded from a file. Fixed effects have a population mean of one and are strictly positive. A parameter (ν with superscript given by the fixed effect name, so ν^w in this case) controls the strength of the fixed effect, which is either set or inferred. See §2.2.3 for details.
- **Functions** – A selection of different functions can be added into equations:
 - $\exp(x)$ – The exponential function.
 - $\log(x)$ – The logarithm function.
 - $\sin(x)$ – The sine function.
 - $\cos(x)$ – The cosine function.
 - $\text{step}(x)$ – The step function (if $x > 0$ returns 1, else 0).
 - $\text{pow}(x|y)$ – The power function x^y .
 - $\text{max}(x|y)$ – The maximum function (if $x > y$ returns x , else returns y).
 - $\text{min}(x|y)$ – The minimum function (if $x < y$ returns x , else returns y).
 - $\text{abs}(x)$ – The absolute function (if $x > 0$ returns x , else returns $-x$).

- $\text{sqrt}(x)$ – The square root function.
- $\text{thresh}(x|y)$ – The threshold function (if $x < y$ returns 0, else returns x).
- $\text{ubound}(x,y)$ – An upper bounded function (if $x > y$ returns ∞ , else returns x).

2.1.11 Naming requirements

The following characters are not allowed in names or equations: ““£!\$%&;@~#\?→”. Additionally, names for species, classifications and compartments (as well as fixed and individual effects) cannot contain: “| *x_{}<>⟨⟩[]()=Σ””. Finally, parameters are further required to not contain: “+/-0123456789.” (to avoid confusion with arithmetic operators and numbers).

The screenshot shows the BiCI software interface with the title "Model parameters" at the top. On the left, there is a sidebar with sections: Home, Model (selected), Compartments, Parameters, Description, Simulation, and Inference. The main area displays a list of parameters categorized into several groups:

- A: Parameters**: cv, m, P, α, β, γ, Δ, λ
- B: Individual effects**: $\eta(t)$ ~ Shifted LN($\Omega^{g,g} \otimes A^g$) (with a "Load A^g matrix" button)
- C: Fixed effects**: $\langle w \rangle \propto \exp(X_i^w \times v^w)$ (with a "Load X^w vector" button)
- D: Splines**: $\eta(t)$ Knots = start,20,40,60,80,end. Options: Log-Normal (selected), Normal, Value = 0.5 (with a "Smooth" button)
- E: Constants**: P = 68000000, D_{l,r} = Distance Matrix(100,100) (with a "View" button)
- F: Reparameterised**: M_{l,r} = 1/(1+pow(D_{l,r}/Δ|α)) (with a red "X" icon)
- G: Distributions**: N_i ~ log-normal(m,cv) (with a "Split" button and a red "X" icon)
- H: Derived**: κ(t) = η(t)/λ (with a red "X" icon)
- I: Factors**: f_v Weight = [3,2] (with a "Weight" button and a red "X" icon)

At the bottom, there are buttons for Constant, Reparam., Distribution, Derived, and Factor.

Figure 4 – Model parameters. This shows how different parameters are treated in the model. A: All model parameters, separated into categories. B: Individual effects, C: Fixed effects, D: Splines, E: Constants, F: Reparameterisations, G: Distributions, H: Derived, I: Factors.

2.2 Parameters

The “Model→Parameters” page in Fig. 4 summarises how parameters are treated in the model. The following sections describe different features of this page:

2.2.1 Parameter summary

Figure 4A shows a list of all model parameters, ordered by category. First there are univariate parameters, followed by vectors and matrices (and more generally tensors). Following this are parameters associated with variances and correlations for any individual effects in the model. Finally, parameters for fixed effects are shown. Clicking on these buttons gives a brief description of the parameter and allows for the user to jump to the part of the model in which the parameter appears.

2.2.2 Individual effects

In Fig. 4B we see that an individual effect $[g]$ has been added to the model. This represents a vector with a value for each individual in the system. Individual effects are assumed to be sampled from the following shifted log-normal distribution:

$$[g] = \text{Shifted LN}(\Omega^{g,g} \otimes \mathbf{A}) \quad (1)$$

The “shifted” notation corresponds to the mean on the log-scale deviating from zero to ensure that the population average of the individual effect is exactly one (this makes sense, because individual effects are used to investigate variation across individuals, not changes in this mean).

The covariance in Eq.(1) comes from the tensor product of two parts:

- **Individual effect covariance** – This sets the overall variance in the individual effect across the population (on the log scale)¹². In this particular case it is a scalar $\Omega^{g,g}$, but when more than one individual effect is included, it becomes a symmetric matrix¹³. This is parameterised through its diagonals and correlation coefficients¹⁴.
- **Relationship matrix A** – Captures correlations in individual effects between individuals. These are optional, and included by selecting the “Ind. Cor.” checkbox (Fig. 4B) (if uncheck, the \mathbf{A} matrix is simply set to the identity matrix).
The \mathbf{A} matrix is commonly used in quantitative genetic analysis (such that related individuals are more likely to share similar values for their individual effect¹⁵) and can be derived either from pedigree information or genetic data. Typically, this matrix will be 1 along its diagonal, 0.5 for a parent/sibling relationship, 0.5 for full-sibs, 0.25 for half-sibs, and become increasingly small for more and more distant relatives. If selected, it is necessary to load up an \mathbf{A} matrix:

REQUIRES – Information can be loaded in one of two ways: (1) The full matrix can be loaded by providing a data table with headings given by the individual IDs and a square matrix of elements that defines the

¹² If this value is 0.4, it would imply a standard deviation of 0.2, which would set ~20% variation in the individual effect around a mean of one.

¹³ E.g., if we imagine two individual effects $[g]$ and $[f]$, a 2×2 covariance matrix $\Omega^{f,g}$ would be defined.

¹⁴ E.g., for two individual effect $[g]$ and $[f]$ we get diagonal variance parameters $\Omega^{g,g}$ and $\Omega^{f,f}$ and an off-diagonal correlation parameter $\omega^{g,f}$.

¹⁵ E.g., if a parent is more susceptible than average, it might be likely that its offspring would also be more susceptible.

relationships between individuals, or (2) a pedigree can be loaded by a data table with headings “ID”, “sire” and “dam”, which gives the ID of each individual in the population along with its parents.

Note, A can also include IDs of individuals who do not appear in the compartmental model itself, and individual effect estimates can be made for these individuals.

2.2.3 Fixed effects

Fixed effects allow for individual quantities to be incorporated into the model and act as covariates that can modify transition rates, means etc... Fixed effects are indicated in equations by triangular brackets, e.g. $\langle \mathbf{w} \rangle$. This represents a vector with a value for each individual in the system calculated according to:

$$\langle \mathbf{w} \rangle = ce^{\mathbf{X}^w \nu^w} \quad (2)$$

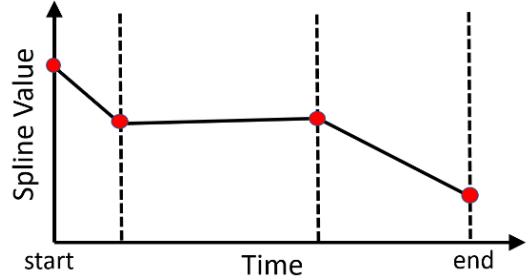
The constant of proportionality c is chosen such that $\langle \mathbf{w} \rangle$ has a population-wide average of one. The vector \mathbf{X}^w stores individual information (e.g. weight), which acts as a covariate. The strength of the interaction is governed by fixed effect parameter ν^w . If this is zero, it means that $\langle \mathbf{w} \rangle$ is one across all individuals. If it is non-zero, it means that $\langle \mathbf{w} \rangle$ varies across individuals and so relates whatever equation $\langle \mathbf{w} \rangle$ is in with the covariate.

Incorporating a fixed effect means that we must load the covariate vector \mathbf{X}^w . This is done by clicking on the “Load \mathbf{X}^w vector” in Fig. 4C.

REQUIRES – A data table containing at least the following columns: “ID” a unique identifier for individuals and “Value” stores the individual covariate.

2.2.4 Splines

Splines are used to account for time variation in model parameters (this is indicated by placing (t) after the parameter definition) (Fig. 4D). Splines are assumed to be piecewise linear, which means the parameter value changes linearly between specified time points, otherwise known as “knots”. The diagram on the right shows four knots, with times indicated by the vertical dashed lines (one at the start and end, with two intermediate points). The knot times can be arbitrarily set, although they must be monotonically increasing (sharp increases or decreases in spline value can be accommodated by having a very short time period between successive knots, e.g. a single time-step). In this example, the spline is defined by a parameter vector with four elements that determine the values for the red circles. Values at other times are linearly interpolated (black lines).



Clicking on the “Knots” value (Fig. 4D) opens a text box used to set the knot times (comma separated). The special key words ‘start’ and ‘end’ are used to represent the overall time range for simulation or inference. Intermediate points can be added, e.g. in the example in Fig. 4D the knots are set to ‘start,20,40,60,80,end’ which means that values for the parameter are defined at the start, at times 20, 40, 60, 80 and then at the end.

Optionally, smoothing can be added to splines. This is used to represent the fact that parameter values might not be expected to jump drastically between consecutive knots, but rather undergo some smooth change. Smoothing can be applied in two possible ways:

- **Normal** – The expected distribution of the parameter value at one knot on the spline is expected to be normally distributed with respect to the value at the previous knot. This approach is appropriate when the parameter can become *negative as well as positive*.
- **log-normal** – The expected distribution in the log of the parameter value at one knot on the spline is expected to be normally distributed with respect to the log of the value at the previous knot. This is appropriate when the parameter is *strictly positive*.

In both cases, the “Value” can be set, which specifies the standard deviation in the normal (a smaller value implies more stringent smoothing).

2.2.5 Constants

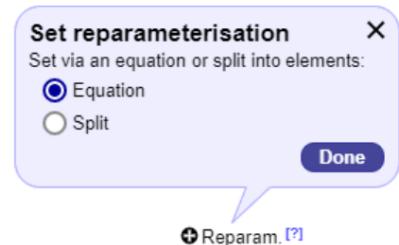
Parameters can be set as constant by clicking “ Constant”. A constant takes that value regardless of whether simulation or inference is performed. If a parameter is only set for the purposes of simulation, this should be done under the “Simulation→Parameter” page. In the example in Fig. 4E, the parameter P (which represents a population) is set to a fixed value. The distance matrix $D_{l,l'}$ is, by definition, constant.

As additional benefit of setting parameter vectors and tensors to be constant is that their values are not output from the model when it is run, which means reduced file sizes and memory requirements when using the interface.

2.2.6 Reparameterisations

Reparameterisation expresses one parameter (vector/matrix/tensor) in terms other parameters. It is typically used as a way of reducing the overall number of parameters to be estimated¹⁶.

Setting a parameter to be reparameterised can be done by clicking “ Reparam”. Two options are available:



- **Equation** – Here the parameter is written in terms of an equation. In the example in Fig. 4F, the contact matrix across locations is reparameterised in terms of a power-law spatial kernel. If the parameter contains any dependencies, these can also be reflected in the equation (in the example the indices l and l' appear in both the left and right-hand-sides of the equation).
- **Split** – In this case, each element of the parameter is defined by a separate univariate equation. Such equations can be loaded from a file:
REQUIRES – A data table containing columns with headings given by the indices of the tensor and a final “Equation” column (see §2.1.10 for details on formatting).

2.2.7 Distributions

Setting elements of a parameter vector/matrix/tensor to be sampled from a distribution is similar to specifying a prior (and uses the same set of possibilities). Here, however, new parameters are introduced that characterise the distribution (*e.g.* the mean, standard deviation etc...). These go on to have their own prior, incorporating a so-called “hierarchical” model.

¹⁶ An example of this is when a contact matrix $M_{l,l'}$ of interactions, which contains L^2 elements where there are L geographical locations, is reparameterised as a power-law spatial kernel, which only contains three parameters.

Distributions for parameters can be set by clicking “ Distribution”. The example in Fig. 4G shows that parameter vector N_l is sampled from a log-normal distribution with mean m and coefficient of variation cv .

A parameter could have one sampling distribution for all elements, or separate distributions for each element. In the latter case, the “Split” checkbox is selected (Fig. 4G). If there are many elements to set then it may be convenient to load them from a file. This is done using the same format as for priors in §4.2.1, except that here the distribution quantities (means, standard deviations, *etc...*) can be set as parameters rather than numerical quantities.

2.2.8 Derived

A derived quantity is functionally related to other model parameters through an equation (Fig. 4H). Such a quantity usually has some physical meaning (*e.g.* the reproduction number R_0 in an epidemic model, which is related to the transmission and recovery rates).

2.2.9 Factors

A factor is a way to incorporate a categorical fixed effect into a model (Fig. 4I)¹⁷. It differs from an ordinary parameter because it is defined to have a (potentially weighted) average of exactly one.

Mathematically speaking, suppose the parameter has E elements, indexed by e (note, the parameter may be multi-dimensional). In the simplest case, the factor requires that $\frac{1}{E} \sum_e v_e = 1$, where v_e is the value of element e . If a greater weighting needs to be placed on certain elements, a weight factor w_e can be incorporated, and the constraint becomes:

$$\sum_e v_e \phi_e = 1, \quad \text{where } \phi_e = \frac{w_e}{\sum_e w_e}. \quad (3)$$

Because Eq.(3) involves a sum of terms that add to one, it is natural that a Dirichlet prior is used under inference. Here we define a “modified Dirichlet”, which applies a normal Dirichlet distribution to $v_e \phi_e$ with alpha values specified by

$$\alpha_e = \left(\frac{E-1}{\sigma^2} - 1 \right) \phi_e. \quad (4)$$

where σ is a single hyperparameter which constrains deviations from one. This choice leads to the prior for v_e having a mean of one and a variance

$$\sigma^2 \frac{\frac{1}{\phi_e} - 1}{E-1}. \quad (5)$$

Note, in the absence of weights, *i.e.* $\phi_e = 1/E$, this simplifies to a variance of σ^2 for all elements (or equivalently a standard deviation of σ). The choice of $\sigma=0.5$ represents a relatively uninformative prior choice (allowing for a large ~50% variation in element values v_e). On the other hand, if σ is chosen to be small (*e.g.* $\sigma=0.01$), this would ensure that elements v_e would only be permitted small deviations from one.

¹⁷ Note, if a parameter is set to be a factor, it must have at least one index and cannot be time-varying.

BICI

Model

- Compartments
- Parameters
- Description

Simulation

Inference

Model 2.1: SIRD model with branching A

Objective

- Introduce branching in a population-based model.
- Here infected individuals can either recover or die.

Model

- A single population-based species 'People' is created.
- This contains a classification 'DS' which stands for 'disease status'.
- DS contains four compartments: susceptible S, infected I, recovered R and dead D. Together they are known as the "SIRD model".
- A transmission rate β determines the rate at which individuals become infected.
- A recovery rate γ determines the probability per unit time an infected individual recovers.
- A mortality rate κ determines the probability per unit time an infected individual dies.

Population

- This consists of 100 initially susceptible individuals and two infected.

Edit

Figure 5 – Description. A: Text panel giving a description of the data, model, and analysis (this example is taken from M2.1 in §8), B: Edit this description.

Initial conditions [?]

Type	Details	Number	Spec.	Table
Init. Pop.	-	102	View	Edit

Model

- Initial Conditions
- Parameters
- Run

Initial Conditions

Run

Inference

B: Init. Pop. [?] **C**: Add Pop. [?] **D**: Remove Pop. [?] **E**: Add Ind. [?] **F**: Remove Ind. [?] **G**: Move Ind. [?]

Figure 6 – Initial conditions. This page is used to load initial conditions (as well as any addition or removal of individuals after the start time). A: Table showing loaded data, B: Add information about the initial population, C: Add populations during the simulation, D: Remove populations during the simulation, E: Add individuals to the system, F: Remove individuals from the system, G: Move individuals within the system. Note, C and D are only available for population-based models, and E-G are only available for individual-based models.

2.3 Description

Figure 5 shows the “Model→Description” page that allows users to provide a brief description of the model and any analysis performed. This is not only useful to keep track for personal use, but also makes it easier and more transparent for others to understand what has been done. The description can be edited by clicking on the “Edit” button.

A markdown format is used that allows for the following concise formatting options: “# Title”, “## Subtitle”, “- Bullet point”, “*italic text*” and “**bold text**”. In a Latex style, parameters can be added by enclosing in dollar symbols, e.g. “\$a\$” or “\$b^super_sub\$”.

3) Simulation

Simulation relies on specification of model parameters, along with the initial conditions, and stochastically predicts the dynamics of the compartmental model over a period of time.

3.1 Initial conditions

Figure 6 shows the page where initial conditions are loaded (as well as any individuals or populations externally added, moved or removed after the start time). The table at Fig. 6A shows all loaded information, which can be viewed, edited or deleted. The lower menu bar allows the user to add new data. Below we list the various possibilities:

3.1.1 The initial population

The initial population defines the number of individuals in different compartments at the start time. For population-based models, this must be specified (or if omitted it implies the initial population is zero). For individual-based models, an alternative is to leave the initial population unspecified and add known individuals at the start time (see §3.1.4).

Clicking “⊕ Init. Pop.” (Fig. 6B) allows for the initial population to be specified. This can be defined in one of two different ways:

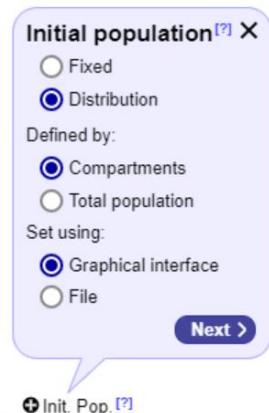
(1) **Fixed** – The populations in the initial state are defined and fixed.

When there is only one classification, each of the compartmental populations is specified. However, when there is more than one classification, two possibilities exist:

- *Focal* – Here a “focal” classification is selected and populations in different compartments are defined for this. For other classifications the percentages of individuals in different compartments are set (note, one compartment is left unspecified, because it is automatically calculated such that the total adds up to 100%). This method is approximate in the sense that it relies on uncorrelated distributions across different classifications.
- *All* – The populations are specified for all possible combinations of compartments across all classifications. This method exactly specifies the initial population.

The population specification is set using:

- *Graphical interface* – Here the point-and-click interface is used to specify populations and/or percentages in the text boxes provided.
- *File* – A file specifying the initial populations:



FOCAL REQUIRES – A data table containing columns “Compartment” and “Population”. The “Compartment” column must go through all compartments in the focal classification and all but one for the other classifications (the remaining is automatically calculated such that percentages/fractions add up to 100%). In the case of the focal classification, the “Population” column gives the number of individuals in each compartment. Other classifications are represented by percentages.

ALL REQUIRES – A data table containing columns with headings given by the classification names along with “Population”, which gives the initial number of individuals within each compartmental combination.

(2) Distribution – The initial population is assumed to be sampled from a distribution. This is set in the following way:

- *Focal* – A population distribution is specified for each compartment in the “focal” classification. For other classifications a Dirichlet distribution is set to give the probability of being in different compartments (this is defined by alpha values α_c for each compartment c such that the probability of being in c is $\alpha_c / \sum_d \alpha_d$).
- *All* – A population distribution is placed on the overall initial population N^{total} , and a Dirichlet distribution is specified for all possible combinations of compartments across all classifications.

The distribution specification is set using:

- *Graphical interface* – The point-and-click interface is used to specify distributions and α values in the text boxes provided.
- *File* – A file specifying the initial distribution:

FOCAL REQUIRES – A data table containing columns “Compartment” and “Distribution”. The “Compartment” column must go through all compartment in the model. In the case of the focal classification, the “Distribution” column gives the distribution the population number is sampled from, e.g. “uniform(10,20)” (see below for formatting). Other classifications are represented by α values used to defined the Dirichlet distributions.

ALL REQUIRES – A data table containing columns for each classification and a final column with “Alpha”. This column defines a Dirichlet distribution across all possible compartment combinations. The distribution for the overall population number must be set once the file is loaded.

Since population distributions must be a strictly positive, they can be selected from the following four possibilities:

- “uniform(min,max)” – A flat prior distribution between a minimum and maximum value.
- “exp(mean)” – An exponential distribution with a specified mean.
- “gamma(mean,cv)” – A gamma distribution with a specified mean and coefficient of variation.
- “log-normal(mean,cv)” – A log-normal distribution with a specified mean and coefficient of variation.

3.1.2 Adding populations to the system

Clicking “⊕ Add Pop.” (Fig. 6C) allows for data on any additional populations that are added to the system after the start time. Note, this is for population-based models only.

REQUIRES – A data table containing at least the following columns: “t” the time the individuals are added, columns for each classification to denote into which compartment the individuals enter the system and “Population”, which gives the number of individuals added.

3.1.3 Removing populations from the system

Clicking “⊕ Remove Pop.” (Fig. 6D) allows for data on any populations that are removed from the system after the start time. Note, this is for population-based models only.

REQUIRES – A data table containing at least the following columns: “t” the time the individuals are removed, columns for each classification to denote from which compartment the individuals leave the system and “Population”, which gives the number of individuals removed.

3.1.4 Adding individuals

Clicking “⊕ Add Ind.” (Fig. 6E) allows for data on any individuals added to the system. These individual can either be added at the start or at any later time.

REQUIRES – A data table containing at least the following columns: “ID” a unique identifier for individuals, “t” the time individuals are added, and columns for each classification to denote into which compartment the individuals enter the system. These compartmental specifications can be exact, e.g. ‘S’, or reflect any potential uncertainty (see §4.1.2.1 for details on how this is implemented).

3.1.5 Removing individuals

Clicking “⊕ Remove Ind.” (Fig. 6F) allows for data on any individuals removed from the system.

REQUIRES – A data table containing at least the following columns: “ID” a unique identifier for individuals and “t” the times individuals are removed.

Simulation parameter values

Parameters

- $a = 1.4$
- $m = 10$ (A)
- $\sigma = 2$

Parameter vectors

- $h_s \sim \text{normal}(m, \sigma)$ (B)
- $\gamma_s = [0.1, 0.2]$ (C)
- $\eta(t) = [1, 1.1, 1.3, 1.2, 0.9, 0.5]$ (D)

Parameter matrices

- $\beta_{s,s'} = [[0.4, 0.5], [0.3, 0.3]]$ (E)

Individual effects

- $\Omega^{g,g} = 1.4$ (F)

Fixed effects

- $v^w = 0.3$ (G)

(H) Sample View View View

Figure 7 – Simulation parameter values. This page is used to specify parameter values before simulation can be performed. A: Univariate parameters, B: Distributions, C: Vectors, D: Splines, E: Matrices, F: Individual effect parameters, G: Fixed effect parameters, H: View the parameter definition.

3.1.6 Moving individuals

This moves individuals to specified compartments in a specified classification at specified times (note, this movement is enforced, and not as a result of an induced change, as would be modelled by a transition).

REQUIRES – A data table containing at least the following columns: “ID” a unique identifier for individuals, “t” the time individuals are moved and “To”, which gives the destination compartment.

3.2 Parameters

Figure 7 illustrates the “Simulation→Parameters” page. This page shows all the parameters that need to be specified before the model can be simulated. They fall into a variety of different categories:

- **Univariate** (Fig. 7A) – Clicking on the blue values opens up a text box that can be edited.
- **Distributions** (Fig. 7B) – If a parameter has been defined as coming from a distribution (on the “Model→Parameters” page), the “Sample” checkbox allows the user to select whether or not to directly set the parameter values, or sample as part of the simulation procedure.
- **Vectors** (Fig. 7C) – Elements of a vector depend on different compartments within a classification (the subscript on the vector corresponds to the classification index). Clicking on the blue value opens up an editor. Values for large vectors can be loaded from a file:

REQUIRES – A data table containing a column heading given by the index of the vector and “Value”, which gives the numerical values of the vector elements.

- **Splines** (Fig. 7D) – These are treated like vectors, but here elements correspond to the knot times specified on the “Model→Parameters” page. Values for splines with many knots can be loaded from a file:

REQUIRES – A data table containing column headings “t” and “Value”. The “t” column contains a list of the knot time, e.g. ‘start’, 20, ..., ‘end’ which must correspond with those specified on the “Model→Parameters” page.

- **Matrices** (Fig. 7E) – These are quantities that depend on compartments in two classifications (which could be the same, as here, or different). Values for large matrices can be loaded from a file:
REQUIRES – A data table containing column headings given by the indexes of the matrix and “Value”.
- **Individual effect parameters** (Fig. 7F) – Here variances are specified that determine the variation in individual effect across the population (see §2.2.2 for details). Variances use the name “ Ω ” with superscript given by the individual effect name (g in this case). For more than one individual effect, correlation coefficients are also specified, which use the name “ ω ”. For example, a model containing two individual effects [g] and [f] would contain variances $\Omega^{g,g}$ and $\Omega^{f,f}$ and correlation parameter $\omega^{f,g}$ (note, values for variances must always be positive and correlations must be between -1 and 1).
- **Fixed effect parameters** (Fig. 7G) – The strength of a fixed effect is determined by a parameter with name “ μ ” and superscript given by the fixed effect name, w in this case (see §2.2.3 for details).

When editing tables of parameter values, the tab key can be used to jump from one element to the next (this can make altering values somewhat quicker as it avoids excessive mouse clicking). Furthermore, values can be copied and pasted using the usual keyboard shortcuts¹⁸.

Once parameters have been specified, a number of different visualisations can be made by clicking on the “View” buttons (Fig. 7H).

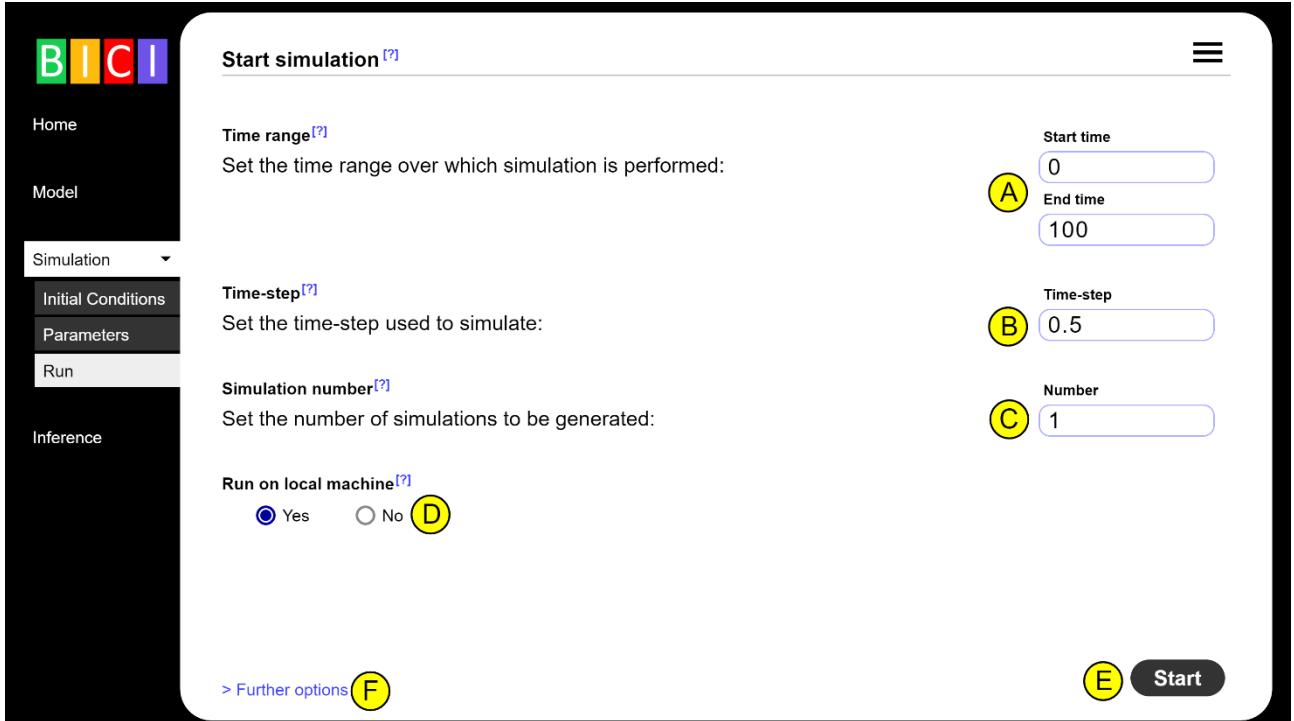


Figure 8 – Running a simulation. This page is used to set up a simulation. A: Time range, B: Time-step, C: Number of simulations to be performed, D: Where the simulation is done, E: Click to start, F: Further options available.

3.3 Run

Figure 8 shows the “Simulation→Run” page, which allows the user to set various options before starting a simulation:

- **Time range** (Fig. 8A) – Sets the time period over which the simulation is performed.
- **Time-step** (Fig. 8B) – A finite time-step is used to approximate the governing equations. A smaller value provides more accurate results but takes computationally longer. Typically setting this to around 20% of the shortest expected transition period gives fast, accurate results.
- **Number** (Fig. 8C) – This sets how many simulations are performed. Because simulations are stochastic, so each will provide a different potential realisation from the system. Visualising the overall envelope of many simulations can be a useful way to predict probability distributions for likely future behaviour.
- **Run on local machine** (Fig. 8D) – BICI can either be run on the local machine (for small jobs) or on a Linux cluster (for big jobs, see §6.4). Because simulation is generally much faster than inference, most jobs can be run locally.

¹⁸ On windows: ctrl+x for cut, ctrl+c for copy, and ctrl+v for paste.

Clicking on the “Start” button (Fig. 8E) gets a simulation under way. Once complete, the user is directed to the “Results” page (see §3.4).

3.3.1 Further options

Additional features can be specified by clicking on “Further options” (Fig. 8F):

- **Individual max** - When one or more species use an individual-based model, this sets the maximum number of individuals allowed (exceeding this limit results in an error message). Such a threshold is introduced to stop BICI running out of memory when the individual number diverges. By default, this is set to 20,000 individuals. The text box can be used to make this limit higher, if required.
- **Parameter output** – If the number of elements in a tensor exceeds this value, results are not output. This is introduced to stop very large output files (for example it might avoid the distance matrix being output every MCMC iteration). By default, this is set to a maximum of 1000 elements, but can be changed using the text box.
- **Set random seed** – Simulation and inference rely on pseudorandom number generators. These start with an initial value called a “seed”. If the seed option is turned off (as it is by default), this seed is randomly generated by the BICI interface (in which case BICI will produce different results each time it is run). If the seed is set, however, BICI will consistently give the same results, despite being stochastic. The seed can take any value between 0 and 10,000.

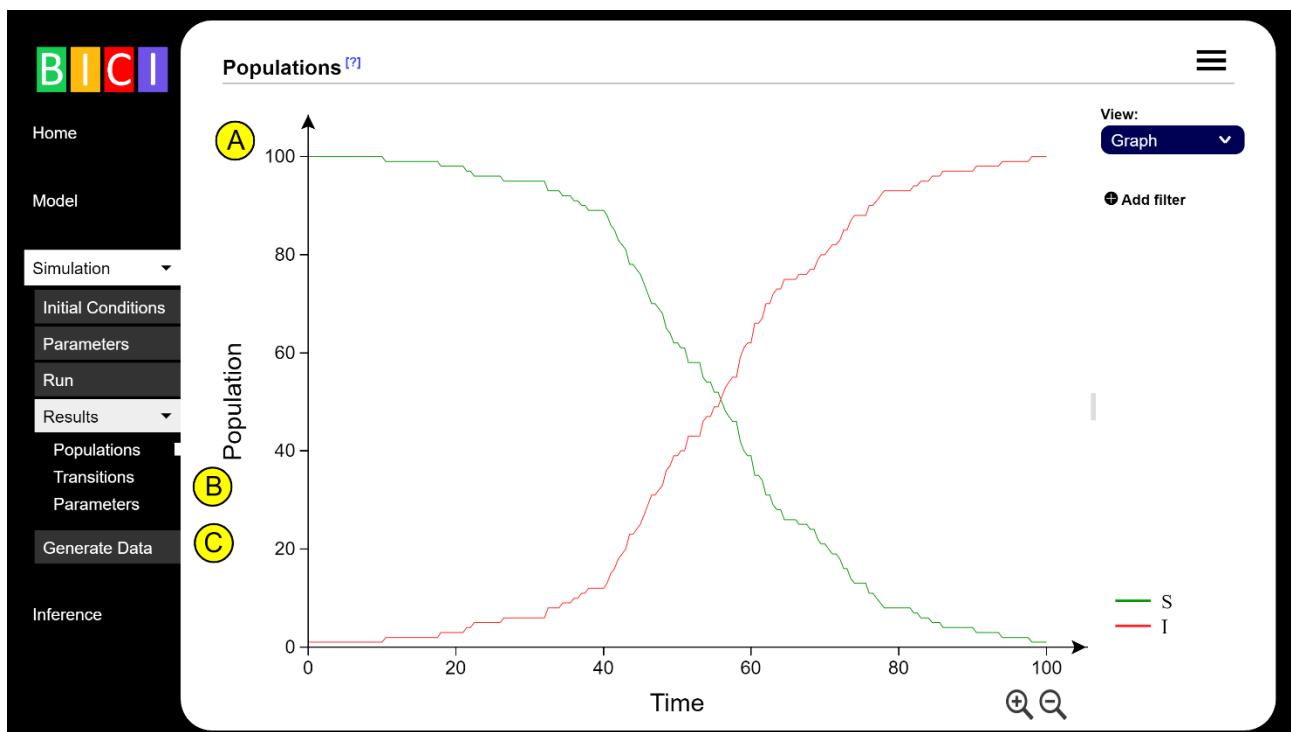


Figure 9 – Results from a simulation. A: This shows the time variation in the populations in the S and I compartments (taken from M1.1 in §8), B: Click on different menu options for visualisations of populations, transition or parameters, C: Click to generate simulated data.

3.4 Results

The “Simulation→Results” page shows outputs from a simulation. The illustrative example in Fig. 9A gives a plot of the populations in different model compartments as a function of time. Here there are initially 100 susceptible individuals (S compartment, denoted by the green line) and just a single infected (I

compartment, denoted by the red line). Due to disease transmission, the susceptible individuals increasingly make the transition to becoming infected.

Different sections on the main menu allow for a variety of visualisations (Fig. 9B) focusing on “Populations”, “Transitions” and “Parameters” (and also “Individuals” in the case of individual-based models). Further details for these are provided in §4.4 and Appendices C-F.

Simulated data [?]				
Type	Details	Number	Spec.	Table
Add Ind.	-	101	View	Edit X
Compartment	DS	505	View	Edit X
Transition	S→I:0.5	51	View	Edit X
Diag. Test	DS Se=1 Sp=1	505	View	Edit X

Below the table:

- Individual
- + Compartment [?]
- + Transition [?]
- + Diag. Test [?]
- + Genetic data [?]

Bottom right: Copy

Figure 10 – Generating simulated data. A: A table of simulated data sources, B: Choose data type, C: Select data source to be simulated, D: Click to copy simulated data to the inference page.

3.5 Generating data

The “Simulation→Generate Data” page on Fig. 10 can be used to create data based on simulated results. These hypothetical datasets can then be used as inputs for performing inference. This not only provides a good test to show that inference is able to successfully recover model parameters, but it is also an invaluable tool in experimental design, because it allows the user to estimate the likely power of a given setup¹⁹.

The drop-down menu (Fig. 10B) allows for different data types to be selected: “Init. Cond.”, “Individual”, “Population” and “Additional”. These are examined in the following sections:

3.5.1 Simulated initial conditions data

Clicking these options essentially recreates any data that was entered on the “Simulation→Initial Conditions” page.

¹⁹ In particular, it allows for an estimation of model parameter uncertainties, which are often key to understanding whether a given experiment would be sufficiently informative to answer a scientific question under study.

3.5.1.1 Init. Pop

Click “⊕ Init. Pop.” to generate data that represents the initial population from the simulation. Selecting “Done” adds this data to the list in Fig. 10A (as it does with all the other options below).

3.5.1.2 Adding populations to the system

Click “⊕ Add Pop.” to generate data for any populations added after the start time.

3.5.1.3 Removing populations from the system

Click “⊕ Remove Pop.” to generate data for any populations removed after the start time.

3.5.1.4 Adding individuals

Click “⊕ Add Ind.” to generate data for individuals added in the simulation.

3.5.1.5 Removing individuals

Click “⊕ Remove Ind.” to generate data for individuals removed in the simulation.

3.5.1.6 Moving individuals

Click “⊕ Move Ind.” to generate data for individuals moved in the simulation.

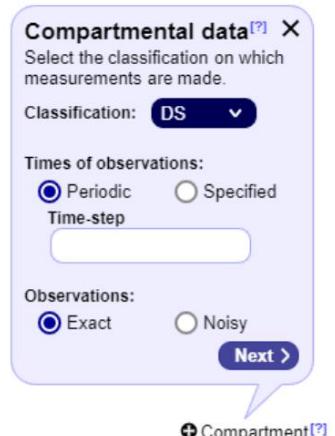
3.5.2 Simulated individual-level data

3.5.2.1 Compartment data

Click “⊕ Compartment” (Fig. 10C) to generate compartmental data.

The classification on which compartments are measured is selected from the drop-down menu (see right, ‘DS’ in this example). The timings of the compartmental observations can either be made at periodic intervals, or at specified time points.

The observations can be chosen to be exact (which will generate results like ‘S’ or ‘I’) or noisy. In the latter case, a specified value represents the probability p of observing the correct compartment. The remaining probability $1-p$ is equally distributed across all other compartments. For example, in an SIR model if $p=0.9$ and the true compartment is I, the simulated output would be ‘S:0.05|I:0.9|R:0.05’ (see §4.1.2.1 for formatting compartmental uncertainties).



3.5.2.2 Transition data

Click “⊕ Transition” to generate transition data. Please refer to §4.1.2.2 for further details.

3.5.2.3 Diagnostic test data

Click “⊕ Diag. Test” to generate disease diagnostic test data. The option to either measure periodically or at specified time points is given. Please refer to §4.1.2.3 for further details.

3.5.2.4 Genetic data

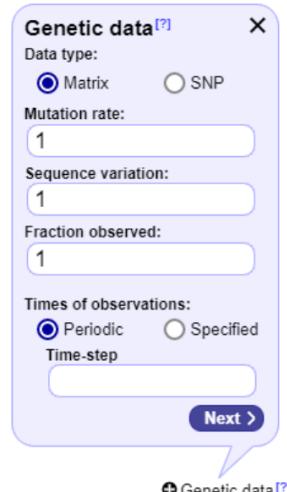
Click “⊕ Genetic data” to generate genetic data.

The nature of this data depends on its type:

- **Matrix** – For a given set of pathogen genetic measurements, this generates a matrix of SNP differences between those measurements.
- **SNP** – For a given set of pathogen genetic measurements, this option outputs simulated SNP data. If selected, it is necessary to specify a text “root” used for SNP names, along with the number of SNP base pairs to be simulated.

Other options are as follows:

- **Mutation rate** – This sets the rate at which the virus genome mutates (in base pair mutations across the whole genome per unit time).
- **Sequence variation** – This allows for variation in sequences for infections entering the system. The number of changes compared to a consensus sequence is drawn from a Poisson distribution with this specified mean.
- **Fraction observed** – Gives the proportion of infected individuals observed at a given time point (in real datasets this fraction is often quite low).
- **Times of observations** – The times at which individuals are observed. This can either be set as periodic or at specified time points.



⊕ Genetic data [?]

3.5.3 Simulated population-level data

3.5.3.1 Population data

Click “⊕ Population” to generate population data. Measurements can either be made periodically, or at specified time points. Please refer to §4.1.3.1 for further details.

3.5.3.2 Population-level transition data

Click “⊕ Pop. Transition” to generate population-level transition data. Measurements can either be made periodically, or at specified time points (note, here the observed number of transitions is measured between these time points). Please refer to §4.1.3.2 for further details.

3.5.4 Simulated additional data

3.5.4.1 Individual effect data

Click “⊕ Ind. Effect” to generate individual effect data. Select the individual effect required from the drop-down menu, and the data source generated will contain the simulated values for the individual effect.

When performing inference this is useful because it allows for the prediction accuracy to be estimated (a metric that determines how well inference is able to recover true individual effect values).

3.5.4.2 Individual group data

Click “⊕ Ind. Group” to specify groups of individuals in the simulated population. This can be useful when estimating prediction accuracies for specified groups. Please refer to §4.1.4.2 for further details.

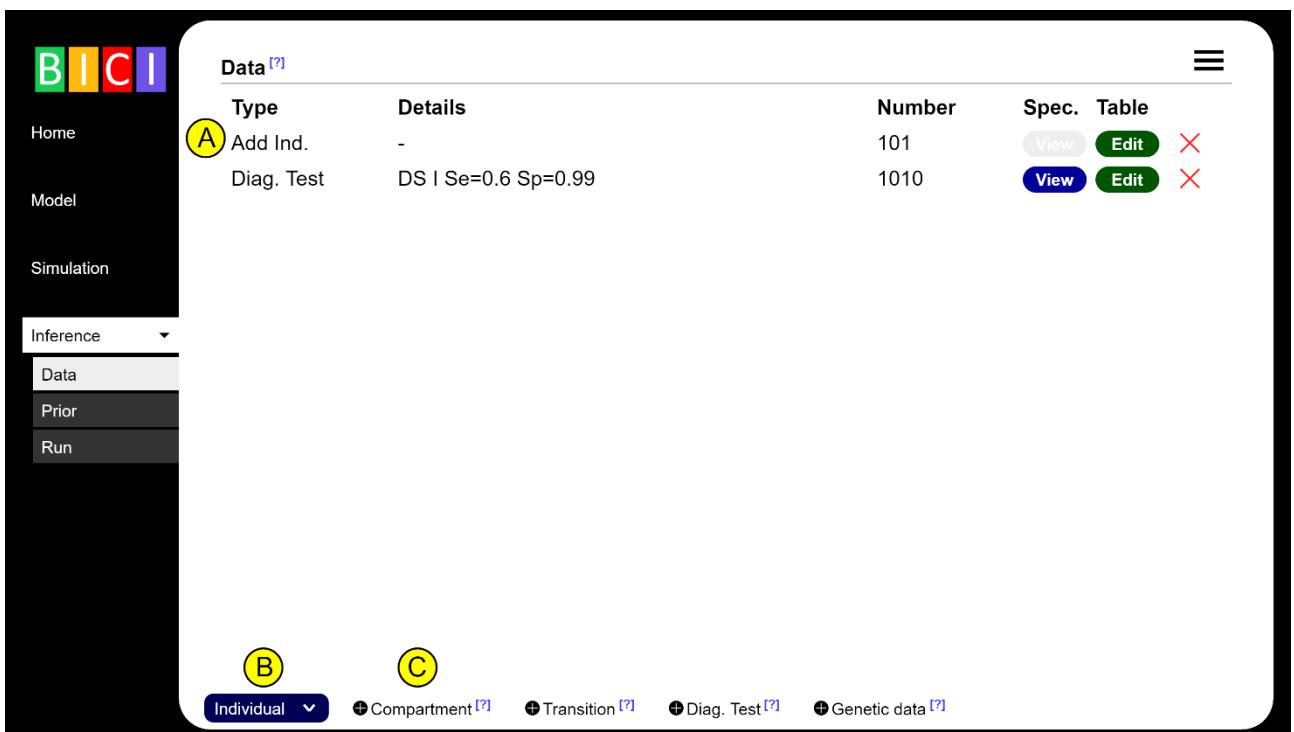


Figure 11 – Loading data. A: Table of data sources, B: Choose data type, C: Select data source to be added.

4) Inference

Bayesian inference is the process of combining data (which can take a variety of forms and may be inherently noisy) with previous knowledge regarding model parameters (the prior) to generate the best available estimates for model parameters and system dynamics (the posterior).

4.1 Data

Rather than loading the data all at once, the user loads different sources of data one at a time (in any order). As illustrated in Fig. 11A, the “Inference→Data” page shows the data sources currently loaded²⁰.

The data comes in different types, as selected using the drop-down menu in the bottom left-hand corner (Fig. 11B):

- **Init. Cond.** (§4.1.1) – Provides information about the initial conditions, as well as any individuals that are added, removed or enforced to move compartment.
- **Individual** (§4.1.2) – Individual-level data, such as the times at which individuals undergo a given transition, when they are in a given compartment or disease diagnostic test results. Genetic sequence data for the pathogen can also be added, if available.
- **Population** (§4.1.3) – Population-level data, such as time-series compartmental population measurements, or time-series population-level transition numbers.

²⁰ This example is taken from example C4 in §8 and shows two sources related to adding individuals at the start time and disease diagnostic test results.

- **Additional** (§4.1.4) – This provides addition information, not used for inference. Here true values for individual effects can be loaded (such that prediction accuracies can be calculated) or individual groups can be specified (*e.g.* sires and progeny in a quantitative genetics model).

Data is loaded into a BICl using a dialogue box (see right). A: If a data table has previously been loaded (*e.g.* for another data source) it can be directly selected from the list, B: This determines if the table has a heading row at the top (recommended), C: Two formats for data tables are supported, either columns are comma separated ('.csv' files) or tab separated ('.txt'/'.tsv' files), D: Clicking "Upload" allows the user to upload a new file.



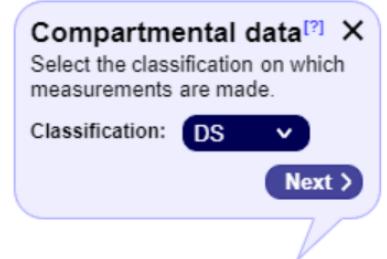
4.1.1 Initial conditions data

This uses exactly the same set of possibilities as for simulation, so the reader is referred to §3.1.

4.1.2 Individual-level data

4.1.2.1 Compartment data

Click “⊕ Compartment” (Fig. 11C) to add compartmental data. This provides information about the compartments individuals are in at specified time points, *e.g.* it could be used to denote the fact that individual “Ind. 1” is observed to be in the infected I compartment at time $t=3.45$ and “Ind.2” is in the recovered R compartment at time $t=9.21$.



REQUIRES – A data table containing at least the following columns: “ID” a unique identifier for individuals, “t” the measurement times and the classification on which the measurements are made, *e.g.* “DS”.

The last column contains compartmental specifications that can be made in a number of different ways to reflect any potential uncertainty:

- **Precisely** – The known compartment is exactly specified, *e.g.* ‘S’ would mean that the individual is in the S compartment at the measurement time.
- **Multiple choices** – Different possibilities are given, separated by the ‘|’ character, *e.g.* ‘S|I’ would mean an equal probability of being in S or I.
- **Probabilities** – The probabilities of being in any potential compartments are explicitly specified, *e.g.* ‘S:0.7|I:0.3’ would mean a 70% probability of being in S and a 30% probability of being in I. Such an expression could also contain parameters, *e.g.* ‘S:p|I:1-p’, that could be estimated during inference.
- **Unspecified** – This is denoted by ‘.’ and indicates missing data.
- **Non-existence** – If an individual is not in the system when a compartmental observation is made, this can be indicated by the ‘!’ character.

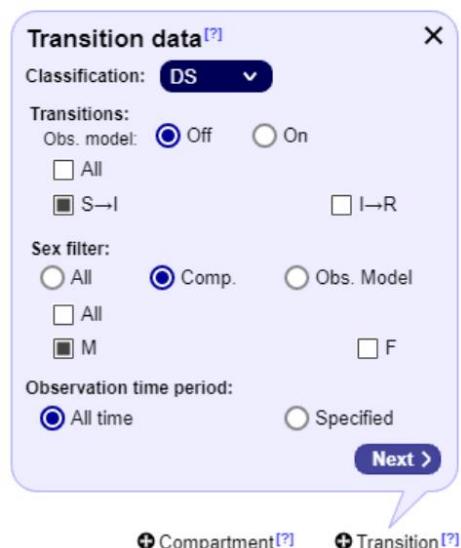
4.1.2.2 Transition data

Click “ Transition” to add transition data. This provides information about the timings of individual transitions, e.g. it could be used to store the precise infection times for a group of individuals.

REQUIRES – A data table containing at least the following columns: “ID” a unique identifier for individuals²¹ and “t” the times individuals undergo the transition.

When loading transition data, the following options are available:

- **Classification** – Specifies the classification within which the transition observations are made.
- **Transitions** – Checkboxes can be used to select the transition (or transitions) under observation. Alternatively, an observation model can be specified that gives the probability of observing a given transition, e.g. ‘ $P(S \rightarrow I)=0.3$ ’ would mean a 30% probability of observing an infection transition. Note, parameters can be placed into the observation model (e.g. ‘ $P(S \rightarrow I)=p$ ’ or ‘ $P(S \rightarrow I)=\eta(t)$ ’ to incorporate time variation) whose parameters can be estimated during inference.
- **Compartmental filter** – When there is more than one classification, the compartments in the other classifications can be used to filter which individuals are actually observed. Three possibilities exist:
 - *All* – All compartments are observed.
 - *Comp.* – Uses checkboxes to indicate observed compartments.
 - *Obs. Model* – Allows for specification of an observation model (again, this can involve parameters with potential time variation).
- **Observation time period** – Two options exist:
 - *All time* – Goes over the entire inference period.
 - *Specified* – Allows for a time window to be set.



²¹ Note, if no ID is associated with the observations, one can be arbitrarily be made up, e.g. if only recovery times are known from an epidemic then the first recovered individual can be called “Ind. 1”, the second “Ind. 2”, and so on and so forth.

4.1.2.3 Diagnostic test data

Click “⊕ Diag. Test” to add disease diagnostic test data. This provides a (noisy) way to assess if individuals are infected or not, e.g. such data could be used to denote the fact that individual “Ind. 1” was tested at time $t=34.5$ and gave a positive result, whereas “Ind. 2” was tested at time $t=14.0$ and was found to be negative.

REQUIRES – A data table containing at least the following columns: “ID” a unique identifier for individuals, “t” the times the individuals were tested, and “Result”, which provides the binary results of the test (see below for what values this can take).

When loading diagnostic test data, the following options are available:

- **Classification** – Specifies the classification within which the observations are made.
- **Infected compartment(s)** – Checkboxes are used to denote which compartments the test aims to give a positive result for.
- **Diagnostics** – Values for the test sensitivity and specificity are set (note, these can be parameters estimated under inference).
- **Text values** – The text used to represent positive and negative test results in the data file (e.g. ‘0’/‘1’ or ‘+ve’/‘-ve’).

4.1.2.4 Genetic data

Click “⊕ Genetic data” to incorporate pathogen genetic sequence data. Such data can help inform how a pathogen moves between individuals, e.g. if a genetic sequence for the pathogen is observed in individual “Ind. 1” at $t=10.5$ and a closely related (or identical) genetic sequence is observed in “Ind. 2” at time $t=15.6$, it is evidence that the pathogen may have been transmitted from “Ind. 1” to “Ind. 2”.

Pathogen genetic data can be particularly good at estimating rates of transmission between groups (e.g. different species) or geographical spread.

Two types of data can be loaded:

- **Matrix** – A matrix of SNP differences between a set of pathogen genetic measurements.
- **SNP** – Raw SNP data for a set of pathogen genetic measurements.

The following options must be set:

- **Mutation rate** – A parameter giving the rate at which the virus genome mutates (in base-pair changes across the whole genome per unit time).
- **Sequence variation** – A parameter that allows for variation in sequences for infections entering the system. The number of changes compared to a consensus sequence is drawn from a Poisson distribution with this specified mean.

Diagnostic tests [?]

Select the classification the test is sensitive to:

Classification: DS

Select the compartments the test is positive to:

All S I R

Sensitivity and specificity of the diagnostic test:

Sensitivity: 0.6 Specificity: 0.99

Set the text used to represent test results:

Positive text: + Negative text: -

Next >

⊕ Diag. Test [?]

Genetic data [?]

Data type:

Matrix SNP

Mutation rate:

1m

Sequence variation:

V

SNP root:

SNP

Next >

⊕ Genetic data [?]

- **SNP root** – Gives the root name for SNP columns in the file (for SNP data).

REQUIRES – A data table containing at least the following columns: “ID” gives a unique name for each individual, “t” giving the time of the observation. The following columns depend on the type of observation:

- **Matrix** – A column “Obs” provides a unique name for each genetic observation. Elements down this column must also be column headings in the table that represent a square matrix giving base-pair differences between the observations.
- **SNP** – The data table has a series of columns giving sequence data which have ‘SNP root’ at the beginning of their column name. Elements of these columns are expected to contain two characters denoting the version of the SNP, e.g. ‘AA’, ‘AB’, ‘BA’, or ‘BB’.

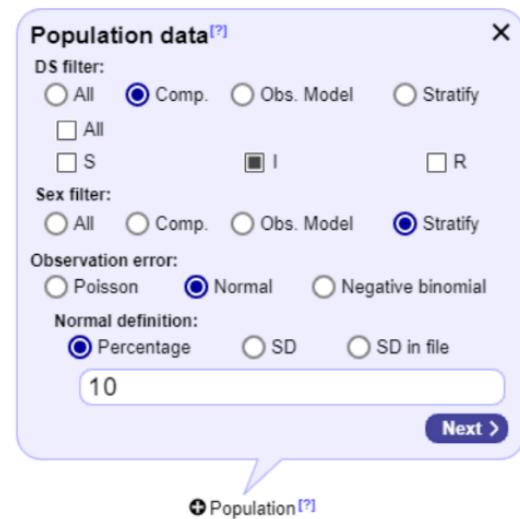
4.1.3 Population-level data

4.1.3.1 Population data

Click “+ Population” to add population data. This informs how many individuals are in a specified compartment (or compartments) at different times, e.g. it could be measurements for the number of infected individuals at a number of different time points (so-called time series data).

REQUIRES – A data table containing at least the following columns: “t” the time at which population measurements are made and “Population”, which gives the estimated number of individuals.

The population under study can be filtered for each of the classifications using the following options:



- **All** – All individuals within the classification are counted.
- **Comp.** – Selected checkboxes are used to choose certain compartments (e.g. only infected individuals in compartment I).
- **Obs. Model** – An observation model can be specified that gives the fraction of individuals observed in one or more compartments, e.g. ‘ $P(S)=0.3$ ’ would mean that only a 30% fraction of individuals are observed in the S compartment (note, fractions can be larger than one, corresponding to cases in which the observed data is proportional to the population, e.g. a measurement of pathogen level would be expected to be proportional to the number of infected individuals). Parameters can be placed into the observation model (e.g. ‘ $P(S)=p$ ’ or ‘ $P(S)=\eta(t)$ ’ to incorporate time variation) that can be estimated during inference.
- **Stratify** – The data file contains an additional column (with heading given by the classification name) that specifies the compartment corresponding to that particular observation.

The observation error captures noise in the measurement process itself. This assumes that the observed value is drawn from a distribution with mean given by the underlying value.

Three possible distributions can be selected from:

- **Poisson** – This has a variance equal to the mean.
- **Normal** – This allows for more flexibility. The standard deviation can be defined in one of three different ways:
 - *Percentage* – The standard deviation is set to a percentage of the mean.
 - *SD* – The standard deviation is explicitly fixed.
 - *SD in file* – The data table contains an additional column “SD” that specifies the standard deviation.
- **Negative binomial** – This distribution is overdispersed with respect to the Poisson distribution. Specifically, the variance is given by the mean divided by a probability p . This probability can be defined to one of two ways:
 - *Value of p* – The probability p is explicitly fixed.
 - *p in file* – The data table contains an additional column “p” which specifies its value.

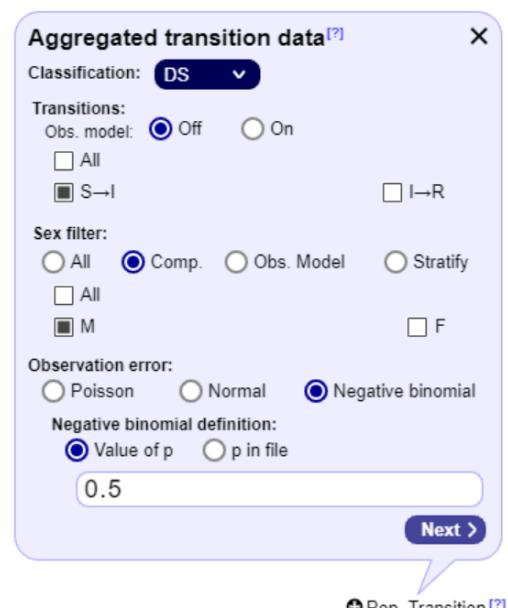
4.1.3.2 Population-level transition data

Click “⊕ Pop. Transition” to add aggregated population-level transition data. Here the observation period is sliced into divisions (usually of equal size, although this is not a requirement) and the overall number of transitions of a specified type is counted within these divisions, *e.g.* such data could represent the number of population-wide cases per week during an epidemic. The size of the divisions would usually be chosen to be sufficiently large such that a reasonable number of events is expected to occur²², but sufficiently small to retain temporal resolution.

Note, if data on transition timings exist, it is usually best to incorporate these as individual transition data, as described in §4.1.2.2, rather than population-level data²³.

REQUIRES – A data table containing at least the following columns: “Start” and “End” that give the start and end time of the divisions over which transitions are counted (note, the end time of one division would normally be the same as the start time for the next) and “Number”, which gives the estimated number of transitions.

Either one or multiple transitions can be selected within a given classification. Alternatively, an observation model can be specified that denotes the fraction of transitions observed (*e.g.* $\mathbb{P}(S \rightarrow I) = 0.5$ ’ would correspond to only 50% of infection transitions being observed).



²² Say, more than five during the peak of an epidemic outbreak, such that stochastic noise from the Poisson process does not dominate.

²³ Note, if no ID is associated with the observations, one can be arbitrarily be made up, *e.g.* if only recovery times are known from an epidemic, the first recovered individual can be called “Ind. 1”, the second “Ind. 2”, and so on and so forth.

If, rather than time series data, only the overall number of transitions is known, this can be incorporated by setting the start and end times to encompass the entire time period, *e.g.* this could be used when only the total number of recoveries is known at the end of an epidemic (as estimated from a serological test).

See §4.1.2 for further details on classification filters and the observation error.

4.1.4 Additional data

4.1.4.1 Individual effect data

Click “⊕ Ind. Effect” to load up true values for individual effects (to compare against estimated values from inference). Note, these true values are usually generated from simulated data (see §3.5.4.1). This data type has no influence on inference.

REQUIRES – A data table containing the column “ID” a unique identifier for individuals and “Value”, which gives the true individual effect values.

4.1.4.2 Individual group data

Click “⊕ Ind. Group” to load up groupings of individuals. Note, this is used for visualisation only (*e.g.* to view individual effects for parents) and has no bearing on inference. The group must be given a name (*e.g.* “sires”) and individuals within the group can either be specified in the interface (using checkboxes) or from a file:

REQUIRES – A data table containing the column “ID”, a unique identifier for individuals that lists the members of the group.

4.1.5 Table editing

When loading or editing data tables, BICI provides some simple tools that can be accessed by either clicking on the table headings or row numbers:

- **Search** – Find rows containing a specified string. If multiple results are returned, these can be iterated through. A wildcard ‘*’ can be used to represent any ignored text (*e.g.* “S*” would find all entries beginning with ‘S’, or “*ind*” would return all entries containing ‘ind’).
- **Replace** – Replace one text string with another along a column. Wildcards ‘*’ can be used in this replacement. For example, searching for “S*” and replacing it with “R*” would replace all entries beginning with ‘S’ to corresponding entries beginning with ‘R’. Note, the same number of wildcards must be used in both the search and replace text boxes.
- **Delete** – Delete rows for which a given column value corresponds to a specified string. A wildcard ‘*’ can be used to represent text that is ignored (*e.g.* “S*” would delete all entries beginning with ‘S’, or “*ind*” would delete all entries containing ‘ind’)
- **Order** - Orders rows based on elements along a column. This ordering can either be performed numerically or alphabetically.

Data table	
t	Population
1	3
2	12
3	62
4	93
5	100

Parameter priors

Parameters

 $\eta \sim \text{uniform}(0,10)$ (A)

Parameter vectors

 $G_g \sim \text{normal}(0,\sigma^G)$
 $\beta_v \sim [\text{gamma}(1,0.3), \text{gamma}(1.5,0.5)]$

Individual effects

 $\Omega^{g,g} \sim \text{exp}(1)$

Fixed effects

 $v^w \sim \text{normal}(0,2)$

Split Split (B)

Figure 12 – The prior. A: A list of all parameters requiring priors, B: Selecting this checkbox allows for the prior to be specified for different elements of the parameter.

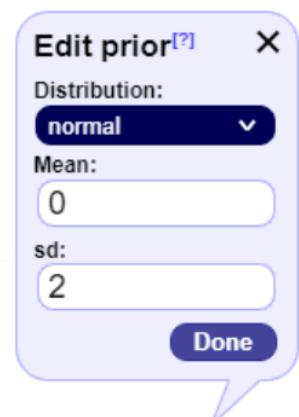
4.2 Prior

The “Inference→Prior” page on Fig. 12 allows for prior distributions to be assigned to model parameters (note, this list doesn’t include constants, distributions or derived parameters).

When Bayesian inference is performed, priors are used to encapsulate any previous knowledge about model parameters. One common approach is to identify a range of plausible values a given parameter is likely to take (based on expert judgment), and use this to inform a uniform distribution.

Clicking on the prior definitions (Fig. 12A) allows for them to be edited. For most parameters, prior distributions can be selected from the following eight different possibilities:

- **uniform** – A flat prior distribution between a minimum and maximum value.
- **exp** – An exponential distribution with a specified mean (only appropriate if the parameter is strictly positive).
- **normal** – A normal distribution with a specified mean and standard deviation.
- **gamma** – A gamma distribution with a specified mean and coefficient of variation (for positive parameters only).
- **log-normal** – A log-normal distribution with a specified mean and coefficient of variation (for positive parameters only).



- **beta** – A beta distribution with specified α and β values (for parameters that go between 0 and 1 only). Note, the distribution mean is given by $\alpha/(\alpha + \beta)$.
- **bernoulli** – A Bernoulli distribution with a specified mean (for binary parameters that can take the values 0 or 1).
- **fix** – Fixes a parameter to a particular value.

When parameters are set to be categorical factors, a special modified Dirichlet distribution “mdir” is used (see §2.2.9).

When editing tables of priors, the tab key can be used to jump from one element to the next (this can make altering values somewhat quicker as it avoids excessive mouse clicking). Furthermore, priors can be copied and pasted using the usual keyboard shortcuts²⁴.

4.2.1 Split

If the “Split” checkbox is selected (Fig. 12B), it means that a prior specification is made separately for each element of the parameter. If many such specifications are required, these can be loaded from a data file:

REQUIRES – A data table containing columns with headings given by the indices of the tensor and a final “Prior” column. See below for the text format used for this last column.

4.2.2 Text representation for a prior or distribution

When written in text format, priors (or distributions) are expressed in the following way, dependent on distribution:

- “**uniform(min,max)**” – A uniform distribution with specified minimum and maximum, e.g. “uniform(4,5)”.
- “**exp(mean)**” – An exponential distribution with specified mean (only appropriate if the parameter is strictly positive).
- “**normal(mean,sd)**” – A normal distribution with a specified mean and standard deviation.
- “**gamma(mean,cv)**” – A gamma distribution with a specified mean and coefficient of variation (for positive parameters only).
- “**log-normal(mean,cv)**” – A log-normal distribution with a specified mean and coefficient of variation (for positive parameters only).
- “**beta(α,β)**” – A beta distribution with specified alpha and beta values.
- “**bernoulli(mean)**” – A Bernoulli distribution with a specified mean (for binary parameters that can take the values 0 or 1).
- “**fix(value)**” – Fixes a parameter to a particular value.
- “**mdir(σ)**” – A modified Dirichlet distribution with specified sigma. Note, this is only used when parameters are specified as categorical factors (see §2.2.9).

²⁴ On windows: ctrl+c for copy, and ctrl+v for paste.

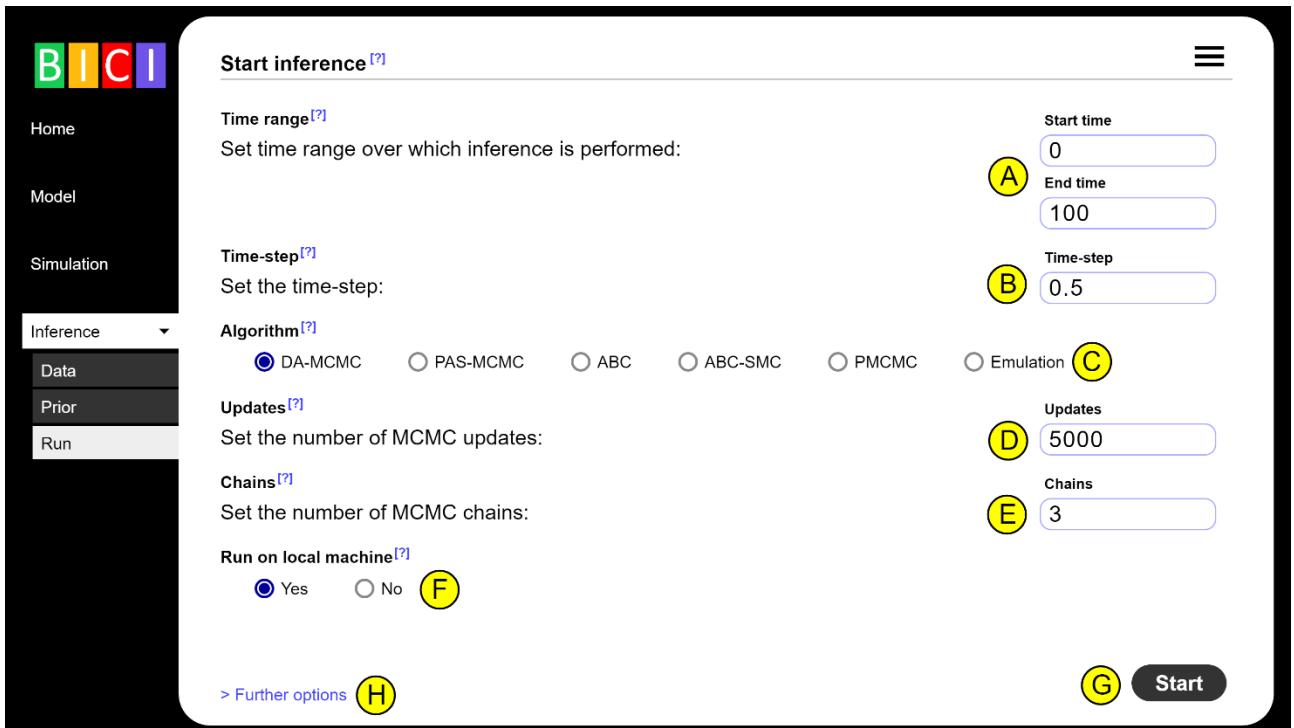


Figure 13 – Running inference. This page is used to set up inference. A: Time range, B: Time-step, C: Algorithm, D: Number of MCMC “updates”, E: Number of MCMC chains, F: Where to run, G: Click to start, H: Further options available.

4.3 Run

The “Inference→Run” page allows the user to set various options before starting inference:

- **Time range** (Fig. 13A) – Sets the period over which inference is performed.
- **Time-step** (Fig. 13B) – For approximate methods it is often necessary to specify a time-step. This should be chosen to be sufficiently long to allow the algorithm to be fast, but sufficiently short for it to be accurate. A typical compromise would be to set it to around 20% the expected shortest transition distribution mean.
- **Algorithm** (Fig. 13C) – Different algorithms can be used to perform inference (see next section).
- **Run on local machine** (Fig. 13F) – BICI can either be run on the local machine (for small jobs) or on a Linux cluster (for big jobs, see §6.4).

Inference is run by clicking on the “Start” button at Fig.13G. Once complete, the user is directed to the “Results” page (see §4.4).

4.3.1 Inference algorithms

Many different inference algorithms exist within the literature, of which a few have been implemented into BICI. Some are exact, some are approximate, and some require communications between interacting processes (which requires them to be run on a Linux cluster).

4.3.1.1 DA-MCMC

Data augmentation MCMC is the standard approach to Bayesian inference (although it can be slow for many individuals). This option requires further specifications:

- **Updates** (Fig. 13D) – An “update” constitutes a whole series of “proposals” that make changes to different aspects of the model (*i.e.* different parameters and/or parts of the state) in a range of ways. The number of updates informs how long MCMC takes to run. Running with more updates may produce a better approximation to the posterior, but it can also become computationally wasteful. *A priori* it is difficult to know what this number should be, but the default value of 5000 is usually a good first guess. MCMC diagnostics can be used to determine if inference needs to be rerun using more updates (see §4.4.7 for details).
- **Chains** (Fig. 13E) – MCMC “chains” are run in parallel, so helping to speed up computation. Each chain is run on a separate CPU core. Since modern computers typically have 4-8 cores, so the default number of chains is set to 3 (to avoid the computer slowing down too much). An additional advantage of running parallel chains is that it tells us something about the reliability of the results through the so-called “Gelman-Rubin” statistic (see §4.4.7).

4.3.1.2 PAS-MCMC

This generates a power posterior approximation that get successively closer to the true posterior over a series of generations. A “particle” denotes a combination of a parameter set and a system state. MCMC updates act on particles to change their state. In “particle annealed sampling” (PAS), particles initially map out the prior. Over successive generations (where particles are duplicated or culled) the algorithm passes from the prior to the posterior. Once this is complete, a sampling phase generates posterior samples in much the same way as DA-MCMC. Requires:

- **Updates** – The same as DA-MCMC above.
- **Particles** – Sets the number of particles run in parallel. These help to stop the system getting stuck in metastable states in the burn-in phase. Particles become chains after burn-in.
- **Update per generation** – This sets how many MCMC updates are performed per generation during the burn-in phase (this is set as a percentage of the overall number of MCMC update, with 1% used by default²⁵). A larger number implies slower annealing, but takes longer.

4.3.1.3 ABC

Approximate Bayesian Computation. Estimates the posterior by simulating from the model and comparing the output with the data (population-based data only). Only those samples that are sufficiently close (as measured by an error function) are used for the posterior approximation. For simplicity the error function is defined as minus the log of the observation probability. Requires:

- **Samples** – Sets the number of posterior samples to be generated.
- **Acceptance** – This gives the fraction of simulated sample retained as posterior samples (must be between zero and one). Smaller values yield better results, but take computationally longer.

4.3.1.4 ABC-SMC

Approximate Bayesian Computation using Sequential Monte Carlo (ABC-SMC) is a version of ABC that improves posterior estimates over a series of generations (population-based data only). Requires:

- **Samples** – Sets the number of posterior samples to be generated.
- **Generations** – This gives the number of generations over which the posterior estimate is improved (typically between 5 and 10).

²⁵ Hence, if 100 generations are required during burn-in, the CPU time for burn-in and sampling would be approximately the same.

4.3.1.5 PMCMC

Particle MCMC targets the true posterior and makes use of multiple “particles” (stochastic simulations from the model) run in parallel.

4.3.1.6 Emulation

Here an emulator is used to approximate the posterior surface. This can be much faster than other approaches, provided the number of model parameters is not too large.

4.3.2 Further options

Additional features can be specified by clicking on “Further options” (Fig. 13H):

- **Output parameter samples** (DA-MCMC and PAS-MCMC) – Because MCMC samples are correlated, it makes sense to thin the results (to limit computational processing and memory requirements). BICI works by specifying a certain number of output parameter samples to be generated, which are drawn equally along the MCMC chain(s). Consequently, increasing the number of MCMC updates does not change the number of output parameter samples generated, but it does reduce any potential correlations between those samples. By default, 1000 parameter samples are generated. Selecting a higher value yields smoother posterior distribution plots, but requires more memory and CPU processing.
- **Output state samples** (DA-MCMC and PAS-MCMC) – As above, but for state samples. By default, 200 state samples are generated (note, this is significantly lower than the default number of parameter samples due to the larger memory requirement needed to represent each state). Selecting a higher value yields less noisy plots for populations, transitions and individuals, but requires more memory and CPU processing.
- **Burn-in** (DA-MCMC) – The burn-in phase in MCMC is the initial period that moves from a starting random guess for parameter values (typically sampled from the prior) to a sample representative of the posterior.

The burn-in period is usually specified as a percentage of the overall number of MCMC updates (by default 20% is used).

Annealing - This is an approach that can help to stop MCMC becoming stuck in metastable states during the burn-in phase. This introduces an “inverse temperature” parameter ϕ that determines to what extent the data is incorporated into the analysis. When $\phi=0$, MCMC maps out the prior and when $\phi=1$, the normal posterior is generated.

Annealing works by gradually varying ϕ from 0 to 1 (at which point the posterior is then sampled).

Precisely how this is done can be selected from the following options:

- *none* – No annealing is performed and $\phi=1$ is fixed during burn-in.
- *scan* – ϕ is optimally scanned using a specified ‘rate’. Note, here the burn-in period depends on this rate (with smaller values yielding more gradual annealing, but taking computationally longer) rather than a percentage.
- *log-auto* – An automatic logarithm model is applied to ϕ .
- *power-auto* – An automatic power-law model is applied to ϕ .
- *power* – A power law model is applied to ϕ , where the ‘power’ property must be set.
- **Acceptance fraction** (ABC-SMC) – In ABC-SMC, first samples are generated and then a cut-off in the error function is automatically chosen such that this specified fraction of samples make up the next generation. Smaller values for this fraction take computationally longer, but with fewer generations required.

- **Kernel size** (ABC-SMC) – Within ABC-SMC, particles are selected from the previous generation with their parameter values perturbed by a kernel. Here the kernel is set to be a multivariate-normal approximation to the posterior, scaled by a selected size. A kernel size of 0.5 works well in most cases.
- **Individual max** – When one or more species use an individual-based model, this sets the maximum number of individuals allowed (exceeding this limit results in an error message). Such a threshold is introduced to stop BICI running out of memory when the individual number diverges. By default, this is set to 20,000 individuals. The text box can be used to make this limit higher, if required.
- **Parameter output** – If the number of elements in a tensor exceeds this value, results are not output. This limit is introduced to stop very large output files when the tensor is not of direct interest (for example it might avoid a spatial interaction matrix being output every MCMC iteration). By default, this is set to 1000, but this can be changed in the text box.
- **Set random seed** – Simulation and inference rely on pseudorandom number generators. These start with an initial value called a “seed”. This seed value leads to a given stochastic realisation from the model (so if the seed number is change, so too is the stochastic output). If the seed option is turned off (as it is by default), the seed is randomly generated by the BICI interface (in which case BICI will produce different results each time it is run). If the seed is set, however, BICI will consistently give the same results, despite being stochastic. The seed can take any value between 0 and 10,000.

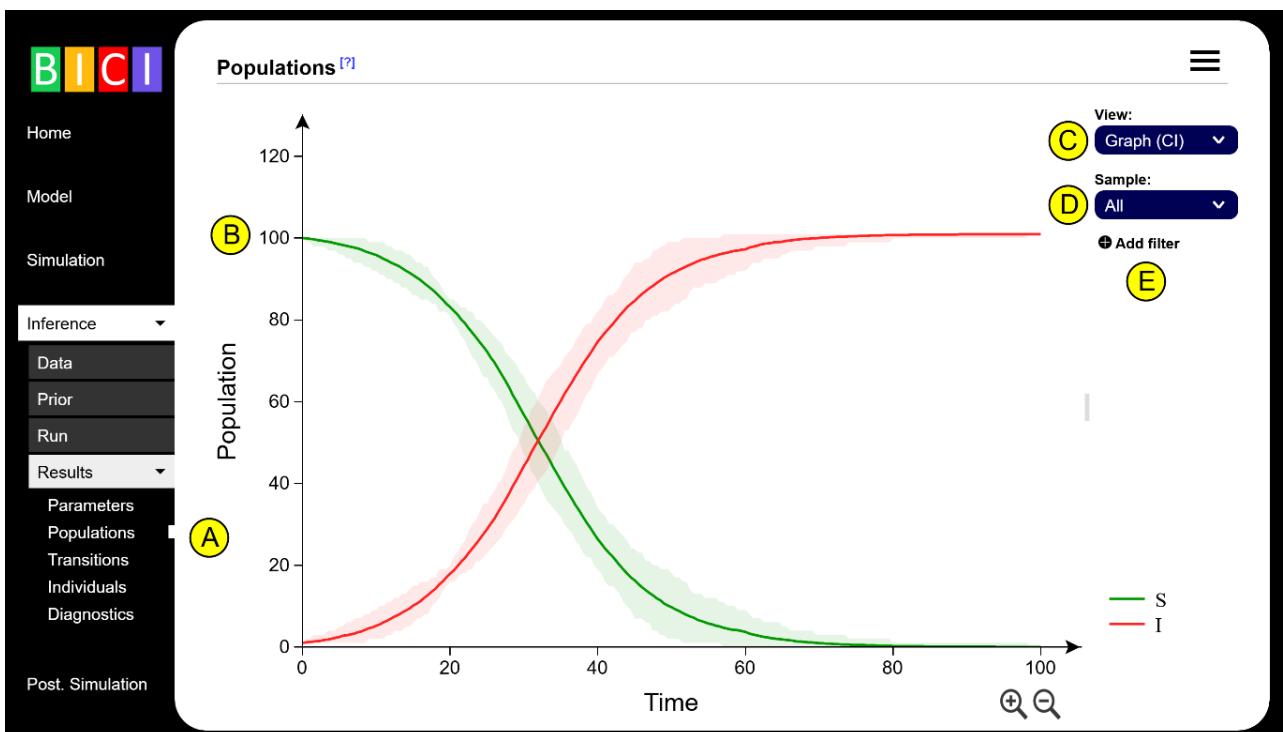


Figure 14 – Results from inference. A: Different types of output can be selected on the main menu, B: This shows an example of a posterior population distribution, C: Different ways to view the posterior can be selected, D: Individual samples can be selected, E: Filters can be added.

4.4 Results

Based on the data entered in §4.1 and prior knowledge from §4.2, it is generally not possible to identify model parameters with perfect precision (or transition events for that matter, unless they are specified by the data). Rather, there exists a distribution in these quantities known as the “posterior”, which expresses

both a best guess for parameters (*i.e.* posterior means) and range in values consistent with the data (*i.e.* credible intervals).

The “Inference→Results” page (Fig. 14) shows outputs to visualise this posterior (see Appendices C–F for examples). The menu bar (Fig. 14A) lists different possible features of the posterior to investigate, and these are discussed in the following sections:

4.4.1 Parameters

These plots provide visualisations and statistics for the parameter samples. Various options can be selected on the right-hand menu (depending on the situation):

- **View** – This list is divided into a number of different possibilities:
 - *Univariate* – These show results for all univariate parameters in the model.
 - *Multivariate* – Any parameters represented by vectors, matrices or tensors are given a separate section that can be viewed.
 - *Scatter* – Shows scatter plots of one parameter against another.
 - *Correlation* – Shows a matrix of correlation coefficients between different model parameters.
 - *Likelihoods* – Shows different likelihoods and the prior (all log transformed) which add up to give the posterior probability.
- **Graph** – Provides alternative ways to view the selected parameter:
 - *Trace* – Shows consecutive parameter samples drawn from an MCMC chain.
 - *Sample* – Provides raw samples for ABC approaches.
 - *Distribution* – Shows the estimated posterior distribution for a given parameter. This can either be viewed through binning or using kernel density estimation (KDE) [2] (selected using the settings button in the bottom left-hand corner). Under inference the Bayes factor can be calculated by clicking on the “BF” button (see §4.4.6) [3].
 - *Statistics* – For each parameter the data table provides the posterior mean, 95% credible interval as well as the effective sample size (ESS) and the Gelman-Rubin statistics (GR) (see §4.4.7 for details).

4.4.2 Populations

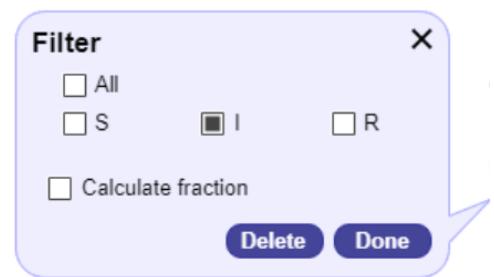
An example is shown in Fig. 14B, which shows the posterior distribution for the populations in different compartments for a selected classification as a function of time (means are shown by the solid lines and the shading represents 95% credible intervals).

Various options can be selected on the right-hand menu (depending on the situation):

- **View** (Fig. 13C) – Determines how the results are plotted:
 - *Graph (CI)* – Gives the population means, with shaded regions providing 95% credible intervals.
 - *Graph (line)* – Plots lines for each sample.
 - *Slice* – Provides posterior distributions at a specified time.

- *Compartment* – Plots the compartmental model, with colours representing the populations in different compartments (e.g. useful for looking at the spatial spread of disease).
- *Density* – Like ‘Compartment’, but expands the shading using Voronoi tessellation (e.g. good when plotting geographical points).
- *Data* – Allows for inferred populations to be compared to measured population data.
- **Sample** (Fig. 13D) – Allows for individual MCMC samples to be viewed.
- **Time** – Specifies the time used with the ‘Slice’ option.
- **Classification** – Select the classification being viewed.
- **Chain** – Select to filter results for a specific MCMC chain.

Population filters – The “⊕ Add filter” option (Fig. 14E) allows for filters to be added (e.g. to focus on individuals in a given disease state or at a specific location). Once a classification has been chosen, the checkboxes allow the user to specify only certain compartments (e.g. in the example on the right, only the infected ‘I’ individuals are selected).



Clicking on the “Calculate fraction” checkbox means results are shown by the ratio of the population in the select compartment(s) compared to the population in all of them (this is particularly useful for plotting quantities such as the disease prevalence).

4.4.3 Transitions

These plots show how the rates in different transitions vary as a function of time.

Various options can be selected on the right-hand menu (depending on the situation):

- **View** – Determines how the results are plotted:
 - *Graph (CI)* – Gives the rate means, with shaded regions providing 95% credible intervals.
 - *Graph (line)* – Plots lines for each sample.
 - *Data* – Allows for the inferred transition rate to be compared to measured population-level transition data.
- **Sample** – Allows for individual samples to be viewed.
- **Classification** – Select the classification being viewed.
- **Time-step** – Selects the time-step over which rates are estimated (larger values generate smoother graphs, but sacrifice temporal accuracy).
- **Chain** – Select to filter results for a specific MCMC chain.

The “⊕ Add filter” option allows for population filters to be added (see previous section).

4.4.4 Individuals

Various options can be selected on the right-hand menu (depending on the situation):

- **View** – Determines how the results are plotted:

- *Timeline* – Gives timelines for each individual in the system. These plots show how the compartmental status of different individuals in the system change as a function of time. The colours on the timelines correspond to those used for the compartments in the model.
- *Table* – Provides information about each individual (such as individual-level data, covariates for fixed effect or posterior estimates for individual effects).
- *Trans. Tree* – Shows the transmission tree. Note, this is only available when the transmission tree is turned on within the species (see §2.1.1).
- *Phylo. Tree* – Shows a phylogenetic tree. Note, this is only available when genetic data has been added to the analysis.
- **Sample** – Allows for individual samples to be viewed.
- **Classification** – Select the classification being viewed.
- **Chain** – Select to filter results for a specific MCMC chain.

Individual posterior plots – Under the ‘Timeline’ view, clicking on one of the ID names on the left-hand side allows for plots focussing on a specific individual’s posterior. These show time variation in the posterior probability of being in different compartments. Various options can be selected on the right-hand menu (depending on the situation):

- **View** – This is divided into a number of different possibilities:
 - *Graph* – Gives the mean posterior probability, with shaded regions providing 95% credible intervals.
 - *Compartment* – Plots the mean posterior probability using the compartmental model (here colour is used to represent compartments with high probability).
 - *Density* – Like ‘Compartment’, but expands the shading using Voronoi tessellation.
- **Sample** – Allows for individual samples to be viewed.
- **Classification** – Select the classification being viewed.
- **Chain** – Select to filter results for a specific MCMC chain.

4.4.5 Diagnostics

This provides a variety of visualisations to determine how well the model is fitting the data.

Various options can be selected on the right-hand menu (depending on the situation):

- **View** – Determines what is plotted:
 - *Trans. (exp.)* – Shows how transition rates in the posterior compare to rates estimated from the model. Systematic differences can help to pinpoint misspecification in the model.
 - *Trans. (dist.)* – Shows the cumulative probability distribution (see below) for a given transition. If the model is entirely correct, this distribution should be uniform between zero and one. Deviation from uniform indicates misspecification in the model. If the distribution is skewed to the right-hand-side it indicates that transitions have a faster rate

(or shorter duration) compared to what would be expected from the model, and *vice-versa*.

- *Trans. (bias)* – Shows the degree of skewness in the cumulative probability distribution, as measured by the first moment. A positive value implies that the transition rate in the posterior is in excess of the rate predicted by the model, and *vice-versa*.
- *Trans. (p-val)* – Deviations from uniform for the cumulative probability distribution can indicate model misspecification. Such deviations do not necessarily affect the skewness (*e.g.* fitting an exponential distribution which in reality is gamma distributed) and so an alternative measure is provided. Here the distribution is split into 10 equal divisions, and a p-value is calculated using Pearson's chi-squared test. This plot shows the posterior distribution in -log(p-value), where large values indicate significant deviation from the model (the dashed line represents a notional p-value of 0.05).
- *Proposals* – Provided information about MCMC proposals and other diagnostic information.
- **Sample** – Allows for individual samples to be viewed.
- **Time-step** – Selects the time-step over which rates are estimated (larger values generate smoother graphs, but sacrifice temporal accuracy).
- **Classification** – Select the classification being viewed.
- **Chain** – Select to filter results for a specific MCMC chain.

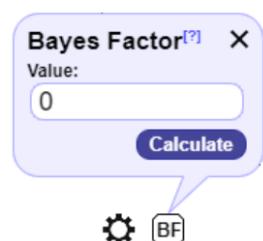
The “ Add filter” option allows for population filters to be added (see §4.4.2).

Cumulative probability — The calculation of this quantity depends on the type of model:

- **Individual-based** – Here each individual transition is sampled from a distribution. The cumulative probability represents the probability of sampling a time later than the realised transition time.
- **Population-based** – The realised number of transitions within a timestep is sampled from a Poisson distribution with a given mean. In this case, the cumulative probability is the probability of obtaining this realised number or fewer.

4.4.6 Estimating the Bayes factor

A Bayes factor (BF) is the ratio of the likelihood for one particular hypothesis to the likelihood for another [3]. The BF comparing the full model to one in which a particular parameter is fixed (usually to zero, such that the parameter is effectively removed from the model) can be calculated using the “BF” button under a posterior parameter distribution (see §4.4.1). This is one way to determine which parameters in the model are redundant, so allowing for the model to be simplified in a step-wise fashion. A BF between 3 and 10 represents moderate evidence for one model over another, and exceeding 10 is considered strong [4].



4.4.7 MCMC diagnostics

Diagnostics provide an objective way of determining if a sufficient number of MCMC updates have been iterated to reliably trust the results (if this is not the case, the number of updates in Fig. 13D must be increased and the analysis re-run).

Two different measures are shown on the parameter statistics page (see §4.4.1):

- **Effective sample size (ESS)** – This takes into account the fact that successive MCMC samples are correlated, and provides an estimate for the effective number of independent draws from the posterior. An ESS exceeding 200 is considered indicative of good mixing. Note, the ESS is not guaranteed to monotonically increase as the number of MCMC updates goes up (in fact if it is less than 100 it often fluctuates wildly).
- **The Gelman-Rubin statistic \hat{R}** – On top of poor mixing, another difficulty faced by MCMC is multimodality. This characterises a situation in which one MCMC chain ends up in one posterior mode (and perhaps mixes well), but another goes to a completely different mode. One way to check for this problem is to calculate the Gelman–Rubin convergence diagnostic \hat{R} (otherwise known as the “potential scale reduction factor”) [5]. This is defined as the ratio of the overall pooled variance (*i.e.* calculated using samples taken from all chains combined) to the mean variance within each chain. Since MCMC is initialised randomly in parameters space, \hat{R} starts large and is expected to approach one after many MCMC updates. Values less than 1.05 are considered to be indicative of convergence. If, however, \hat{R} is larger than this threshold (even after a very large number of updates) it points to the existence of posterior multimodality or MCMC chains becoming stuck in metastable states. If the former, the results from BICI cannot be trusted. If the latter, introducing an anneal phase may help (see §4.3.1). Note, \hat{R} relies on comparing independent MCMC chains, and so cannot be calculated for a single chain.

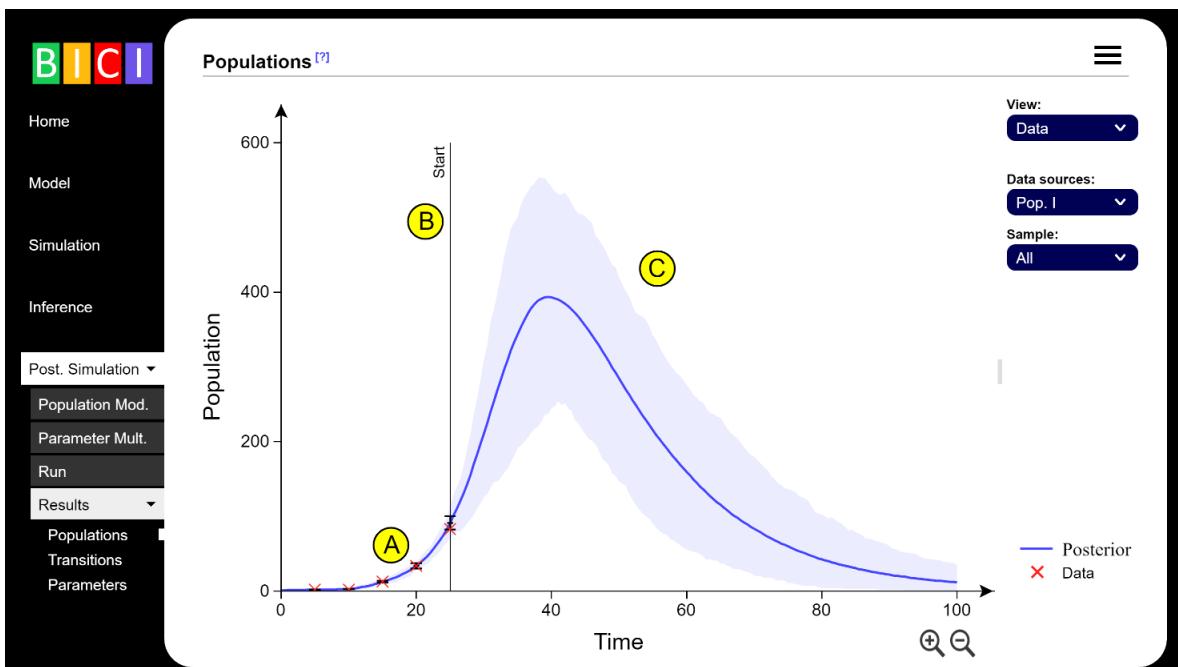


Figure 15 – Posterior simulation. This example shows future predictions. A: The red crosses represent hypothetical data from mid-way through an epidemic (this data gives the estimated number of infected individuals in the population), B: This vertical line represents the present time, C: This represents the predicted future dynamics of the epidemic (solid line represents the mean and the shading represents a 95% probability region).

5) Posterior simulation

So-called “posterior simulations”, simulate from the model whilst making use of the parameter and state samples generated under inference. They can be used in four different contexts:

- **Posterior predictive checking** – This is used to check if the model replicates the dynamics of the real system. It’s not a formal model fit diagnostic, but rather more of a sanity check to make sure the model is reasonable. Here the model is simulated using the posterior parameter samples and the output is compared against the data used for inference. If this data lies within the 95% probability region, it is indicative the model is a good one. If not, an alternative model may be sought.
- **Future prediction** – As above, but here the end time is extended such that future behaviour is estimated based on current data. An example of this is given in Fig. 15.
- **Scenario analysis** – As above, but with future parameters modified to see what effect this has on system dynamics.
- **Counterfactual analysis** – This takes all the data (*e.g.* after an epidemic) and asks the question, how would things have turned out differently if certain interventions had been made?

5.1 Population modification

The “Post. Simulation→Population Mod.” page allows for addition, removal and enforced movement of individuals on top of that used for the inference model (see §3.1 for details).

5.2 Parameter multipliers

The “Post. Simulation→Parameter Mult.” page allows for parameter “multipliers” to be added. A multiplier is a (potentially) time-varying factor that multiplies a specified parameter within the model. They are defined in the same way as splines (see §1.4.5 for an example).

5.3 Run

The “Post. Simulation→Run” page allows the user to set various options before starting a posterior simulation:

- **Time range** – Sets the period over which the posterior simulation is performed. How this is chosen is dependent on the analysis being performed:
 - *Posterior predictive checking* – This time range would be the same as for inference (this is the default setting).
 - *Future prediction and scenario analysis* – The start time would be set to the end time of inference and the end time would be extended to some timepoint in the future.
 - *Counterfactual analysis* – The start time would be set to the time of a proposed intervention and the end time would be the same as for inference.
- **Simulation number** – This sets how many posterior simulations are performed. Because simulations are stochastic, so each will provide a different set of results (by default, 200 are generated).

5.4 Results

The “Post. Simulation→Results” page shows results from posterior simulation. The outputs are essentially the same as for inference, so please refer to §4.4 for details (and see Appendices C-F for examples).

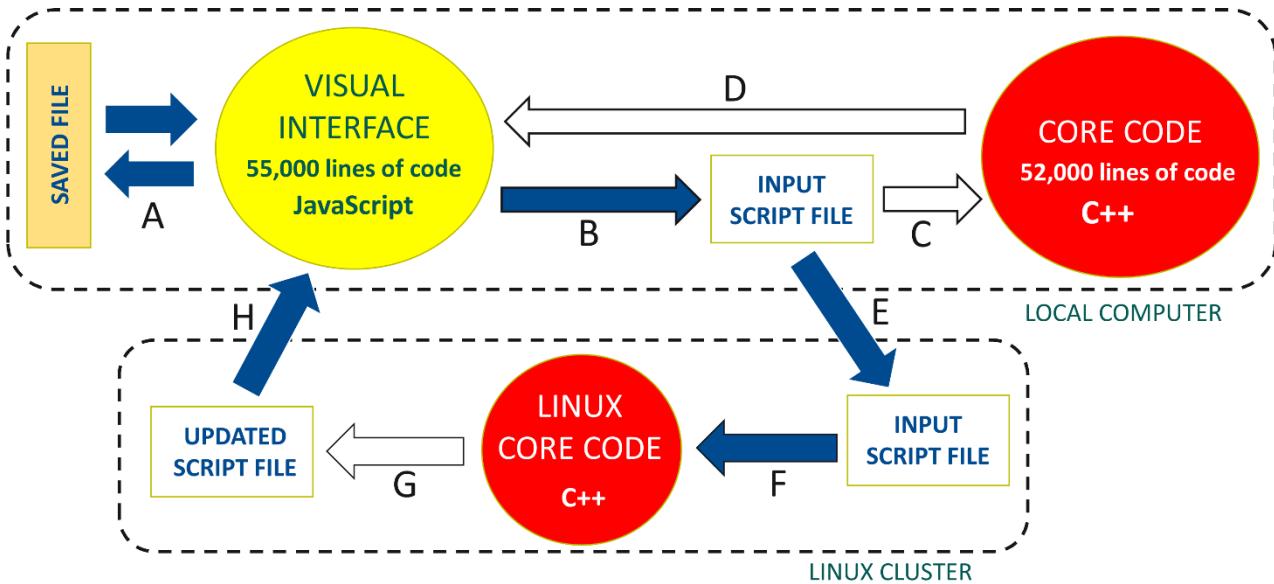


Figure 16 – BICI architecture. This diagram represents how BICI works. Arrows show interactions between different elements (solid/empty correspond to user initiated or not). A: Files can be loaded and saved from the interface in ‘.bici’ format, B: When the “Start” button is pressed the interface creates an input BICI-script file, C: When running on the local machine, this file gets loaded by the core code, which performs the analysis, D: The core code seamlessly passes the results back to the interface for visualisation, E: If the code is to be run on a Linux cluster, the script file is copied to the cluster, F: Execution is initiated using a terminal command (see §6.4), G: The core code is run and the results are embedded into the script file, H: The script file is copied back to the local machine and loaded into the interface for visualisation.

6) Inputs and outputs

6.1 BICI architecture

The diagram in Fig. 16 shows how BICI works. The code is actually split in two:

- **Visual interface** – This is written in JavaScript and runs on the desktop by means of NW.js.
- **The core code** – Performs analysis when BICI is executed (*i.e.*, by clicking on the “Start” button). This is written in highly efficient C++ code.

The various arrows in the diagram show routes by which the different elements interact.

6.2 Loading and saving BICI files

Loading and saving is done from the drop-down menu in the top right-hand corner of the page. The following file options are available:

- **Load** – This can be used to load a model/analysis in ‘.bici’ format.
- **Save** – Saves the model and data in ‘.bici’ format using the current path (note, this doesn’t save any results, so the file size is relatively small).
- **Save As...** – Saves the model and data in ‘.bici’ format using a specified path.



- **Save Simulation** – Saves simulation results (*i.e.* any parameter and state samples), along with the model/data used to generate them.
- **Save Inference** – Saves inference results, along with the model/data used to generate them (note, these files can be large because they store many samples of the model state).
- **Save Post. Sim.** – Saves posterior simulation results, along with the model/data used to generate them and inference samples.
- **View Code** – Used to view the code for the currently loaded model.

6.3 Exporting

Exporting from BICI is done from the drop-down menu in the top right-hand corner.

The following export options are available:

- **Script** – This is like “Save”, but in addition to a ‘.bici’ script file, a data folder is created that contains all the data (this makes the BICI-script more readable because it no longer contains all the data).
- **Print** – Prints the graph/plot/table currently being viewed.
- **Image** – Creates an image file for the graph/plot/table currently being viewed (in ‘.png’ format).
- **Table** – Creates a file for a table currently being viewed (in ‘.csv’ format).
- **Params** – Exports posterior parameter samples.
- **States** – Exports posterior state samples.

6.4 Running on a Linux cluster

On the “Simulation→Run”, “Inference→Run” and “Post. Simulation→Run” pages there is the option to “Run on local machine”. If this is selected as “No” it is assumed BICI is to be run on a Linux cluster.

To run BICI in this way, the following steps must be followed:

- **Create BICI file** – Click on the “Start” button (Fig. 13G) to create and save the BICI-script file. Copy this to the cluster where BICI is going to be run.
- **Executable** – The executable “bici-para” must also be copied from the BICI main folder to the cluster (*e.g.* this could be in the same directory as the BICI file).
- **Run** – BICI is run using:

`mpirun -n [core] ./bici-para [file.bici] [run]`

where:

- [core] is replaced with the number of CPU cores (this is typically the number of MCMC chains or PAS particles, but may be different if, *e.g.*, more than one chain is run per core).
- [file.bici] is replaced by the name of the BICI file that was created.
- [run] is replaced with “sim”, “inf” or “post-sim”, depending on what is being run.

Note, if the error “mpirun: command not found...” is encountered, MPI can be loaded with the terminal command “*module load mpi/openmpi-x86_64*”.

- **Visualise** – Once BICI has run, it embeds the results into [file.bici]. The user copies this back to their local computer and loads using the BICI interface.

```

#####
##### DESCRIPTION #####
description text="description.txt"

#####
##### DETAILS #####
data-dir folder="Examples\EX_M1-1-data-files"

simulation start=0 end=100 timestep=0.5

inference start=0 end=100 timestep=0.5

post-sim start=0 end=100

#####
##### DEFINE MODEL AND DATA FOR SPECIES PEOPLE #####
species name="People" type="population"

# SPECIES MODEL

class name="DS" index="a"

comp name="S" color="#009900" x=-16 y=-7
comp name="I" color="#ff2222" x=-2 y=-7

trans name="S->I" value="exp(rate: $\beta$ *{I})"

# SIMULATION INITIAL CONDITIONS

init-pop-sim type="fixed" focal="DS" file="init-pop-sim-People.csv"

#####
##### PARAMETERS #####
param name=" $\beta$ " value="0.001" prior="uniform(0,0.1)"

```

Figure 17 – Example of BICI-script. This example is taken from M1.1 in §8. A description of all BICI commands is given in Appendices A and B.

7) BICI-script

BICI-script is the means by which models and analyses are stored. Appendices A provides a list of BICI commands, ordered by section. Alternatively, Appendix B provides an alphabetically ordered index of commands for easy reference.

7.1 A simple example

Figure 17 shows an example of a simple BICI-script taken from M1.1 in §8. Different commands inform different parts of the model. This sets up a population-based species “People”, and creates a classification “DS”. It adds two compartments, ‘S’ and ‘I’, into this classification, and a transition going between the two with rate ‘ $\beta \times \{I\}$ ’.

For the most part BICI commands can be declared in any order. A notable exception to this is when the command relates to a specific species or classification. For example, “comp” adds a compartment to a classification, and so relies on that classification being declared earlier in the BICI-script²⁶.

The “data-dir” command in Fig. 17 informs the location of the data directory. This directory contains two files:

- “**description.txt**” - Provides a text description of the analysis:

```
# Model 1.1: Simple SI model

## Objective
- Introduce the simplest possible population-based epidemiological model.

## Model
- A single population-based species 'People' is created.
- This contains a classification 'DS' which stands for 'disease status'.
- DS contains two compartments: susceptible $$ and infected $I$. Together they are known as the "SI model".
- A transmission rate $\beta$ determines the rate at which individuals become infected.

## Population
- This consists of 100 initially susceptible individuals and one infected.
```

A markdown format is used (this allows for the following concise formatting: “# Title”, “## Subtitle”, “- Bullet point”, “*italic text*” and “**bold text**”). Additionally, parameters can be added by enclosing in dollar symbols, e.g. “\$a\$” or “\$b^super_sub\$”.

- “**init-pop-sim-People.csv**” – Contains information about the initial population under simulation:

```
"Compartment", "Population"
"S", 100
"I", 1
```

This states that the ‘S’ compartment starts with 100 individuals and the ‘I’ compartment starts with 1 individual.

For convenience, loading and saving BICI files is done using just a single ‘.bici’ file. This is achieved by embedding any additional data files, like the two above, within the script itself (by surrounded the file text using “[...]]” quotes).

7.2 Directly running BICI core code

Whilst running BICI is very easy using the interface, there may be instances when the user might want to run BICI script directly using the core code, *i.e.* bypassing the interface entirely. For example, if BICI was to be used as part of a pipeline in which the script is generated, run using the BICI core code, and the outputs then used for further processing.

The core code can be run using the following command:

²⁶ Note, the “set” command can be used to shift focus to a particular species or classification.

```
./bici_core.exe [file.bici] [run]
```

where:

- [file.bici] is the name of the BICI file.
- [run] is replaced with “sim”, “inf” or “post-sim”, depending on what is being run.

Once run, the outputs get appended to the ‘.bici file’. For example, if the script in Fig. 17 is run using:

```
./bici_core.exe example.bici sim
```

the following text will be added to the end of the file ‘example.bici’:

```
# OUTPUT

sim-param file="Sample/param_0.csv"

sim-state file="Sample/state_0.csv"
```

The command “sim-param” and “sim-state” store the parameters and states generated by the simulation (see sections below). Because the ‘data-dir’ command is defined in Fig.17, so BICI creates a directory “Sample” into which the parameter and state samples are saved. Otherwise, they are embedded directly into the BIC-script (see §7.1).

Note, running a new simulation deletes any existing “sim-param” and “sim-state” commands (and similarly for inference and posterior simulation).

```
State, "β", L^markov, L^non-markov, L^ie, L^dist, L^obs, L^genetic-proc, L^genetic-obs, L^init, Prior
0, 0.0216825, -21.6833, 0, 0, 0, -1096.93, 0, 0, 0, 2.30259
5, 0.00195176, -168.457, 0, 0, 0, -337.866, 0, 0, 0, 2.30259
10, 0.00195176, -260.835, 0, 0, 0, -186.026, 0, 0, 0, 2.30259
15, 0.00195176, -235.276, 0, 0, 0, -185.507, 0, 0, 0, 2.30259
20, 0.00167761, -238.172, 0, 0, 0, -127.469, 0, 0, 0, 2.30259
25, 0.00110069, -161.962, 0, 0, 0, -35.5761, 0, 0, 0, 2.30259
30, 0.00110069, -149.216, 0, 0, 0, -33.3597, 0, 0, 0, 2.30259
35, 0.00124248, -150.245, 0, 0, 0, -31.3309, 0, 0, 0, 2.30259
40, 0.0011392, -154.806, 0, 0, 0, -31.5356, 0, 0, 0, 2.30259
45, 0.00105984, -155.325, 0, 0, 0, -31.969, 0, 0, 0, 2.30259
50, 0.000734078, -158.384, 0, 0, 0, -31.4645, 0, 0, 0, 2.30259
55, 0.000916653, -158.112, 0, 0, 0, -34.5334, 0, 0, 0, 2.30259
60, 0.000968149, -148.057, 0, 0, 0, -31.9131, 0, 0, 0, 2.30259
65, 0.000871328, -152.784, 0, 0, 0, -32.0237, 0, 0, 0, 2.30259
70, 0.00109355, -148.759, 0, 0, 0, -31.3993, 0, 0, 0, 2.30259
75, 0.00109995, -150.393, 0, 0, 0, -30.7269, 0, 0, 0, 2.30259
```

Figure 18 – Parameter output file. This is an extract from of a parameter output file. The column “State” gives the MCMC update numbers from which the samples are taken (note, because of thinning this number goes up in steps of 5 in this example). Different columns provide posterior samples for different parameters, e.g. column 2 provides results for the parameter β . Other columns related to log-likelihoods and the log of the prior probability (see §7.3.1 for details).

7.3 BICI-script outputs

7.3.1 Parameter samples

These are stored in BICI-script using the commands “sim-param”, “inf-param” or “post-sim-param” (depending on whether simulation, inference or posterior simulation has been performed).

Figure 18 shows an example, taken from inference using M1.1 in §8. Here the first few posterior parameter samples generated are displayed (typically, the data table will contain 1000 rows in all).

```
{
    timepoint 0:0.5:100
}

<<STATE 1000>>

<PARAMETERS>
"β",0.00104521
L^markov,-144.692
L^non-markov,0
L^ie,0
L^dist,0
L^obs,-31.4348
L^init,0
Prior,2.30259

<SPECIES "People">

<INITIAL POPULATION>
compartment,population
S,100
I,1

<TRANSITIONS>
S->I:Z10,1,Z18,1,Z9,1,Z8,1,Z14,1,Z3,1,Z3,1,Z1,1,Z5,1,1,Z1,2,Z3,1,1,Z1,1,3,1,2,Z1,2,1,1,1,Z2,1,1,2...
```

Figure 19 – State output file (population-based). This is an extract from of a state output file. The curly brackets at the top denote header information (applicable to all samples) and the various tags specify different parts of the state sample. (see §7.3.2 for details).

```
{
    timepoint 0:0.5:100
    0:Sim-People-Ind-1
    1:Sim-People-Ind-2
    2:Sim-People-Ind-3
    :
    :
}
<<STATE 1000>>

<PARAMETERS>
"β",0.000838725
L^markov,-435.588
L^non-markov,0
L^ie,0
L^dist,0
L^obs,0
L^init,0
Prior,2.30259

<SPECIES "People">

<INDIVIDUALS>
name,source,events
0,no,S:0 S->I:85.1085
1,no,S:0 S->I:70.529
2,no,S:0 S->I:65.4514
:
```

Figure 20 – State output file (individual-based). This is an extract from of a state output file. The curly brackets at the top denote header information (applicable to all samples) and the various tags specify different parts of the state sample (see §7.3.2 for details).

7.3.2 State samples

These are stored in the BICI-script using the commands “sim-state”, “inf-state” or “post-sim-state” (depending on whether simulation, inference or posterior simulation have been performed).

The formatting for the state sample depends on whether it has been derived from a population or individual-based species:

Population-based sample – Figure 19 shows an example taken from inference using M1.1 in §8. The curly brackets at the top denote header information (which applies to all state samples). “timepoint” here indicates that inference time goes from 0 to 100 in steps of 0.5.

“<<STATE 1000>>” is used to denote the fact the sample was taken after 1000 MCMC updates.

The various tags in Fig. 19 provides information about the sample:

- <PARAMETERS> – This stores the parameter set.
- <SPECIES> – Indicates which species is being referred to.
- <INITIAL POPULATION> – Stores the initial population in the different model compartments.
- <TRANSITIONS> – For each transition, this stores the number of individuals passing down the transition for each time-step. Note, this uses concise formatting, where the “Z” character followed by a number indicates the number zeros represented (such an approach drastically reduces file size when transitions are sparse).
- <TRANSDISTPROB> – Provides information about transition probability distributions. This is used to inform diagnostic information (see §4.4.5).
- <DERIVED> – Stores samples for any time-varying derived parameters.

Individual-based sample – Figure 20 shows an example taken from inference using M1.2 in §8. The header provides a “dictionary” that relates individual names to numbers used in the state samples (this is done to reduce file size). Many of the tags are the same as above. However, rather than transition information being stored, here the dynamics of individuals are represented:

- <INDIVIDUALS> – This table provides information about whether the individuals originate from a source, their initial compartment and any events they undergo.
- <PHYLOTREE> - Provides information about the phylogenetic tree (if pathogen genetic data is used in the analysis).

As can be seen, storing the system state typically requires a much larger files size than storing the model parameters. For this reason, state sample are usually thinned more than parameter samples.

8) Examples

Example applications can be selected on the main page (see Fig. 1D). These demonstrate the versatility of BICI applied to a wide variety of different models and data scenarios. They can be altered or experimented with in any way, and new users are encouraged to try different possibilities as a way to familiarise themselves with the BICI interface (note, reloading examples from the main page returns them to their default settings). The source BICI-scripts for these examples can be found in the “Examples” directory.

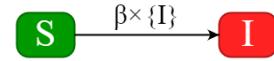
Sections M1- M6 go through different types of model. These examples can be simulated to see their dynamic behaviour (using different visualisations). Simulated data can be generated on the “Simulation→Generate Data” page (see §3.5), from which inference can then be performed.

M1 Simple epidemiological models

This section gives some examples of simple epidemiological models, illustrating both individual and population-based approaches.

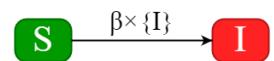
M1.1: SI population-based model (PBM)

- Purpose:**
- Introduce the simplest possible population-based epidemiological model.
 - This type of model keeps track of how the populations in the different compartments change over time.
- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains two compartments: susceptible S and infected I. Together they are known as the “SI model”.
 - A transmission rate β determines the rate at which individuals become infected.
- System:**
- 100 initially susceptible individuals and one infected.



M1.2: SI individual-based model (IBM)

- Purpose:**
- Introduce the simplest possible individual-based epidemiological model.
 - Under simulation or inference, we see timelines for each of the individuals in the system.
- Model:**
- A single individual-based species “People” is created.
 - This contains a classification “DS”, which stands for 'disease status'.
 - DS contains two compartments: susceptible S and infected I.
 - A transmission rate β determines the rate at which individuals become infected.
- System:**
- 100 initially susceptible individuals and one infected.



M1.3: SIR model (IBM)

- Purpose:**
- Introduce the SIR epidemiological model.



- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for 'disease status'.
 - DS contains three compartments: susceptible S, infected I and recovered R. Together they are known as the “SIR model”.
 - A transmission rate β determines the rate at which individuals become infected.
 - A recovery rate γ determines the rate at which individuals recover.
- System:**
- 100 initially susceptible individuals and one infected

M1.4: SIR model with Erlang distribution (PBM)

- Purpose:**
- Introduce the Erlang distribution.
 - In population-based models, transitions are normally required to be exponentially distributed.
 - The Erlang distribution is a gamma distribution with a specified integer shape parameter.
 - This can be used to more realistically model disease progression within individuals.
 - Note, Erlang transitions must be unbranching.
- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains three compartments: susceptible S, infected I and recovered R.
 - A transmission rate β determines the rate at which individuals become infected.
 - A recovery duration m determines how long it takes for individuals to recover on average.
 - The shape parameter is fixed to $k=3$.
- System:**
- 100 initially susceptible individuals and one infected.



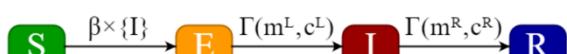
M1.5: SIR model with gamma distribution (IBM)

- Purpose:**
- Introduce non-Markovian (NM) transitions.
 - In individual-based models there is greater flexibility to specify the probability distribution over which transitions occur.
 - This example uses a gamma distributed infectious period with a mean and coefficient of variation (which is a dimensionless number that specifies the standard deviation divided by the mean).
 - Other possibilities include: Erlang, log-normal, Weibull, and fixed period.
 - NM transitions can be used to more realistically model disease progression within individuals.
- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains three compartments: susceptible S, infected I and recovered R.
 - A transmission rate β determines the rate at which individuals become infected.
 - A recovery duration m determines how long it takes for individuals to recover on average.
 - A coefficient of variation parameter c determines the variation about the mean m .
- System:**
- 100 initially susceptible individuals and one infected.



M1.6: SEIR model with exposed period (IBM)

- Purpose:**
- Introduce a model with a so-called “exposed” compartment.
 - This new compartment allows for a period of time after an individual becomes infected but before they are infectious.
- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains four compartments: susceptible S, infected I, exposed E and recovered R. Together they are known as the “SEIR model”.
 - A transmission rate β determines the rate at which individuals become infected.
 - A latent duration m^L (with coefficient of variation c^L) determines how long, on average, individuals remain in the exposed state.
 - An infectious duration m^I (with coefficient of variation c^I) determines how long, on average, individuals can pass on their infection to others.



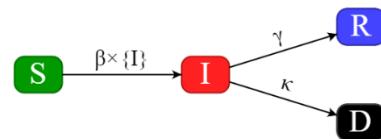
- System:**
- 100 initially susceptible individuals and two infected.

M2 Additional epidemiological models

These examples look at some additional features, such as branching, reverse transitions and stratification.

M2.1: SIRD model with branching using transition rates (PBM)

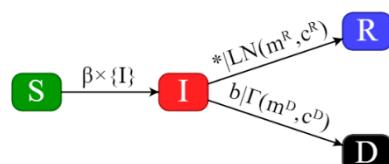
- Purpose:**
- Introduce branching in a population-based model.
 - Here infected individuals can either recover or die.



- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains four compartments: susceptible S, infected I, recovered R and dead D. Together they are known as the “SIRD model”.
 - A transmission rate β determines the rate at which individuals become infected.
 - A recovery rate γ determines the probability per unit time an infected individual recovers.
 - A mortality rate κ determines the probability per unit time an infected individual dies
- System:**
- 100 initially susceptible individuals and two infected.

M2.2: SIRD model with branching probability (IBM)

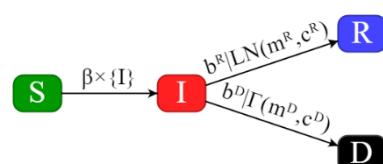
- Purpose:**
- Introduce branching in an individual-based model.
 - Here infected individuals can either recover or die.
 - A branching probability is used to determine if an individual dies.
 - Note, one of the branches had probability “*” because it is derived from others (such that the total adds to one).
 - In this example the transition distributions down the branches are different (one being log-normal and the other gamma distributed).



- Model:**
- A single individual-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains four compartments: susceptible S, infected I, recovered R and dead D.
 - A transmission rate β determines the rate at which individuals become infected.
 - Branching probability b determines the probability of dying.
 - Mean m^R and coefficient of variation c^R specify the log-normally distributed period to recovery.
 - Mean m^D and coefficient of variation c^D specify the gamma distributed period to death.
- System:**
- 100 initially susceptible individuals and two infected.

M2.3: SIRD model with branching factors (IBM)

- Purpose:**
- Introduce branching factors in an individual-based model.
 - Here infected individuals can either recover or die.
 - Unlike branching probabilities, branching factors specify the *relative* probability of going down a branch (note, their sum does not need to add to one).

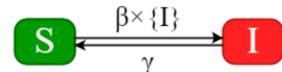


- They are implemented by clicking on the I compartment and selecting 'Use branching factors'.
- Branching factors are useful, e.g., for models in which time-varying covariates affect branching probabilities.
- The transition distributions down the branches are different (one being log-normal and the other gamma distributed).

- Model:**
- A single individual-based species "People" is created.
 - This contains a classification "DS", which stands for "disease status".
 - DS contains four compartments: susceptible S, infected I, recovered R and dead D.
 - A transmission rate β determines the rate at which individuals become infected.
 - Branching factors b^D and b^R determine the relative probability of dying or recovering.
 - The probability of dying is $b^D/(b^D + b^R)$.
 - Mean m^R and coefficient of variation c^R specify the log-normally distributed period to recovery.
 - Mean m^D and coefficient of variation c^D specify the gamma distributed period to death.
- System:**
- 100 initially susceptible individuals and two infected.

M2.4: SIS model (PBM)

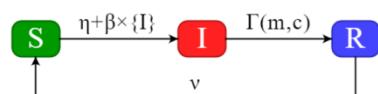
- Purpose:**
- Introduce a model that allows individuals to be infected multiple times.
 - This can lead to endemic diseases.



- Model:**
- A single population-based species "People" is created.
 - This contains a classification "DS", which stands for "disease status".
 - DS contains two compartments: susceptible S and infected I.
 - The backward transition to the I state means that this model is known as the "SIS model".
 - A transmission rate β determines the rate at which individuals become infected.
 - A recovery rate γ determines the rate at which individual return to being susceptible.
- System:**
- 100 initially susceptible individuals and one infected.

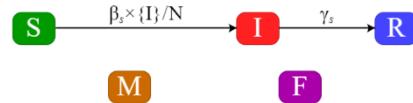
M2.5: SIRS model with waning immunity (IBM)

- Purpose:**
- Introduce the concept of waning immunity.
 - For many diseases recovery is not permanent.
 - Due to mutations in the pathogen, an individual can become reinfected at a later date.
 - This waning immunity is incorporated into the standard SIR model by a backward transition from R to S.
 - Such a model can generate multiple epidemics.
- Model:**
- A single population-based species "People" is created.
 - This contains a classification "DS", which stands for "disease status".
 - DS contains three compartments: susceptible S, infected I and recovered R.
 - An external force of infection parameter η provides a small proportion of infections from outside the system.
 - A transmission rate β determines the rate at which individuals become infected.
 - A recovery duration m (and coefficient of variation c) determines how long, on average, it takes for individuals to recover.
 - A waning immunity coefficient v determines the rate at which individuals become susceptible again.
- System:**
- 100 initially susceptible individuals and two infected.



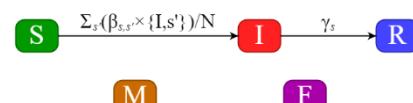
M2.6: SIR model with demographic stratification (PBM)

- Purpose:**
- Introduce an example of stratification into the SIR epidemiological model.
 - Here the population is divided into males and females.
- Model:**
- A single population-based species “People” is created.
 - This contains classifications “DS”, which stands for “disease status”, and “Sex”.
 - “DS” contains three compartments: susceptible S, infected I and recovered R.
 - “Sex” contains the compartments M and F for male and female.
 - A transmission rate β_s determines the rate at which individuals with sex s become infected.
 - A recovery rate γ_s determines the rate at which individuals recover with sex s .
 - N is the total number of individuals (this model assumes frequency dependent transmission).
- System:**
- 10,000 initially susceptible individuals and 2 infected.
 - A 50/50 split between males and females.



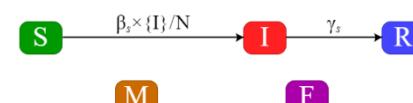
M2.7: SIR model with differential infectivity and demographic stratification (PBM)

- Purpose:**
- Introduce differences in infectivity in a stratified SIR epidemiological model.
 - Here the population is divided into males and females.
 - A 2x2 transmission matrix is used to describe transmission between these two demographic groups.
- Model:**
- A single population-based species “People” is created.
 - This contains classifications “DS”, which stands for “disease status”, and “Sex”.
 - “DS” contains three compartments: susceptible S, infected I and recovered R.
 - “Sex” contains the compartments M and F for male and female.
 - A transmission rate $\beta_{s,s'}$ determines the rate at which individuals with sex s become infected from individuals with sex s' .
 - A recovery rate γ_s determines the rate at which individuals recover with sex s .
 - N is the total number of individuals (this model assumes frequency dependent transmission).
- System:**
- 10,000 initially susceptible individuals and 2 infected.
 - A 50/50 split between males and females.



M2.8: SIR model with demographic stratification (IBM)

- Purpose:**
- The same as M2.6, but this uses an individual-based model.
 - Introduce an example of stratification into the SIR epidemiological model.
 - Here the population is divided into males and females.
- Model:**
- A single population-based species “People” is created.
 - This contains classifications “DS”, which stands for “disease status”, and “Sex”.
 - “DS” contains three compartments: susceptible S, infected I and recovered R.
 - “Sex” contains the compartments M and F for male and female.



- A transmission rate β_s determines the rate at which individuals with sex s become infected.
- A recovery rate γ_s determines the rate at which individuals recover with sex s .
- N is the total number of individuals (this model assumes frequency dependent transmission).

System:

- 10,000 initially susceptible individuals and 2 infected.
- A 50/50 split between males and females.

M3 Spatial epidemiological models

These models incorporate spatial stratification in different ways.

M3.1: Metapopulation model using geographical regions (PBM)

Purpose:

- Introduce a spatial epidemiological model that uses regions (these are bounded areas on a map).
- These boundaries are loaded using GeoJSON format (a commonly used format, often available to download for geographical, political and administrative boundaries).
- Here Scotland is divided into 32 local authorities.
- A constant matrix of interactions is derived from census commuting data.



Model:

- A single population-based species “People” is created.
- This contains classifications “DS”, which stands for “disease status”, and “Location”.
- “DS” contains three compartments: susceptible S, infected I and recovered R.
- “Location” contains 32 regions.
- A transmission rate β determines the rate at which individuals transmit disease.
- A constant matrix of interactions $M_{l,l}$ is proportional to the contact rate between individuals in the same and different regions (this can be estimated from census commuting data).
- A recovery rate γ determines the rate at which individuals recover.
- An external force of infection η is used to initiate an epidemic.

System:

- Initially susceptible individuals, with regional populations taken from census data.

M3.2: Metapopulation model using geographical points (PBM)

Purpose:

- Introduce a spatial epidemiological model that uses geographical points (these have a defined longitude and latitude).
- The 100 largest cities in the world are incorporated into the model.



Model:

- A single population-based species “People” is created.
- This contains classifications “DS”, which stands for “disease status”, and “Location”.
- “DS” contains three compartments: susceptible S, infected I and recovered R.
- “Location” contains compartments for 100 world cities.
- A transmission rate β determines the rate at which individuals transmit disease within a city.
- Parameter λ controls the spread of disease between cities.
- A recovery rate γ determines the rate at which individuals recover.

- Constant matrix $M_{l,l'}$ is proportional to the interactions between cities (e.g., this can be obtained from movement data).
 - N_l is the population of city l .
 - P is the global population.
- System:**
- Initially susceptible individuals, with city populations taken from census data.
 - Two infected individuals start in Tokyo.

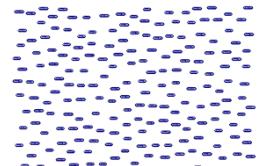
M3.3: Metapopulation model using a distance kernel (PBM)

- Purpose:**
- Introduce spatial kernels.
 - A kernel is a specified function that gives the probability the disease is transmitted a certain distance.
 - Typically, they have a high value at close distance, but decay away to zero in the large distance limit.
 - This example uses a so-called “power-law” spatial kernel.
 - The 100 largest cities in the UK are incorporated into the model.
- Model:**
- A single population-based species “People” is created.
 - This contains classifications “DS”, which stands for “disease status”, and “Location”.
 - “DS” contains three compartments: susceptible S, infected I and recovered R.
 - “Location” contains 100 geographical points.
 - A transmission rate β determines the rate at which individuals transmit disease within a city.
 - The parameter λ determines the rate at which individuals transmit disease between cities.
 - η is an external force of infection.
 - A recovery rate γ determines how fast individuals recover.
 - A matrix $M_{l,l'}$ gives the transmission between cities. This is reparameterised as a power-law distribution, where Δ gives the range over which transmission occurs and α is the exponent (larger values imply a quicker decay with distance).
 - N_l is the population in city l .
 - P is the UK population.
- System:**
- Initially susceptible individuals, with city populations taken from census data.



M3.4: Farm-based model using a distance kernel (IBM)

- Purpose:**
- Introduce farm-based models.
 - In these models farms are represented as geographical points (either as compartments with x,y positions, or using longitudes and latitudes).
 - Each farm contains a single “individual” that is used to denote the disease status of that farm.
 - A spatial kernel allows for spread of disease between farms.
- Model:**
- A single individual-based species “Farm” is created.
 - This contains classifications “DS”, which stands for “disease status”, and “Location”.
 - “DS” contains three compartments: susceptible farm S, infected farm I and recovered farm R.
 - “Location” contains 200 farm positions.
 - A transmission rate β determines the rate at which susceptible farms become infected due to surrounding infected farms.



- A recovery rate γ determines how quickly farms recover.

System: • A single individual is added to each farm location. All farms are initially susceptible except for one infected.

M4 Ecological models

Compartmental models are often applied to epidemiological applications, but they can equally be used in other settings. Here we look at some examples from ecology.

M4.1: Logistic growth population model (IBM)

Purpose:

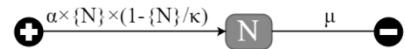
- Introduce a simple model to represent the dynamics of population in a wildlife setting.
- This uses a logistic growth model.
- Initially the rate at which individuals are born is proportional to the number of individuals currently present.
- Finite resources limit the size of the population through a carrying capacity κ .

Model:

- A single population-based species “Animal” is created.
- This contains a classification “Population”.
- Population contains a compartment N , which represents the number of individuals.
- A birth rate α determines how fast individuals are born.
- A mortality rate μ determines how fast individuals die.

System:

- The initial state starts with just a single individual.



M4.2: A predator-prey model (PBM)

Purpose:

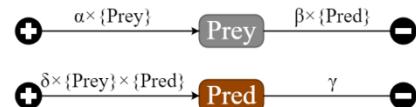
- Introduce a simple model to demonstrate predator-prey dynamics.
- This uses the well-known Lotka-Volterra equations.
- The birth rate of prey is proportional to their population.
- Prey die at a rate proportional to the number of predators.
- The birth rate of predators is proportional to their population as well as the population of prey (because predators rely on prey for food).
- Predators die at a constant rate.

Model:

- A single population-based species “Animal” is created.
- This contains a classification “Population”.
- Population contains two compartments: “Prey” and “Pred”.
- A birth rate α determines how fast prey are born.
- A mortality rate β determines how fast prey die.
- A birth rate δ determines how fast predators are born.
- A mortality rate γ determines how fast predators die.

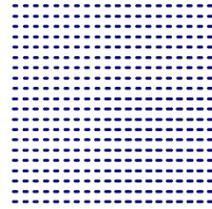
System:

- This initial state starts with 100 predators and 100 prey.



M4.3 A spatial diffusion model (IBM)

- Purpose:**
- Introduce a model in which a landscape is divided into a grid of cells.
 - Individuals move across this landscape by randomly hopping from one grid cell to a neighbouring grid cell.



- Model:**
- A single individual-based species 'Animal' is created.
 - This contains a classification "Location".
 - "Location" contains a 20x20 grid of cells represented by compartments.
 - A hopping rate ω determines the rate at which individuals move between cells.
- System:**
- 100 individuals initially in cell G10-10.

M4.4: A species presence/absence distribution model (IBM)

- Purpose:**
- Introduce a simple presence/absence model for an invasive plant species.
 - A map of the UK is split into a grid of 25km squares.
 - A covariate is used that affects the probability the species will invade a given square (e.g. it could represent some environmental variable such as altitude or mean temperature).



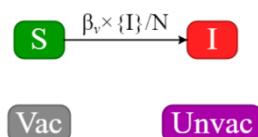
- Model:**
- A single individual-based species "Plant" is created.
 - This contains classifications "Presence" and "Location".
 - "Presence" contains two compartments: "No" and "Yes".
 - "Location" contains compartments representing the grid cells (559 in total).
 - Each grid cell contains a single individual that represents the cell's status.
 - A transmission rate β determines the rate at which cells start growing the plant.
 - A fixed effect $\langle g \rangle$ is used to add a covariate affecting plant spread.
 - A spatial power-law transmission kernel is used to model the geographical spread of the plant.
- System:**
- Initially, a single grid cell is set to contain the invasive plant species.

M5 Disease transmission experiments

Disease transmission experiments can be used to identify factors affecting the passing of infection from one individual to another. These can either be fixed categorical effects, such as vaccination status, covariates, such as weight, or quantitative genetic traits.

M5.1: Single contact group, investigating susceptibility (IBM)

- Purpose:**
- Introduce the concept of a disease transmission experiment.
 - These are used to identify factors which affects how readily disease transmits between individuals.
 - This example considers the effect of vaccination on susceptibility to disease.



- Model:**
- A single individual-based species "Animal" is created.
 - This contains classifications "DS", which stands for "disease status", and "Vaccination".
 - "DS" contains two compartments: susceptible S and infected I.

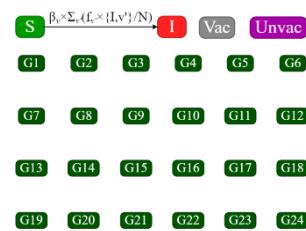
- “Vaccination” contains two compartments: “Vac” and “Unvac”, which represent vaccinated and unvaccinated individuals.
- A transmission rate β_v determines the rate at which individuals with vaccination status v become infected.

System: • 100 initially susceptible individuals and one infected.

M5.2: Multiple contact groups, investigating infectivity (IBM)

Purpose:

- Introduce disease transmission experiment for quantifying differences in infectivity.
- For this it is necessary to combine information from multiple contact groups.
- In this example we consider the effect of vaccination on susceptibility and infectivity.



Model:

- A single individual-based species 'Animal' is created.
- This contains classifications “DS”, which stands for “disease status”, “Vaccination” and “Group”.
- “DS” contains two compartments: susceptible S and infected I.
- “Vaccination” contains two compartments: “Vac” and “Unvac”, which represent vaccinated and unvaccinated individuals.
- “Group” contains 24 contact groups.
- A transmission rate β_v determines the susceptibility of individuals with vaccination status v .
- A factor f_v determines the relative infectiousness of individuals with vaccination status v .

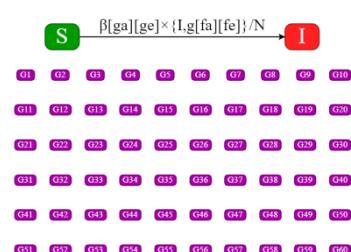
System:

- Each group consists of 10 individuals, half of which are initially infected.
- The vaccination status of the infected and susceptibility individuals is permuted to create blocks of 4 that are replicated 6 times (to make 24 groups in total).

M5.3: Quantitative genetics model for susceptibility/infectivity (IBM)

Purpose:

- Introduce disease transmission experiment for estimating quantitative genetic contributions to susceptibility and infectivity.
- A population of full-sib families is randomly allocated into contact groups.



Model:

- A single individual-based species “Animal” is created.
- This contains classifications “DS” and “Group”.
- “DS” contains two compartments: susceptible S and infected I.
- “Group” contains 60 contact groups.
- A transmission rate β determines how fast individuals become infected.
- Individual effects [ga] (for genetic contribution) and [ge] (for environmental contribution) characterise individual variation in susceptibility.
- Individual effects [fa] (for genetic contribution) and [fe] (for environmental contribution) characterise individual variation in infectivity.
- $\Omega^{fa,fa}$ and $\Omega^{ga,ga}$ give genetic variances.
- $\omega^{fa,ga}$ gives genetic correlation between traits.
- $\Omega^{fe,fe}$ and $\Omega^{ge,ge}$ give environmental variances.

- $\omega^{fe,ge}$ gives environmental correlation between traits.

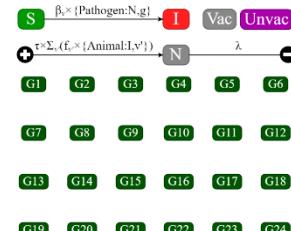
System:

- Each contact group consists of 10 individuals, two of which have been randomly infected.
- The population is made up of 60 sires who each have 10 progeny. The 600 progeny overall are randomly allocated across the contact groups.

M5.4: Environmental pathogen accumulation model (IBM/PBM)

Purpose:

- Rather than modelling direct transmission of disease between individuals, here we assume that pathogen accumulates in the environment.
- A disease transmission experiment with multiple contact groups is assumed.



Model:

- An individual-based species "Animal" is created.
- This contains classifications "DS", which stands for "disease status", "Vaccination" and "Group".
- "DS" contains two compartments: susceptible S and infected I.
- "Vaccination" contains two compartments: "Vac" and "Unvac", which represent vaccinated and unvaccinated individuals.
- "Group" contains 24 contact groups.
- A population-based species "Pathogen" models the environmental accumulation.
- This contains classifications "Population", which stores the amount of pathogen, and "Group", which is cloned from 'Animal'.
- A transmission rate β_v determines the susceptibility of individuals with vaccination status v .
- A factor f_v determines the relative infectiousness of individuals with vaccination status v .
- τ determines the pathogen shedding rate from infected individuals.
- λ determines the decay of the pathogen in the environment.

System:

- Each group consists of 10 individuals, half of which are initially infected.
- The vaccination status of the infected and susceptibility individuals is permuted to create blocks of 4 that are replicated 6 times (to make 24 groups in total).

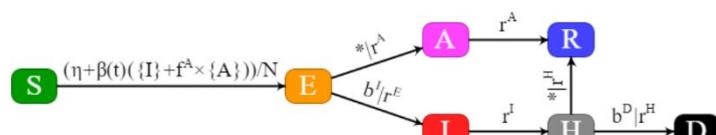
M6 COVID-19 models

These examples look at a variety of different models aimed at understanding real-world epidemic outbreaks.

M6.1: Simple (PBM)

Purpose:

- Show a possible model of COVID-19.
- Here additional compartments are used to not only more accurately model the disease, but such that the model aligns with publicly available data.



Model:

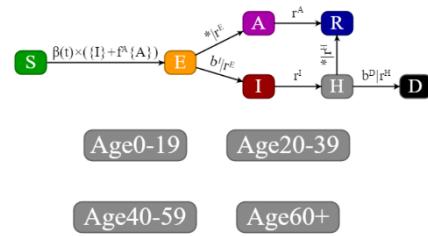
- A single population-based species "People" is created.
- This contains a classification "DS", which stands for "disease status".
- DS contains seven compartments: susceptible S, exposed E, infected I, recovered R, asymptomatic A, hospitalised H and dead D.

- An external force of infection is controlled by parameter η (in reality this could be a spline estimated from raw case data and flight information).
- The transmission rate $\beta(t)$ is time varying to capture the influence of lockdown measures and vaccination.
- Parameter f^A determines the relative infectiousness of asymptomatic A individuals (compared with infected I individuals).

System: • The UK population, which is initially susceptible.

M6.2: Age-structured model (PBM)

- Purpose:** • This adds age stratification into model 6.1.
- Model:** • A single population-based species “People” is created.
 • This contains classifications “DS”, which stands for “disease status”, and “Age”.
 • “DS” contains seven compartments:
 susceptible S, exposed E, infected I, recovered R, asymptomatic A, hospitalised H and dead D.
 • “Age” contains 4 age classifications: 0-19, 20-39, 40-59 and 60+.
 • An external force of infection is controlled by parameter η (in reality this could be a spline estimated from raw case data and flight information).
 • The transmission rate $\beta(t)$ is time varying to capture the influence of lockdown measures and vaccination.
 • An age mixing $M_{a,a'}$ captures the relative contact rate between different age groups.
- System:** • The UK population, which is initially susceptible.



The following illustrate feature of simulation, inference or posterior simulation:

A Simulation features

These examples look at various simulation features.

A1: Multiple simulations (PBM, M1.1)

- Purpose:** • On the “Simulation→Run” page the number of simulations is set to 100 (instead of the usual 1).
 • When run, the simulated outputs exhibit a range of system dynamics reflecting inherent stochasticity within the system.
 • Shaded regions in the population plots show the envelope containing 95% of simulations.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains two compartments: susceptible S and infected I.
 • A transmission rate β determines the rate at which individuals become infected.
- System:** • 100 initially susceptible individuals and one infected.

A2: Uncertain initial conditions — single classification (PBM, M1.1)

- Purpose:** • Introduce uncertainty in the initial population.
 • This considers the case of a model with just a single classification.

- On the 'Simulation→Initial Conditions' page the initial population is set as a distribution (rather than fixed).

• 100 simulations are performed to see stochastic variation.

Model:

- A single population-based species "People" is created.
- This contains a classification "DS", which stands for "disease status".
- DS contains two compartments: susceptible S and infected I.
- A transmission rate β determines the rate at which individuals become infected.
- The initial population in the S compartment is sampled uniformly in the range between 900 and 1100.
- The initial population in the I compartment is sampled uniformly in the range between 0 and 50.

A3: Uncertain initial conditions — multiple classifications — focal selected (PBM, M2.6)

Purpose:

- Introduce specification of uncertainty in the initial population when there is more than one classification.
- The initial population is given on the "Simulation→Initial Conditions" page.
- Here the disease status is a specified "focal" classification.
- Distributions for the initial populations in focal compartments are set (in this case chosen to be uniform distributions).
- For the other classification ("Sex") a Dirichlet distribution is used to sample the *fraction* of individuals in each compartment.
- 100 simulations are performed to see stochastic variation.

Model:

- A single population-based species "People" is created.
- This contains a classification "DS", which stands for "disease status", and a classification "Sex".
- "DS" contains three compartments: susceptible S, infected I and recovered R.
- "Sex" contains the compartments M and F for male and female.
- A transmission rate β_s determines the rate at which individuals with sex s become infected.
- A recovery rate γ_s determines the rate at which individuals recover with sex s .
- N is the total number of individuals (this model assumes frequency dependent transmission).

System:

- This consists of initially S uniformly sampled between 9000 and 11000, I uniformly sampled between 0 and 100 and R uniformly sampled between 0 and 100.
- There is a 30/70 split between males and females, as determined from the Dirichlet distribution.

A4: Uncertain initial conditions — multiple classifications — total population selected (PBM, M2.6)

Purpose:

- Introduce specification of uncertainty in the initial population when there is more than one classification.
- The initial population is given on the 'Simulation→Population' page.
- In this case a prior is placed on the total population.
- A Dirichlet distribution is placed on all compartmental combinations.
- 100 simulations are performed to see stochastic variation.

Model:

- A single population-based species "People" is created.
- This contains classifications "DS", which stands for "disease status", and "Sex".
- "DS" contains three compartments: susceptible S, infected I and recovered R.
- "Sex" contains the compartments M and F for male and female.
- A transmission rate β_s determines the rate at which individuals with sex s become infected.

- A recovery rate γ_s determines the rate at which individuals recover with sex s .
 - N is the total number of individuals (this model assumes frequency dependent transmission).
- System:**
- The initial population is sampled from a uniform distribution between 9000 and 11000.
 - A Dirichlet distribution is used to sample the compartmental composition of the initial state (with most individuals in the S compartment).

A5: Add / remove individuals (PBM, M2.6)

- Purpose:**
- Introduce adding and/or removing populations after the initial conditions are set.
 - Relevant for a population-based model.
 - The 'Simulation→Initial Conditions' page show how the additions and removals are defined.
- Model:**
- A single population-based species "People" is created.
 - This contains classifications "DS", which stands for "disease status", and "Sex".
 - "DS" contains three compartments: susceptible S, infected I and recovered R.
 - "Sex" contains the compartments M and F for male and female.
 - A transmission rate β_s determines the rate at which individuals with sex s become infected.
 - A recovery rate γ_s determines the rate at which individuals recover with sex s .
 - N is the total number of individuals (this model assumes frequency dependent transmission).
- System:**
- 10,000 initially susceptible individuals and 2 infected.
 - A 50/50 split between males and females.

A6: Add / move / remove individuals (IBM)

- Purpose:**
- To show how individuals can be added, moved and removed from the system.
 - Relevant for an individual-based model.
 - 100 susceptible individuals are added at time $t=0$.
 - Here we have an SIR model within two fish tanks: T1 and T2.
 - 10 infected individuals are added to tank T1.
 - At time $t=100$, 10 specified individuals are moved from tank T1 to tank T2. This then initiates an epidemic in tank T2.
- Model:**
- A single population-based species "People" is created.
 - This contains classifications "DS", which stands for "disease status", and "Tank".
 - "DS" contains two compartments: susceptible S and infected I.
 - "Tank" contains two fish tank compartments: T1 and T2.
 - A transmission rate β determines the rate at which individuals become infected.
- System:**
- 100 initially susceptible individuals and one infected.

B) Population-level data types

These examples look at different types of population-level data.

B1: Time series population observations (PBM, M1.1)

- Purpose:**
- Inference is done using time-series population data.
- Data:**
- On the 'Inference→Data' page, data is added that gives a time-series estimate for the number of individuals in the I compartment.
- Model:**
- A single population-based species "People" is created.
 - This contains a classification "DS", which stands for "disease status".

- DS contains two compartments: susceptible S and infected I.
- A transmission rate β determines the rate at which individuals become infected.

System: • 100 initially susceptible individuals and one infected.

B2: Time series population-level transition observations (PBM, M1.1)

Purpose: • Inference is done using time-series population-level transition data.

Data: • On the 'Inference→Data' page, population-level transition data is added.
• This gives time-series estimates for the number infection transitions at periodic intervals with timestep $\Delta t=7$.

Model: • A single population-based species "People" is created.
• This contains a classification "DS", which stands for "disease status".
• DS contains two compartments: susceptible S and infected I.
• A transmission rate β determines the rate at which individuals become infected.

System: • 100 initially susceptible individuals and one infected.

B3: Stratified time series population observations (PBM, M2.6)

Purpose: • Introduce inference on stratified population data.

Data: • On the "Inference→Data" page, data is added which gives time series estimates for the populations in each the compartmental combinations.

Model: • A single population-based species "People" is created.
• This contains classifications "DS", which stands for "disease status", and "Sex".
• "DS" contains three compartments: susceptible S, infected I and recovered R.
• "Sex" contains the compartments M and F for males and females.
• A transmission rate β_s determines the rate at which individuals with sex s become infected.
• A recovery rate γ_s determines the rate at which individuals recover with sex s .
• N is the total number of individuals (this model assumes frequency dependent transmission).

System: • 10,000 initially susceptible individuals and 2 infected.
• A 50/50 split between males and females.

B4: Stratified population observations from multiple compartments (PBM, M2.6)

Purpose: • Introduce inference on multiple population time series that target different combinations of compartments.

Data: • Stratified population data is added on the 'Inference→Data' page.
• One time series provides stratified observations on the infected population.
• A second time series provides overall observations for individuals in either compartments I or R. For example, this could represent a serological test.

Model: • A single population-based species "People" is created.
• This contains classifications "DS", which stands for "disease status", and "Sex".
• "DS" contains three compartments: susceptible S, infected I and recovered R.
• "Sex" contains the compartments M and F for males and females.
• A transmission rate β_s determines the rate at which individuals with sex s become infected.
• A recovery rate γ_s determines the rate at which individuals recover with sex s .
• N is the total number of individuals (this model assumes frequency dependent transmission).

System: • 10,000 initially susceptible individuals and 2 infected.
• A 50/50 split between males and females.

B5: Combined population-based data sources in a COVID-19 model (PBM, M6.1)

- Purpose:**
- Show how multiple population-based data can be used to inform a COVID-19 model.
 - Using this data, it is possible to estimate branching probabilities and the time variation in the transmission rate $\beta(t)$.
 - However, this sort of data will not estimate transition rates once infected, so informative priors have been used for these.
- Data:**
- Population-level transition data is available for E→I (cases), I→H (hospitalisations) and H→D (hospital deaths).
 - Population data for the combined population in E, I and A (survey PCR data).
 - Population data for the combined population in E, I, A and R (seroprevalence data).
- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains seven compartments: susceptible S, exposed E, infected I, recovered R, asymptomatic A, hospitalised H and dead D.
 - An external force of infection is controlled by parameter η (in reality this could be time-varying and estimated from raw case data and flight information).
 - The time-varying spline $\beta(t)$ captures transmission between individuals.
- System:**
- The UK population, which is initially susceptible.

B6: Time series population observations (IBM, M1.2)

- Purpose:**
- Introduce population-based data incorporated into an IBMs.
- Data:**
- On the 'Inference→Data' page population data is added.
 - This gives time-series estimates for the number of individuals in the infected I compartment.
- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains two compartments: susceptible S and infected I.
 - A transmission rate β determines the rate at which individuals become infected.
- System:**
- 100 initially susceptible individuals and one infected.

C) Individual-level data types

These examples look at different types of individual-level data.

C1: Known transition events — infection and recovery (IBM, M1.3)

- Purpose:**
- Shows an inference example where all transition events are known.
- Data:**
- Known infection and recovery times for all individuals in an SIR model.
- Model:**
- A single population-based species “People” is created.
 - This contains a classification “DS”, which stands for “disease status”.
 - DS contains three compartments: susceptible S, infected I and recovered R.
 - A transmission rate β determines the rate at which individuals become infected.
 - A recovery rate γ determines the rate at which individuals recover.
- System:**
- 100 initially susceptible individuals and one infected.

C2: Incomplete transition events - recovery only (IBM, M1.3)

- Purpose:**
- Show an inference example where only recovery transition events are known.
 - We see that it is still possible to infer β and γ , albeit with more uncertainty.
- Data:**
- Recovery times for individuals in an SIR model.
- Model:**
- A single population-based species “People” is created.

- This contains a classification “DS”, which stands for “disease status”.
- DS contains three compartments: susceptible S, infected I and recovered R.
- A transmission rate β determines the rate at which individuals become infected.
- A recovery rate γ determines the rate at which individuals recover.

System: • 100 initially susceptible individuals and one infected.

C3: Compartmental observations (IBM, M1.3)

- Purpose:** • Illustrate compartmental observations.
- Data:** • The disease status of every individual is measured with a 20 time unit interval.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains three compartments: susceptible S, infected I and recovered R.
 • A transmission rate β determines the rate at which individuals become infected.
 • A recovery rate γ determines the rate at which individuals recover.
- System:** • 100 initially susceptible individuals and one infected.

C4: Disease diagnostic test results (IBM, M1.3)

- Purpose:** • Show an inference example where disease diagnostic tests are used.
- Data:** • Tests are made on individuals every 10 time units.
 • The test has a sensitivity of 0.6 and a specificity of 0.99.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains three compartments: susceptible S, infected I and recovered R.
 • A transmission rate β determines the rate at which individuals become infected.
 • A recovery rate γ determines the rate at which individuals recover.
- System:** • 100 initially susceptible individuals and one infected.

C5: A partially observed transition (IBM, M1.3)

- Purpose:** • Show an inference example where some transitions are partially observed.
- Data:** • Here the data collected assumes that all recovery events are observed, but infections are only observed with a $f=50\%$ probability.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains three compartments: susceptible S, infected I and recovered R.
 • A transmission rate β determines the rate at which individuals become infected.
 • A recovery rate γ determines the rate at which individuals recover.
- System:** • 100 initially susceptible individuals and one infected.

C6: A transition observed over a time window (IBM, M1.3)

- Purpose:** • Shows an inference example where a transition is only observed over a specified time window.
- Data:** • All recovery events are observed.
 • Infection events are only observed between times $t=5$ and $t=15$.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains three compartments: susceptible S, infected I and recovered R.
 • A transmission rate β determines the rate at which individuals become infected.
 • A recovery rate γ determines the rate at which individuals recover.

System: • 100 initially susceptible individuals and one infected.

C7: A transition observed in a demographic category (IBM, M2.8)

Purpose: • This example looks at when some transitions are only observed within a subpopulation.

Data: • All recovery transitions are observed.

• Infections are only observed in males.

Model: • A single population-based species “People” is created.

• This contains classifications “DS”, which stands for “disease status”, and “Sex”.

• “DS” contains three compartments: susceptible S, infected I and recovered R.

• “Sex” contains the compartments M and F for male and female.

• A transmission rate β_s determines the rate at which individuals with sex s become infected.

• A recovery rate γ_s determines the rate at which individuals recover with sex s .

• N is the total number of individuals (this model assumes frequency dependent transmission).

System: • 100 initially susceptible individuals and 2 infected.

• A 50/50 split between males and females.

C8: Uncertain compartmental observations (IBM, M1.3)

Purpose: • This example allows for uncertainty in compartmental observation.

• Here fixed probabilities are used, but the observation model can also include parameters that can be estimated during inference.

Data: • Periodic uncertain compartmental observations are made every 20 time units.

Model: • A single population-based species “People” is created.

• This contains a classification “DS”, which stands for “disease status”.

• DS contains three compartments: susceptible S, infected I and recovered R.

• A transmission rate β determines the rate at which individuals become infected.

• A recovery rate γ determines the rate at which individuals recover.

System: • 100 initially susceptible individuals and one infected.

D) Time variation

These examples look at different ways in which time variation can be added into the model.

D1: Time variation in transmission rate (PBM, M1.1)

Purpose: • Introduce a model with time variation in disease transmission.

• The transmission rate $\beta(t)$ is taken to be a piecewise linear spline with knots taken at 6 time points.

Data: • Time series estimates of the infected population.

Model: • A single population-based species “People” is created.

• This contains a classification “DS”, which stands for “disease status”.

• DS contains two compartments: susceptible S and infected I.

• A transmission rate $\beta(t)$ determines the rate at which individuals become infected.

System: • 100 initially susceptible individuals and one infected.

D2: Time variation in transmission rate using a trigonometric function (PBM, M1.1)

Purpose: • In this example the transmission rate is taken to have a time-varying profile using a cosine curve.

- Data:** • Time series population estimates.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains two compartments: susceptible S and infected I.
 • The transmission rate is taken to be $\beta(1+\Delta \times \cos(\alpha \times t))$.
- System:** • 1,000 initially susceptible individuals and one infected.

D3: Time variation in transmission rate through a covariate (PBM, M1.1)

- Purpose:** • Introduce time-varying covariate into transmission rate.
 • The covariate is exponentiated (to ensure it is positive) and modulates the transmission rate through a strength parameter Δ .
- Data:** • Time series population data is used to inform inference.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains two compartments: susceptible S and infected I.
 • The transmission rate is given by $\beta \times \exp(\Delta \times f(t))$, where $f(t)$ is a time-varying covariate.
- System:** • 1,000 initially susceptible individuals and one infected.

D4: Time variation in population-level transition observation probability (PBM, M1.1)

- Purpose:** • Incorporate time variation in observing population transitions.
 • The probability is taken to have the functional form $a+b \times \cos(0.06 \times t)$.
 • To simulate the data $a=0.6$ and $b=0.3$ were used. These values can be inferred from the data.
- Data:** • Partially observed population-level transition data.
 • Time series population data.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains two compartments: susceptible S and infected I.
 • A transmission rate β determines the rate at which individuals become infected.
- System:** • 1,000 initially susceptible individuals and one infected.

D5: Time variation in individual transition observation probability (PBM, M1.2)

- Purpose:** • Incorporate time variation in the probability of observing individual transitions.
 • Like D4 but with an IBM and individual-level data.
 • The probability of observing transitions is given the functional form $a+b \times \cos(0.03 \times t)$.
 • The values $a=0.5$ and $b=0.3$ were used to simulate the data.
 • Parameters a and b can be inferred from inference.
- Data:** • Partially observed transitions.
 • Periodic compartmental observations.
- Model:** • A single population-based species “People” is created.
 • This contains a classification “DS”, which stands for “disease status”.
 • DS contains two compartments: susceptible S and infected I.
 • A transmission rate β determines the rate at which individuals become infected.
- System:** • 100 initially susceptible individuals and one infected.

D6: Time variation in population observation probability (PBM)

- Purpose:** • Incorporate time variation in observing the number of individuals in a population.
 • Here population observations only detect a fraction f of the true population.

- This, for example, could represent the number of individuals captured during a series of capture campaigns.
- Because capturing individuals is often seasonally dependent, we apply a trigonometric function $f=a\exp(b\cos(0.06xt))$.

Data:	• Time series population estimates. • Simulated data used $a=0.1$ and $b=0.3$.
Model:	• A single population-based species "Animal" is created. • This contains a classification "Area" with a single compartment "P". • "P" represents the animal population.
System:	• 1000 individuals.

D7: Time variation in branching probability (PBM, M2.1)

Purpose:	• Illustrate time variation in branching probability for a population-based model.
Data:	• Time series population measurements.
Model:	• A single population-based species "People" is created. • This contains a classification "DS", which stands for "disease status". • DS contains four compartments: susceptible S, infected I, recovered R and dead D. • A transmission rate β determines the rate at which individuals become infected. • A spline $b(t)$ is used to represent the branching probability of dying. • Parameter γ specifies the recovery/death rate.
System:	• 1000 initially susceptible individuals and two infected.

D8: Time-varying covariate affecting branching probability (IBM, M2.2)

Purpose:	• Introduce time variation in a covariate that affects branching probability. • This is implemented using 'branching factors'.
Data:	• Known event times are used to inform inference.
Model:	• A single individual-based species "People" is created. • This contains a classification "DS", which stands for "disease status". • DS contains four compartments: susceptible S, infected I, recovered R and dead D. • A transmission rate β determines the rate at which individuals become infected. • The <i>relative</i> probability of dying is given by branching factor $a\exp(b\cos(0.06\times t))$. • The <i>relative</i> probability of recovery is given by branching factor 1. • Mean m^R and coefficient of variation c^R specify the log-normally distributed period to recovery. • Mean m^D and coefficient of variation c^D specify the gamma distributed period to death.
System:	• 100 initially susceptible individuals and two infected.

E) Individual-based variation

These examples look at how individual-based variation can be incorporated into the model.

E1: Individual fixed effect applied to a transition (IBM, M1.2)

Purpose:	• Introduce individual fixed effect applied to the infection transition.
Data:	• Infection event times.
Model:	• A single individual-based species "People" is created. • This contains a classification "DS", which stands for "disease status". • DS contains two compartments: susceptible S and infected I. • A transmission rate β determines the rate at which individuals become infected. • A fixed effect $\langle g \rangle$ varies the susceptibilities of individuals based on a covariate.

- A parameter μ^g governs the strength of the fixed effect.

System: • 100 initially susceptible individuals and one infected.

E2: Fixed-effect applied to a population (IBM)

Purpose:

- Introduce adding a fixed effect into a population.
- In the SI model the infected population $\{I\}$ determines how quickly other individual become infected.
- Individual variation in infectivity can incorporated through an individual fixed effect.
- The term $\{I\langle f \rangle\}$ sums over infected individuals, but rather than each individual having a unit contribution it is modified by a fixed effect factor $\langle f \rangle$.

Data:

- Infection times.

Model:

- A single individual-based species “Animal” is created.
- This contains classifications “DS”, which stands for “disease status”, and “Group”.
- “DS” contains two compartments: susceptible S and infected I.
- “Group” contains 24 contact groups.
- A transmission rate β determines the rate at which individuals become infected.
- Fixed effect parameter μ^f determines the strength of the fixed effect.

System:

- Each of the 24 groups consist of 10 individuals, two of which are infected.

E3: Individual effect applied to a transition (IBM, M1.2)

Purpose:

- Introduce a random individual effect affecting the susceptibility of individuals.

Data:

- Infection event times.

Model:

- A single individual-based species “People” is created.
- This contains a classification “DS”, which stands for “disease status”.
- DS contains two compartments: susceptible S and infected I.
- A transmission rate β determines the rate at which individuals become infected.
- An individual effect $[g]$ allows for susceptibility variation across individuals.
- The strength of this variation is determined through variance Ω^{gg} .

System:

- 100 initially susceptible individuals and one infected.

E4: Correlated individual effect applied to a transition (IBM, M1.2)

Purpose:

- Introduce a random individual effect affecting the susceptibility of individuals.
- Unlike E3, the individual effect between individuals is correlated through an A relationship matrix.

Data:

- Infection event times.

Model:

- A single individual-based species “People” is created.
- This contains a classification “DS”, which stands for “disease status”.
- DS contains two compartments: susceptible S and infected I.
- A transmission rate β determines the rate at which individuals become infected.
- An individual effect $[g]$ allows for susceptibility variation across individuals.
- The strength of this variation is determined through variance Ω^{gg} .
- An A relationship matrix determines correlations in individual effect.

System:

- 100 initially susceptible individuals and one infected.

E5: Correlated individual effect applied to a population (IBM, M5.3)

Purpose:

- Introduce an individual effect applied to vary the infectiousness of individuals.
- In the SI model the infected population $\{I\}$ determines how quickly other individual become infected.

- The term $\{I[f]\}$ sums over infected individuals, but rather than each individual having a unit contribution it is modified by an individual effect factor $[f]$ (this factor has a population average of 1 but allows for individual variation).
- The individual effect is correlated through a relationship matrix.

Data: • Known infection times.

Model: • A single individual-based species “Animal” is created.

- This contains classifications “DS” and “Group”.

- “DS” contains two compartments: susceptible S and infected I.

- “Group” contains 60 contact groups.

- A transmission rate β determines how fast individuals become infected.

- $[f]$ give individual effects for infectivity.

- Ω^{ff} give the genetic variance in infectivity.

System: • Each contact group consists of 10 individuals, two of which have been randomly infected.

- 60 sires who each have 10 progeny. These 600 progeny are randomly allocated across the contact groups.

E6: Fixed effect applied to a branching probability (IBM, M2.3)

Purpose: • Introduce individual-variation in branching probability through a fixed effect.

- Branching factors are used.

Data: • Periodic compartmental observations.

Model: • A single individual-based species “People” is created.

- This contains a classification “DS”, which stands for “disease status”.

- DS contains four compartments: susceptible S, infected I, recovered R and dead D.

- A transmission rate β determines the rate at which individuals become infected.

- The *relative* probability of dying is given by $b(g)$.

- The *relative* probability of recovering is given by 1.

- Mean m^R and coefficient of variation c^R specify the log-normally distributed period to recovery.

- Mean m^D and coefficient of variation c^D specify the gamma distributed period to death.

- The strength of the fixed effect is determined through parameter μ^g .

System: • 100 initially susceptible individuals of which two are infected.

E7: Individual effect applied to a branching probability (IBM, M2.3)

Purpose: • Introduce individual-variation in branching probability through an individual effect.

- Branching factors are used.

Data: • Known event times.

Model: • A single individual-based species “People” is created.

- This contains a classification “DS”, which stands for “disease status”.

- DS contains four compartments: susceptible S, infected I, recovered R and dead D.

- A transmission rate β determines the rate at which individuals become infected.

- The *relative* probability of dying is given by $b[g]$.

- The *relative* probability of recovering is given by 1.

- Mean m^R and coefficient of variation c^R specify the log-normally distributed period to recovery.

- Mean m^D and coefficient of variation c^D specify the gamma distributed period to death.

- The variance of the individual effect is determined by Ω^{gg} .

- Individual effects are correlated through a relationship matrix A .

System: • 60 sires each have 10 progeny. These 600 progeny are used in the system (of which 2 are initial infected).

F) Parameter definitions

These examples look at how parameters can be reparameterised, set to distributions or derived from other quantities.

F1: Reparameterisation (IBM, M3.4)

- Purpose:**
- Introduce a farm-based model with reparameterisation.
 - A matrix of interactions $M_{l,l'}$, represents transmission of infection between farms.
 - It is reparameterised as a power-law distribution kernel $1/\left(1 + \left(\frac{D_{l,l'}}{\Delta}\right)^\alpha\right)$.
 - $D_{l,l'}$, is a reserved parameter in BICI which gives the distance between locations.
 - A maximum interaction distance is set, which is used to significantly speed up the algorithm (because distant interactions of negligible effect are removed).
- Data:**
- Event times known.
- Model:**
- A single population-based species “Farm” is created.
 - This contains classifications “DS”, which stands for “disease status”, and “Location”.
 - “DS” contains three compartments: susceptible farm S, infected farm I and recovered farm R.
 - “Location” contains farm positions.
 - A transmission rate β determines the rate at which infected farms transmit disease to surrounding farms.
 - A recovery rate γ determines how quickly farms recover.
- System:**
- A single individual is added to each farm location. All farms are initially susceptible except for one infected.

F2: Parameter distribution (IBM, M5.2)

- Purpose:**
- In some instances, it is useful for a parameter to be sampled from a distribution.
 - An example is a random effect in a mixed model.
 - This is illustrated here with a disease transmission experiment.
- Data:**
- Known transition times.
- Model:**
- A single individual-based species “Animal” is created.
 - This contains classifications “DS”, which stands for “disease status”, “Vaccination” and “Group”.
 - “DS” contains two compartments: susceptible S and infected I.
 - “Vaccination” contains two compartments: “Vac” and “Unvac”, which represent vaccinated and unvaccinated individuals.
 - “Group” contains 24 contact groups.
 - A transmission rate β_v determines the susceptibility of individuals with vaccination status v become infected.
 - A factor f_v determines the relative infectiousness of individuals with vaccination status v .
 - The rate of transmission in different groups is modulated by a group-specific factor G_g .
 - The elements in parameter vector G_g are assumed to be drawn from a normal distribution with mean zero and standard deviation σ^G .
- System:**
- Each group consists of 10 individuals, half of which are infected.
 - The vaccination status of the infected and susceptibility individuals is permuted to create blocks of 4 which are replicated 6 times (to make 24 groups in total).

F3: Derived quantities (IBM, M1.3)

- Purpose:** • Derived quantities are functionally dependent on model parameters or populations, but are not themselves within the model.
• In this example the basic reproduction number R_0 is calculated.
- Data:** • Known event times.
- Model:** • A single population-based species “People” is created.
• This contains a classification “DS”, which stands for “disease status”.
• DS contains three compartments: susceptible S, infected I and recovered R.
• A transmission rate β determines the rate at which individuals become infected.
• A recovery rate γ determines the rate at which individuals recover.
• R_0 is derived by equation $N\beta/\gamma$, where N is the number of individuals.
- System:** • 100 initially susceptible individuals and one infected.

G) Incorporating pathogen genetics

G1: Matrix of genetic differences (IBM, M1.2)

- Purpose:** • Genetic observations on the pathogen allow for the transmission tree to be estimated.
- Data:** • Known event times.
• Genetic observations are taken every 20 time units on all individuals in the system.
• A matrix of genetic differences between the observations is used to inform inference.
- Model:** • A single individual-based species “People” is created.
• This contains a classification “DS”, which stands for “disease status”.
• DS contains two compartments: susceptible S and infected I.
• A transmission rate β determines the rate at which individuals become infected.
- System:** • 100 initially susceptible individuals and one infected.

9) License and warranty

BICI is free software under the terms of the GNU General Public License version 3

www.gnu.org/licenses/gpl-3.0.en.html. This allows users to redistribute and/or modify BICI. The program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY.

10) Citing BICI

We kindly request that those who do use BICI analysis in their publications cite this tool:

Pooley CM, Doeschl-Wilson AB, Marion G., *BICI – A flexible tool for simulation, inference and posterior simulation from compartmental models*. To be submitted (2020)

Acknowledgments

BICI makes use of the software NW.js (from the website nwjs.io/) to create the interface. This software is excellent and highly recommended.

References

- [1] C. M. Pooley, S. C. Bishop, A. B. Doeschl-Wilson, and G. Marion, "Estimating genetic and non-genetic effects for host susceptibility, infectivity and recoverability using temporal epidemic data," *bioRxiv*, p. 618363, 2019.
- [2] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065-1076, 1962.
- [3] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 773-795, 1995.
- [4] H. Jeffreys, *The theory of probability*. OUP Oxford, 1998.
- [5] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical science*, vol. 7, no. 4, pp. 457-472, 1992.
- [6] B. Carpenter *et al.*, "Stan: A probabilistic programming language," *Journal of statistical software*, vol. 76, no. 1, 2017.

Appendices

Appendix A: BICI-script commands ordered by section

BICI-script contains 52 possible commands that can define models, load data, specify analyses, and store outputs. This appendix orders these commands following the menu tree structure. A brief description is given, along with a reference to the section in the manual for more details. Appendix B orders command alphabetically, and provides further implementation details.

Model	Command	Description
→ Compartments	species (see §2.1.1)	Adds a species to the model.
	classification / class (see §2.1.2)	Adds a classification to a species.
	compartment / comp (see §2.1.3)	Adds a compartment to a classification.
	compartment-all / comp-all	Adds a list of compartments to a classification, as specified in a data file.
	transition / trans (see §2.1.4)	Adds a transition to a classification.
	transition-all / trans-all	Adds a list of transitions to a classification, as specified in a data file.
	set (see §7.1)	In the BICI-script this sets the current species and/or classification being focused on.
→ Parameters	parameter / param (see §2.2)	Sets details of a parameter.
	fixed-effect (see §2.2.3)	Provides information about the covariate vector \mathbf{X} used for a fixed effect.
	ind-effect (see §2.2.2)	Provides information about a single (or multiple if correlated) individual effect.
	derived / der (see §2.2.8)	Sets a derived quantity in the model that is functionally related to other model parameters.
→ Annotations	box (see §2.1.8)	Adds a box around a specified set of compartments in the workspace.
	label (see §2.1.8)	Adds a label to the model.
	map (see §2.1.8)	Adds a map to the workspace (only in 'latlng' coordinates).
→ Miscellaneous	description /desc (see §2.3)	Adds a description about the model.
	view	Sets how the classification is viewed in the BICI interface.
	data-dir (see §7.1)	Sets the directory where data is stored.

Simulation	Command	Description
→ <i>Initial Conditions</i>	init-pop-sim (see §3.1.1)	Sets the initial population.
	add-ind-sim (see §3.1.4)	Adds individuals to the system at specified times and into specified compartments.
	remove-ind-sim (see §3.1.5)	Removes individuals from the system at specified times.
	move-ind-sim (see §3.1.6)	Moves individuals to specified compartments at specified times.
	add-pop-sim (see §3.1.2)	Adds populations of individuals into specified compartments at specified times.
	remove-pop-sim (see §3.1.3)	Removes populations of individuals at specified times from specified compartments
→ <i>Parameters</i>	parameter / param (see §3.2)	Sets parameter values.
→ <i>Run</i>	simulation / sim (see §3.3)	Specifies simulation details.
→ <i>Output</i>	sim-param (see §7.3.1)	Command created by the BICL core code to store parameter samples generated by simulation.
	sim-state (see §7.3.2)	Command created by the BICL core code to store state samples generated by simulation.

Inference	Command	Description
→ <i>Data Init. Conds.</i>	init-pop (see §3.1.1)	Sets the initial population.
	add-ind (see §3.1.4)	Adds individuals to the system at specified times and into specified compartments.
	remove-ind (see §3.1.5)	Removes individuals from the system at specified times.
	move-ind (see §3.1.6)	Moves individuals to specified compartments at specified times.
	add-pop (see §3.1.2)	Adds populations of individuals into specified compartments at specified times.
	remove-pop (see §3.1.3)	Removes populations of individuals at specified times from specified compartments.
→ <i>Data Individual</i>	comp-data (see §4.1.2.1)	Provides information about the compartment individuals are in at specified time points.
	trans-data (see §4.1.2.2)	Provides information about the timings of individual transitions.
	test-data (see §4.1.2.3)	Adds disease diagnostic test data.

	genetic-data (see §4.1.2.4)	Incorporates genetic sequence data.
→ <i>Data Population</i>	pop-data (see §4.1.3.1)	Adds population data.
	pop-trans-data (see §4.1.3.2)	Adds aggregated population-level transition data (e.g. number of cases per week).
→ <i>Data Additional</i>	ind-effect-data (see §4.1.4.1)	Adds individual effect data.
	ind-group-data (see §4.1.4.2)	Defines grouping of individuals.
→ <i>Prior</i>	parameter / param (see §4.2)	Sets parameter prior.
→ <i>Run</i>	inference / inf (see §4.3)	Specifies inference details.
→ <i>Output</i>	inf-param (see §7.3.1)	Command created by the BICI core code to store parameter samples generated by inference.
	inf-state (see §7.3.2)	Command created by the BICI core code to store state samples generated by inference.
	inf-diagnostics	Outputs diagnostic information about MCMC samplers used in DA-MCMC or PAS-MCMC.

Posterior Simulation	Command	Description
→ <i>Pop. Modification</i>	add-ind-post-sim (see §3.1.4)	Adds additional individuals to the system at specified times and into specified compartments
	remove-ind-post-sim (see §3.1.5)	Removes additional individuals from the system at specified times.
	move-ind-post-sim (see §3.1.6)	Moves additional individuals to specified compartments at specified times.
	add-pop-post-sim (see §3.1.2)	Adds additional populations of individuals into specified compartments at specified times
	remove-pop-post-sim (see §3.1.3)	Removes additional populations of individuals at specified times from specified compartments.
→ <i>Parameter Modification</i>	param-mult (see §5.2)	Sets up time varying factor that multiplies a model parameter.
→ <i>Run</i>	posterior-simulation / post-sim (see §5.3)	Specifies posterior simulation details.
→ <i>Output</i>	post-sim-param (see §7.3.1)	Command created by the BICI core code to store parameter samples generated by posterior sim.
	post-sim-state (see §7.3.2)	Command created by the BICI core code to store state samples generated by posterior sim.

Appendix B: BICI-script commands alphabetically ordered

This appendix provides an alphabetical list of commands used in BICI-script.

Note, all commands that contain ‘file’ also allow for the optional ‘cols’ specification – This can be used to set alternative column headings, e.g. if the headings “ID” and “t” are expected then cols="Name,Time" would use “Name” in place of “ID” and “Time” in place of “t”.

Command	Description
add-ind / add-ind-sim / add-ind-post-sim (see §3.1.4)	Adds individuals to the system at specified times and into specified compartments ('add-ind-sim' is for simulation, 'add-ind' is for inference and 'add-ind-post-sim' is for posterior simulation). Note, used for individual-based models only. <i>E.g. add-ind file="add-ind-data.csv"</i>
	<i>file</i> – The name of the file that contains the data table. This table must contain at least the following columns: “ID” a unique identifier for individuals, “t” the times at which the individuals are added, and columns for each classification to denote into which state the individuals enter. See §3.1.4 for a description of how uncertainty in compartmental state is implemented.
add-pop / add-pop-sim / add-pop-post-sim (see §3.1.2)	Adds populations of individuals into specified compartments at specified times ('add-pop-sim' is for simulation, 'add-pop' is for inference and 'add-pop-post-sim' is for posterior simulation). Note, used for population-based models only. <i>E.g. add-pop file="add-pop-data.csv"</i>
	<i>file</i> – The name of the file that contains the data table. This table must contain at least the following columns: “t” the times at which the populations are added, columns for each of the classifications to denote into which compartment the individuals enter the system, and “Population”, which gives the number of individuals entering.
box (see §2.1.8)	In the workspace this adds a labelled box around a specified set of compartments. <i>E.g. box text="This is a bounding box" comps="S,I"</i>
	<i>text</i> – Provides the text at the top of the bounding box. <i>comps</i> – Lists the compartments (comma separated). <i>text-size</i> (optional) – Positive integer that sets the size of the text (10 by default). <i>color</i> (optional) – Sets the colour of the box (in hexadecimal or RGB format).
classification / class (see §2.1.2)	Adds a classification to the species. <i>E.g. classification name="DS" index="d"</i>
	<i>name</i> – The name of the classification (note, this must be unique and not the same as any species or compartment). <i>index</i> – This is a single lower-case letter used as a mathematical index that represents compartments in the classification. Each classification must have a unique index (note, ‘t’ cannot be used as that is reserved to represent time). <i>coord</i> (optional) – Sets the coordinate system (set to "cartesian" by default). <ul style="list-style-type: none"> • "cartesian" – Positions of compartments made on an <i>x,y</i> grid. • "latlng" – Positions of compartments made using latitude and longitude.

map (optional) – Loads a default world map (only if using "Inglat" coordinates)

- "load" – Load the map.

Cloning – A classification can be cloned from another species. In this case it adopts exactly the same set of compartments (although its transitions may be defined differently). For cloning the syntax is given by:

E.g. classification name="DS" clone="People"

name – The name of the classification.

clone – The name of the species from which the classification is cloned.

compartment /

comp

(see §2.1.3)

This adds a compartment to a classification.

E.g. compartment name="S" x=0 y=1 color="#ff0000"

name – The name of the compartment.

color – The colour, as specified in hexadecimal or RGB format.

x,y (optional) – The position in the model (in "cartesian" coordinates). If these values are not set, BICI will automatically generate the position (this, however, can sometimes lead to confusing compartmental diagrams).

lat,lng (optional) – Latitude and longitude (in "latlng" coordinates). Must be set if the longitude and latitude are being used for the classification.

boundary (optional) – Instead of a location, this sets a file with boundary data in GeoJSON format (in 'latlng' coordinates).

fix (optional) – Fixes the position of the compartment (*i.e.*, so it cannot be dragged by the mouse in the workspace). This can take the values "true" or "false" (by default set to "false").

branch-prob (optional) – If all transitions leaving a compartment are exponential rates, setting this option to "true" ensures that branching probabilities are used. This can take the values "true" or "false" (by default set to "false").

infected (optional) – When 'trans-tree' is set for the species, this option specifies that the compartment is associated with the "infected" state. This can take the values "true" or "false" (by default set to "false").

compartment-all /

comp-all

Adds a list of compartments to a classification, as specified in a data file (if there are numerous compartments this is more concise than using many 'compartment' commands). Note, this option cannot be used with boundary data.

E.g. comp-all color="#00ff00" file="comp-all-data.csv"

file – The name of the file that contains the data table. This table must contain at least the following columns: "name", which provides the name of the compartment, and "x" and "y" (in "cartesian" coordinates) or "lat" and "lng" (in "latlng" coordinates). Other columns in this table can have headings given by any of the properties below (such that they can be individually specified for each compartment).

color (optional) – The colour as specified in hexadecimal or RGB format.

<i>fix</i> (optional) – Fixes the position of the compartment (<i>i.e.</i> so it cannot be dragged by the mouse in the workspace). This can take the values "true" or "false" (by default set to "false").
<i>branch-prob</i> (optional) – If all transitions leaving a compartment are exponential rates, setting this option to "true" ensures that branching probabilities are used. This can take the values "true" or "false" (by default set to "false").
<i>infected</i> (optional) – When 'trans-tree' is set for the species, this option specifies that the compartment is associated with the "infected" state. This can take the values "true" or "false" (by default set to "false").

comp-data (see §4.1.2.1)	Provides information about the compartment individuals are in at specified time points. <i>E.g.</i> comp-data class="DS" file="comp-data.csv"
	<i>class</i> – The name of the classification on which compartmental information is taken.
	<i>file</i> – The name of the file that contains the data table. This table must contain at least the following columns: "ID" a unique identifier for individuals, "t" the measurement times, and the name of the classification on which the state measurement is made. The last column contains entries such as 'S' to denote the individual is susceptible at a given time point. See §4.1.2.1 for details on how uncertainty is implemented.
data-dir (see §7.1)	Sets the directory where data is stored. <i>E.g.</i> data-dir folder="data-files"
	<i>folder</i> – The directory path.
derived / der (see §2.2.8)	Sets a derived quantity in the model that is functionally related to other model parameters through an equation. Such a quantity usually has some physical meaning (<i>e.g.</i> the reproduction number R_0 in an epidemic model, which is related to the transmission and recovery rates). <i>E.g.</i> derived name="R0" eqn=" β/γ "
	<i>name</i> – The name of the new parameter. <i>eqn</i> – The equation that relates it to other parameters in the model.
description / desc (see §2.3)	This adds a description about the model. <i>E.g.</i> description text="file.txt "
	<i>text</i> – A text file provides a description of the model and analysis. A markdown format is used, <i>i.e.</i> this allows for the following concise formatting: "# Title", ## Subtitle", "- Bullet point", "*italic text*" and "**bold text**". Parameters can be added by enclosing in dollar symbols, <i>e.g.</i> "\$a\$" or "\$b^super_sub\$".
fixed-effect (see §2.2.3)	Provides information about a covariate vector \mathbf{X} used for a fixed effect. <i>E.g.</i> fixed-effect name="m" X="X-data.csv"

name – The name of the fixed effect.

X – The name of the file that contains the data table giving information on the covariate vector. This table must contain at least the following columns: “ID” a unique name for each individual and “Value” giving the value of the covariate.

genetic-data

Incorporates genetic sequence data into the analysis.

(see §4.1.2.4)

E.g. genetic-data type="snp" mut-rate="m" seq-var="v" root="SNP" file="genetic-data.csv"

type – Provides the type of genetic data. There are two possibilities:

- "matrix" – This makes use of a matrix of base-pair differences for a set of pathogen genetic observations.
- "snp" – This makes use of raw SNP data for a set of pathogen genetic observations.

root (optional) – Gives the root name for the SNP in the file. Must be specified if ‘type’ is set to "snp".

file – The name of the file that contains the data table. This table must contain at least the following columns: “ID” a unique name for each individual, “t” giving the time of the observation. The following columns depend on the type of observation:

- "matrix" – A column “Obs” provides a unique name for each genetic observation. Elements down this column must also be columns in the table that represent a square matrix giving base-pair differences between the observations.
- "snp" – The data table has a series of columns giving sequence data which have ‘root’ at the beginning of their column name. Elements of these columns are expected to contain two characters denoting the version of the SNP, e.g. ‘AA’, ‘AB’, ‘BA’, or ‘BB’.

ind-effect

Provides information about a single or multiple (if correlated) individual effects.

(see §2.2.2)

E.g. ind-effect name="f,g" A="A-matrix-data.csv"

name – The individual effects are listed (comma separated).

A (optional) – When individual effects are correlated across individuals (e.g. genetically) it is necessary to set up an **A** relationship matrix (which could be pedigree-based or represent a genomic relationship matrix). This is loaded by means of a file that has the individuals in the population in the column headers and a square matrix represented in the table.

A-sparse (optional) – An alternative to ‘A’ and used for a sparse representation of the matrix. This table must have three columns giving the *x* and *y* coordinates in the matrix and then the value. The matrix is assumed to be symmetric, so only values in the upper triangular part of the matrix need to be specified. The order of individuals is provided by ‘ind-list’. Any unspecified elements are assumed to be zero.

ind-list (optional) – Gives a table with heading “Individual”, which gives the order of individuals used for ‘A-sparse’.

ind-effect-data (see §4.1.4.1)	Adds individual effect data (note, such information is usually unknown and typically only available when the data is generated through simulation). <i>E.g.</i> ind-effect-data file="ind-effect-data.csv"
	<i>file</i> – The name of the file that contains the data table. This table must contain at least the following columns: “ID” a unique identifier for individuals and “Value”, which gives values for the individual effect.
ind-group-data (see §4.1.4.2)	Defines a grouping of individuals. Note, this is not used in simulation or inference, but rather allows for visualisation of specified groupings of individuals (<i>e.g.</i> sires, progeny) in the interface, as well as estimates of prediction accuracy for individual effects. <i>E.g.</i> ind-group-data file="ind-group-data.csv"
	<i>file</i> – The name of the file that contains the data table. This table must contain an “ID” column, a unique identifier for individuals.
inf-diagnostics	Outputs MCMC diagnostic information for the DA-MCMC and PAS-MCMC algorithms. <i>E.g.</i> inf-diagnostics chain=0 file="diagnostics0.txt"
	<i>file</i> – The name of the file providing diagnostic information. <i>chain</i> – The MCMC chain number.
inference / inf (see §4.3)	Specifies inference details. <i>E.g.</i> inference start=0 end=100 timestep=0.1 update=1000 algorithm="DA-MCMC"
	<i>start</i> – The start time for inference. <i>end</i> – The end time for inference. <i>timestep</i> – Sets a finite time-step used to approximate the governing equations. A smaller value provides more accurate results but takes computationally longer. Typically setting this to around 20% of the shortest expected transition period gives fast, accurate results. <i>algorithm</i> (optional) – The algorithm used to perform inference (by default set to "DA-MCMC"). It can take the following possibilities: <ul style="list-style-type: none"> • "DA-MCMC" – Data augmentation MCMC provides the most accurate inference (although it can be slow for many individuals). • "PAS-MCMC" – Particle annealed sampling (PAS) generates a power posterior approximation that get successively closer to the true posterior over a series of generations. After annealing, PAS-MCMC gathers parameter and state samples in the same way as DA-MCMC. PAS-MCMC cannot be run on the local machine. • "ABC" – Approximate Bayesian Computation. Estimates the posterior by simulating from the model and comparing the output with the data (population-based data only). Only those samples which are sufficiently close (as measured by an error function) are used for the posterior approximation.

-
- "ABC-SMC" – Approximate Bayesian Computation using Sequential Monte Carlo. A version of ABC that improves posterior estimates over a series of generations (population-based data only).
 - "PMCMC" – Particle MCMC targets the true posterior and makes use of multiple "particles" (stochastic simulations from the model) run in parallel.
 - "MFA" – Stands for mean field approximation. Here an emulator is used to approximate the posterior surface. This can be much faster than other approaches, provided the number of model parameters is not too large.

update (optional, for DA-MCMC or PAS-MCMC) – Sets the number of updates MCMC is iterated for (by default set to 5000). A higher value takes longer to run but may improve the results (because output samples become less correlated).

nchain (optional, for DA-MCMC) – The number of MCMC chains to run (by default set to 3).

chain-per-core (optional, for DA-MCMC) – The number of MCMC chains run per core (by default set to 1). The number of chains must be a multiple of this value.

npart (optional, for PAS-MCMC) – Sets the number of particles used (by default set to 3).

part-per-core (optional, for PAS-MCMC) – The number of particles run per core (by default set to 1). The number of particles must be a multiple of this value.

gen-percent (optional, for PAS-MCMC) – Sets the number of MCMC updates used per generation in the annealing phase as a percentage of the number (by default set to 100).

param-output (optional, for DA-MCMC or PAS-MCMC) – Sets the number of parameter samples generated in the output (by default set to 1000). A larger value yields smoother parameter distribution plots, but consumes more memory and CPU processing when visualised.

state-output (optional, for DA-MCMC or PAS-MCMC) – Sets the number of state samples generated in the output (by default set to 200). A larger value yields smoother population, transition and individual plots, but consumes more memory and CPU processing when visualised.

sample (optional, for ABC) – Sets the number of samples generated (by default set to 1000).

acc-frac (optional, for ABC) – Sets the fraction of simulated samples accepted to be posterior samples (by default set to 0.1). Making smaller improves the posterior estimate but takes longer to run.

sample (optional, for ABC-SMC) – Sets the number of samples generated (by default set to 1000).

acc-frac (optional, for ABC-SMC) – Sets the fraction of simulated samples accepted to be posterior samples (by default set to 0.5).

gen (optional, for ABC-SMC) – Sets the number of generations (by default set to 5).

kernel-size (optional, for ABC-SMC) – Sets the size of the proposal kernel (by default set to 0.5). This size scales the current MVN approximation to the posterior.

burnin-percent (optional, for DA-MCMC) – The percentage of the chain in which burn-in is performed (by default set to 20). Must be in the range 1-90.

anneal (optional, for DA-MCMC) – Type of annealing performed during the burnin phase (by default "none"). It can take the following possibilities:

- "none" – The inverse temperature is fixed to one during burn-in.
- "scan" – The inverse temperature is scanned from zero to one during burn-in. The 'rate' property must be set.
- "log-auto" – An automatic logarithm model is applied to the inverse temperature.
- "power-auto" – An automatic power-law model is applied to the inverse temperature.
- "power" – A power law model is applied. The 'power' property must be set.

rate (optional, for DA-MCMC) – When 'anneal' is set to "scan", this sets how fast burn-in takes place (under this option the overall time for burn-in is *a priori* unknown). By default, set to "0.01".

power (optional, for DA-MCMC) – When 'anneal' is set to "power", this sets the power which controls how the inverse temperature is changed. By default, set to "4".

ind-max (optional) – Sets the maximum number of individuals allowed (by default set to 20000). If exceeded, an error message is generated.

param-output-max (optional) – Sets the maximum size of tensors which can be output (by default set to 1000). This avoids very large output files.

seed (optional) – An integer between 0 and 10,000 that sets the computational seed for the pseudorandom number generator (by default set to a fixed value). Changing this value alters the stochastic realisation of the output.

diagnostics (optional) – Determines if MCMC diagnostics are output (these are used in DA-MCMC and PAS-MCMC to determine how well proposals are working). Can take the values "off" or "on" (by default set to "off"). If "on" the 'inf-diagnostics' command is generated when inference is run.

inf-param This command is created by the BICI core code and stores parameter samples generated by inference.

(see §7.3.1) *E.g. inf-param nchain=1 file="parameter-sample-data.csv"*

nchain (optional) – Denotes the MCMC chain from which the samples were taken.

file – The data file which contains parameter samples generated from inference. The first column "State" stores the MCMC iteration on which the sample was taken. Other columns contain parameters as well as log-likelihoods and the prior.

inf-state This command is created by the BICI core code and stores state samples generated by inference.

(see §7.3.2) *E.g. inf-state nchain=1 file="state-sample-data.txt"*

nchain (optional) – Denotes the MCMC chain from which the samples were generated.

	<i>file</i> – The data file which contains state samples generated from inference. See §7.3.2 for format details.
init-pop / init-pop-sim (see §3.1.1)	Sets the initial population (init-pop-sim is for simulation and init-pop is for inference). <i>E.g.</i> init-pop type="fixed" focal="DS" file="init-pop-data.csv"
	<i>file</i> – The file used to load the data table (see below for details on required columns in this table).
	<i>focal</i> (optional) – Set to a classification name. If this is set, it means that a focal classification is used to specify the population/distribution (see below).
	<i>type</i> (optional) – Determines how the initial population is defined (by default set to "fixed"):
	<ul style="list-style-type: none"> • "fixed" – The initial population is exactly defined. If 'focal' is set – The file must contain two columns with titles "Compartment" and "Population". The "Compartment" column must go through all compartments in the focal classification and all but one for all other classifications (the remaining is automatically calculated such that percentages/fractions add up to 100%). In the case of the focal classification, the "Population" column gives the number of individuals in each compartment. Other classifications are represented by percentages. If 'focal' not set – The file must contain columns for each classification and a final column "Population". This last column gives the number of initial individuals. • "dist" – The initial population is sampled from a distribution. If 'focal' is set – The file must contain two columns with titles "Compartment" and "Distribution". The "Compartment" column must go through all compartments in the model. In the case of the focal classification, the "Distribution" column gives the distribution in population number sampled from, e.g. "uniform(10,20)". Other classifications are represented by α values used to define Dirichlet priors. If 'focal' not set – The file must contain columns for each classification and a final column "Alpha". This column defines a Dirichlet prior across all possible compartment combinations. For this option a prior for the overall population number must be set using the 'prior' property.
	<i>prior</i> (optional) – When 'type' is set to "dist" and 'focal' is unset, this defines the prior for the overall population number, e.g. "gamma(1000,0.5)" (see §4.2.1 for prior options).
label (see §2.1.8)	Adds a label to the model workspace. <i>E.g.</i> label text="This is some text" x=13 y=16 color="#ff2222" text-size=20 <i>text</i> – The text to be displayed. <i>x,y</i> (optional) – The position in the workspace (in "cartesian" coordinates). <i>lat,lng</i> (optional) – Latitude and longitude (in 'latlng' coordinates). <i>color</i> (optional) – The colour of the text (black by default). <i>text-size</i> (optional) – The size of the text (set to 10 by default).

map (see §2.1.8)	Adds a map to the workspace (only in ‘latlng’ coordinates). <i>E.g.</i> map file="UK_boundary.geojson"
	<i>file</i> – The name of the map file (currently only GeoJSON format using latitude and longitude is accepted).
move-ind / move-ind-sim / move-ind-post-sim (see §3.1.6)	Moves individuals to specified compartments at specified times ('move-ind-sim' is for simulation, 'move-ind' is for inference and 'move-ind-post-sim' is for posterior simulation). <i>E.g.</i> move-ind class="DS" file="move-ind-data.csv"
	<i>class</i> – The name of the classification in which the compartmental movement is made. <i>file</i> – The name of the file that contains the data table. This table must contain at least the following columns: "ID" a unique identifier for individuals, "t" the times at which the individuals are moved, and "To", which gives the compartment to which the individuals are moved.
parameter / param (see §2.2 / §3.2 / §4.2)	Sets details of a parameter. <i>E.g.</i> param name=" $\beta_I(t)$ " value="value-data.csv" knot-times="start,10,end" smooth="normal(0.2)" <i>name</i> – The name of the parameter. If it depends on one or more classification, its name is followed by an underscore ‘_’ followed by a comma separated list of any classification indices on which the parameter depends. If the parameter has a time dependency it ends with '(t)'. <i>value</i> (optional) – Sets the value for the parameter under simulation. If the parameter is multivariate (<i>i.e.</i> it has a classification or time dependency), the name of a specification file is used. This file has columns with headings given by indices for each of the dependences (with "t" being used in the case of a time dependency) followed by a column with the heading "Value". Any elements not specified in the rows of the table are set to zero by default. <i>const</i> (optional) – Like 'value', but sets the parameter to a constant (<i>i.e.</i> its takes the same fixed value for simulation, inference or posterior inference). <i>reparam</i> (optional) – Sets the reparameterisation for the parameter. Here there are two possibilities. <ul style="list-style-type: none"> • Equation – The parameter is reparameterised in terms of an equation. • Split – Here 'reparam' is set to a file. This file has columns with headings given by indices for each of parameter dependences (with "t" being used in the case of a time dependency) followed by a column with the heading "Value". Any elements not specified in the rows of the table are set to zero by default. <i>knot-times</i> (optional) – When the parameter is represented by a spline (<i>i.e.</i> it has a '(t)') this sets the times of the spline knots, comma separated. The key words 'start' and 'end' are used to denote the start and end times of the simulation/inference.

smooth (optional) – When the parameter is represented by a spline (*i.e.* it has a '(t)') this sets a smoothing condition on the spline. It can take the following values (where the standard deviation SD determines the strength of the smoothing):

- "normal(SD)" – Used for parameters that can become negative or positive.
- "log-normal(SD)" – Used for parameters that are strictly positive.

prior (optional) – Sets the prior for the parameter. The prior can take any of the following options:

- "uniform(min,max)" – A uniform distribution with specified minimum and maximum, *e.g.* "uniform(4,5)".
- "exp(mean)" – An exponential distribution with specified mean (only appropriate if the parameter is strictly positive).
- "normal(mean,sd)" – A normal distribution with a specified mean and standard deviation.
- "gamma(mean,cv)" – A gamma distribution with a specified mean and coefficient of variation (for positive parameters only).
- "log-normal(mean,cv)" – A log-normal distribution with a specified mean and coefficient of variation (for positive parameters only).
- "beta(α,β)" – A beta distribution with specified alpha and beta values (for parameters that go between 0 and 1 only). Note, the distribution mean is given by $\alpha/(\alpha+\beta)$.
- "bernoulli(mean)" – A Bernoulli distribution with a specified mean (for binary parameters that can take the values 0 or 1). The mean must be between 0 and 1.
- "fix(value)" – Fixes the parameter to a particular value.
- "mdir(σ)" – A modified Dirichlet distribution with specified sigma. Note, this is only used when parameters are specified as categorical factors (see §2.2.9).

prior-split (optional) – In cases in which the parameter has dependencies, it may be useful to set the prior differently for each element. This option is set to a file name giving a table of prior definitions. That file has columns for each of the dependences follow by a column with the heading "Prior". The last column uses the same formatting as for 'prior' above.

dist (optional) – Sets a distribution for the parameter. This is like 'prior' (and has the same set of options), but here the distribution contains new parameters, *e.g.* "normal(m,s)", where 'm' and 's' would go on to have their own priors. So instead of becoming part of the prior the distribution forms part of a so-called 'hierarchical' model. For example, random effects have an assumed normal distribution with a standard deviation that is estimated.

dist-split (optional) – In cases in which the parameter has dependencies, it may be useful to set the distribution differently for each element. This option is set to a file that contains a table of distribution definitions. That file has columns for each of the dependences follow by a column with the heading "Prior". The last column uses the same formatting as for 'dist' above.

sim-sample (optional) – In the case in which a distribution is set (via the "dist" option), setting 'sim_sample' to "true" ensures what when the model is simulated, the parameter is sampled from the specified distribution (rather than

specified directly by the user). It can take the values "true" or "false" (the value "true" is used by default).

factor (optional) – Determines if a parameter is set to be a categorical factor (see §2.2.9). Can we set to "true" or "false" (the value "false" is used by default).

factor-weight (optional) – Determines the weight of a factor (see §2.2.9). If unset, all index options have equal weight.

param-mult Sets up a spline factor that multiplies a parameter when posterior simulation is performed.

(see §5.2) `param name=" β " constant="factor-data.csv" knot-times="start,10,end"`

name – The name of a parameter within the model.

constant – Sets the constant values for the parameter factor along the spline. This file has columns with headings given by indices for each of the dependences in the parameter, "t" which gives the spline knot times, followed by a column with the heading "Value". Any elements not specified in the rows of the table are set to zero by default.

knot-times – This sets the times of the spline knots, comma separated. The key words 'start' and 'end' are used to denote the start and end times of the inference.

pop-data Adds population data to the analysis.

(see §4.1.3.1) `E.g. pop-data filter="DS=I" error="normal:10%" file="pop-data.csv"`

file – The name of the file that contains the data table. This table must contain at least the following columns: "t" the times at which the population measurements are made, columns which specify any filters applied to specify the sub-population under observation (see 'filter' below), and "Population", which gives the estimated population size. In the case in which 'error' is set to "normal:file", a final column "SD" is used to specify the standard deviation in the observation error. If 'error' is set to "neg-binomial:file", a final column "p" is used to specify the probability p in the negative binomial distribution.

filter (optional) – Filters subpopulation. In the example above, the measured sub-population contains only individuals in the 'I' compartment in classification 'DS'. Multiple compartments are separated using the '|' character, e.g. "DS=E|I" would mean that the subpopulation includes those in either the E or I states. The fraction of individual observed can also be set, e.g. "DS=E:0.4|I:0.9" means that 40% of E individuals and 90% of I individuals are observed. Note, fractions can be larger than one because observations could represent something proportional to the number of infected individual (e.g. environmental pathogen concentration). Equations can be used to represent fractions, e.g. "DS=E:p|I: $\eta(t)$ " where the parameter p and spline $\eta(t)$ could be estimated during inference. Filtering can be done on more than one classification by comma separating, e.g. "DS=I, Sex=M" would mean that only male individual in the infected I state are observed. A filter can be placed into the data file using the 'file' keyword, e.g. "Location=file". In this case the file would contain a column with heading "Location" containing the filter.

error (optional) – The observation error captures noise in the measurement process itself. This assumes that the observed value is drawn from a distribution with mean given by the true underlying value. There are three possibilities for this distribution:

- **Poisson** – ‘error’ is set by “poisson”. This has a variance equal to the mean.
- **Normal** – This distribution can be chosen to have a variance less than the Poisson. It can be defined in one of three different ways: 1) As a percentage, e.g. “normal:10%” implies the standard deviation (SD) is 10% the mean, 2) As a SD, e.g. “normal:2.5” implies a fixed SD of 2.5, or 3) Defined by column “SD” in the file, e.g. “normal:file”.
- **Negative binomial** – This distribution has a variance greater than the Poisson. It can be defined in one of two different ways: 1) With a fixed p value, e.g. “neg-binomial:0.5” has $p=0.5$ or 2) Defined by a column “p” in the file, e.g. “neg-binomial:file”.

pop-trans-data Adds aggregated population-level transition data to the analysis (e.g. the number of cases per week).

(see §4.1.3.2) *E.g. pop-trans-data name="S->I" filter="M" file="pop-trans-data.csv"*

name – The name of the transition. Multiple transitions can be specified in the same classification, e.g. “E->A|E->I” would mean that both ‘E→A’ and ‘E→I’ transition would both be counted in the observation. The fraction of observed transitions can also be applied, e.g. “S->I:0.5” would imply that only 50% of transitions are observed. Such fractions can be parameterised e.g. “S->I:p” or “S->I: $\eta(t)$ ”.

file – The name of the file that contains the data table. This table must contain at least the following columns: “Start” and “End” giving the time period over which transitions are counted, any filters applied to specify the sub-population under observation (see ‘filter’), and “Number” which gives the estimated number of transitions. In the case in which ‘error’ is set to “normal:file”, a final column “SD” is used to specify the standard deviation in the observation model. If ‘error’ is set to “neg-binomial:file”, a final column “p” is used to specify the probability p in the negative binomial distribution.

filter (optional) – Filters subpopulation. This uses the same specification as ‘filter’ for ‘pop-data’. Note, filtering cannot be performed on the classification in which the transition occurs.

error (optional) – The observation error captures noise in the measurement process itself. This assumes that the observed value is drawn from a distribution with mean given by the underlying value. There are three possibilities for this distribution:

- **Poisson** – ‘error’ is set by “poisson”. This has a variance equal to the mean.
- **Normal** – This distribution can be chosen to have a variance less than the Poisson. It can be defined in one of three different ways: 1) As a percentage, e.g. “normal:10%” implies the standard deviation (SD) is 10% the mean, 2) As a SD, e.g. “normal:2.5” implies a fixed SD of 2.5, or 3) Defined by column “SD” in the file, e.g. “normal:file”.
- **Negative binomial** – This distribution has a variance greater than the Poisson. It can be defined in one of two different ways: 1) With a fixed p

	value, e.g. "neg-binomial:0.5" has p=0.5 or 2) Defined by a column "p" in the file, e.g. "neg-binomial:file".
posterior-simulation/post-sim	Specifies posterior simulation details. <i>E.g.</i> post-sim start=0 end=100 number=10
(see §5.3)	<p><i>start</i> (optional) – The start time for the posterior simulation (by default set to the inference start time).</p> <p><i>end</i> (optional) – The end time for the posterior simulation (by default set to the inference end time).</p> <p><i>number</i> (optional) – The number of posterior simulations to be run (by default set to 200).</p> <p><i>ind-max</i> (optional) – Sets the maximum number of individuals allowed (by default set to 20000). If exceeded, an error message is generated.</p> <p><i>param-output-max</i> (optional) – Sets the maximum size of tensors which can be output (by default set to 1000). This avoids very large output files.</p> <p><i>resample</i> – Determines if distributions are resampled or taken directly from the posterior. Should be set to a comma separate list of parameter names that need resampling.</p> <p><i>seed</i> (optional) – An integer between 0 and 10,000 that sets the computational seed for the pseudorandom number generator (by default set to a fixed value). Changing this value alters the stochastic realisation of the output.</p>
post-sim-param	This command is created by the BICI core code and stores parameter samples generated by posterior simulation. <i>E.g.</i> post-sim-param file="parameter-sample-data.csv"
(see §7.3.1)	<i>file</i> – The data file which contains parameter samples generated from posterior simulation. The first column "State" stores the posterior simulation number. Other columns contain parameter information as well as log-likelihoods and the prior.
post-sim-state	This command is created by the BICI core code and stores state samples generated by posterior simulation. <i>E.g.</i> post-sim-state file="state-sample-data.txt"
(see §7.3.2)	<i>file</i> – The data file which contains state samples generated from posterior simulation. See §7.3.2 for formatting details.
remove-ind / remove-ind-sim / remove-ind-post-sim	Removes individuals from the system at specified times ('remove-ind-sim' is for simulation, 'remove-ind' is for inference and 'remove-ind-post-sim' is for posterior simulation). <i>E.g.</i> remove-ind file="remove-ind-data.csv"
(see §3.1.5)	<i>file</i> – The name of the file that contains the data table. This table must contain at least the following columns: "ID" a unique identifier for individuals and "t" the times at which individuals are removed.

remove-pop / remove-pop-sim / remove-pop-post- sim	Removes populations of individuals at specified times from specified compartments ('remove-pop-sim' is for simulation, 'remove-pop' is for inference and 'remove-pop-post-sim' is for posterior simulation). Note, used for population-based models only.
(see §3.1.3)	<p><i>E.g. remove-pop file="remove-pop-data.csv"</i></p> <p><i>file</i> – The name of the file that contains the data table. This table must contain at least the following columns: "t" the times at which populations are removed, columns for each of the classifications to denote from which compartment the individuals leave the system, and "Population", which gives the number of individuals leaving.</p>
species	Adds a species to the model.
(see §2.1.1)	<p><i>E.g. species name="People" type="individual"</i></p> <p><i>name</i> – The name of the species.</p> <p><i>type</i> – The type of model. This can take one of the following possibilities:</p> <ul style="list-style-type: none"> • "individual" – An individual-based model (<i>i.e.</i> each individual in the system has a timeline). This allows for individual-based data as well as non-Markovian transitions. • "population" – A population-based model (here the populations in different compartments are tracked as a function of time). The transitions in the model must be Markovian and data must be population-based. <p><i>trans-tree</i> (optional) – Sets if the transmission tree is turned on (<i>i.e.</i>, information about who acquires infection from whom is stored). This can take the values "on" or "off" (set to "off" by default).</p>
set	Sets the current species and/or classification, such that all subsequent commands act on that species/classification.
(see §7.1)	<p><i>E.g. set species="People"</i></p> <p><i>species</i> – The name of the species.</p> <p><i>classification</i> – The name of the classification.</p>
simulation / sim	Specifies simulation details.
(see §3.3)	<p><i>E.g. simulation start=0 end=100 timestep=0.5 number=10</i></p> <p><i>start</i> – The start time for the simulation.</p> <p><i>end</i> – The end time for the simulation.</p> <p><i>timestep</i> – Sets a finite time-step used to approximate the governing equations. A smaller value provides more accurate results but takes computationally longer. Typically setting this to around 20% of the shortest expected transition period gives fast, accurate results.</p> <p><i>number</i> (optional) – The number of simulations to run (by default set to 1).</p>

ind-max (optional) – Sets the maximum number of individuals allowed (by default set to 20000). If exceeded, an error message is generated.

param-output-max (optional) – Sets the maximum size of tensors which can be output (by default set to 1000). This avoids very large output files.

seed (optional) – An integer between 0 and 10,000 that sets the computational seed for the pseudorandom number generator (by default set to a fixed value). Changing this value alters the stochastic realisation of the output.

sim-param	This command is created by the BICI core code and stores parameter samples generated by a simulation. <i>E.g.</i> sim-param file="parameter-sampl-data.csv"
(see §7.3.1)	<i>file</i> – The data file which contains parameter samples generated from simulation. The first column “State” stores the number of the simulation. Other columns contain parameter information as well as log-likelihoods and the prior.
sim-state	This command is created by the BICI core code and stores state samples generated by a simulation. <i>E.g.</i> sim-state file="state-sample-data.txt"
(see §7.3.2)	<i>file</i> – The data file which contains state samples generated from simulation. See §7.3.2 for formatting details.
test-data	Adds disease diagnostic test data. <i>E.g.</i> test-data Se="1" Sp="1" pos="+" neg="-" comp="I" file="test-data.csv"
(see §4.1.2.3)	<i>Se</i> – The sensitivity of the test. This can either be a number between 0 and 1 or a parameter. <i>Sp</i> – The specificity of the test. This can either be a number between 0 and 1 or a parameter. <i>pos/neg</i> (optional) – Sets the text value in the data file used to represent a positive/negative test result (1/0 by default). <i>comp</i> – Sets the compartment(s), comma separated, that the test is sensitive to. <i>file</i> – The name of the file that contains the data table. This table must contain at least the following columns: “ID” a unique identifier for individuals, “t” the times at which the individuals are tested and “Result” which provides the test results.
trans-data	Provides information about the timings of observed individual transitions. <i>E.g.</i> trans-data name="S->I" filter="Sex=M" obsrange="all" file="trans-data.csv"
(see §4.1.2.2)	<i>name</i> – The name of the transition being observed, which is made up of the initial compartment name, followed by ‘->’ and the final compartment name. Source transitions use the ‘+’ character, <i>e.g.</i> "+>S". Sink transition use the ‘-’ character, <i>e.g.</i> "R->". Multiple transitions can be observed, <i>e.g.</i> "E->A E->I" would imply that the observed transition would be one of these two possibilities. A probability can be assigned to observing the transitions, <i>e.g.</i> "S->I:0.3" would mean only a 30% probability of observing S→I transitions. This observation

probability can be parameterised, e.g. "S->I:p" or "S->I: $\eta(t)$ ", where the parameters can be estimated during inference.

filter (optional) – Filters the population being observed (this can apply to any classification apart from the one in which the transition occurs). In the example above, the measured sub-population contains only individuals in the 'M' compartment in classification 'Sex'. Multiple compartments are separated using the '|' character, e.g. "Location=A|C" would mean that the subpopulation includes those in either the A or C locations. The probability of observing the transition can also be given a factor, e.g. "Sex=M:0.4" means that the S→I transitions in M individuals are only observed with a 40% probability. Equations can be used to represent probabilities, e.g. "Sex=M: $\eta(t)$ " where the observation probability spline $\eta(t)$ could be estimated during inference. Filtering can be done on more than one classification by comma separating, e.g. "Location=A, Sex=M" would mean that only transitions on male individual at location A are observed.

obsrange – Determines how the observation period for the transition is specified. It can take the following possibilities:

- "all" – This means that the transition is observed over the entire inference period.
- "specify" – The start and end of the observation period are specified (see below).

start/end (optional) – Used to specify the observation period when 'obsrange' is set to "specify".

file – The name of the file that contains the data table. This table must contain at least the following columns: "ID" a unique identifier for individuals and "t" the times at which individuals undergo transitions.

transition / trans	This adds a transition to a classification. <i>E.g. transition name="S->E" value="[\beta]*{l}"</i>
(see §2.1.4)	<p><i>name</i> – Identifies the transition, with the initial compartment name followed by "->" and the final compartment name. Source transitions use "+", e.g. "+->S". Sink transitions use "-", e.g., "R->-".</p> <p><i>value</i> – Sets the transition distribution. This can take one of the following possibilities:</p> <ul style="list-style-type: none">• "exp(rate:[eq])" – An exponential distribution, where "[eq]" is replaced with an equation that determines the rate at which transitions occurs, e.g. "exp(rate:r)" or "exp(rate:\beta\times{l})".• "gamma(mean:[eq1],cv:[eq2])" – A gamma distribution with equations for mean and coefficient of variation.• "erlang(mean:[eq],shape:[integer])" – An Erlang distribution with equation for mean and fixed positive integer shape.• "log-normal(mean:[eq1],cv:[eq2])" – A log-normal distribution with equations for mean and coefficient of variation.• "weibull(scale:[eq1],shape:[eq2])" – A Weibull distribution with equations for scale and shape.• "period(time:[eq])" – For when the event happens after a certain prescribed period.

Note, source transitions can only take the "exp" option. Population-based models are only valid for the "exp" or "erlang" options.

bp (optional) – This sets the equation for the branching probability. This is only used if more than one transition leaves the initial compartment.

x, y (optional) – In the case of a source or sink transitions, this positions the ‘+’ or ‘-’ symbol in the workspace (in “cartesian” coordinates).

lat, lng (optional) – In the case of a source or sink transitions, this positions the ‘+’ or ‘-’ symbol in the workspace (in ‘latlng’ coordinates).

mid-x, mid-y (optional) – Can be used to add extra points (comma separated) on the line from the initial to final compartments (in “cartesian” coordinates).

mid-lat, mid-lng (optional) – Can be used to add extra points (comma separated) on the line from the initial to final compartments (in ‘latlng’ coordinates).

transition-all / trans-all Adds a list of transitions to a classification, as specified in a data file (if there are numerous transitions, this is more concise than using many separate ‘transition’ commands).

E.g. trans-all file="trans-all-data.csv"

file – The name of the file that contains the transition data table. This table must contain at least the following columns: “name” and “value” (see ‘transition’ for formatting details). Optionally, additional columns can have headings given by other properties in ‘transition’, *i.e.* “bp”, “x”, “y”, “lat”, “lng”, “mid-x”, “mid-y”, “mid-lat”, “mid-lng”.

view This determines the position and scaling used when viewing the model in the BICI interface. This command ensures that when the model is saved and reloaded the workspace view is preserved.

E.g. view x=4 y=0 scale=2

x,y (optional) – The *x* and *y* position at the centre of the workspace view (in “cartesian” coordinates).

lat,lng (optional) – The latitude and longitude at the centre of the workspace view (in ‘latlng’ coordinates).

scale – Determines how zoomed in the view is.

grid (optional) – Plots the grid in the visual interface. Can take the values “off” and “on” (set to “off” by default).

comp-scale (optional) – Scales the size of compartments as they appear on the screen. This can be used to tackle cases in which compartmental labels overlap each other (can take values between 0.1 and 10, by default set to one).

Appendix C: Parameter visualisations

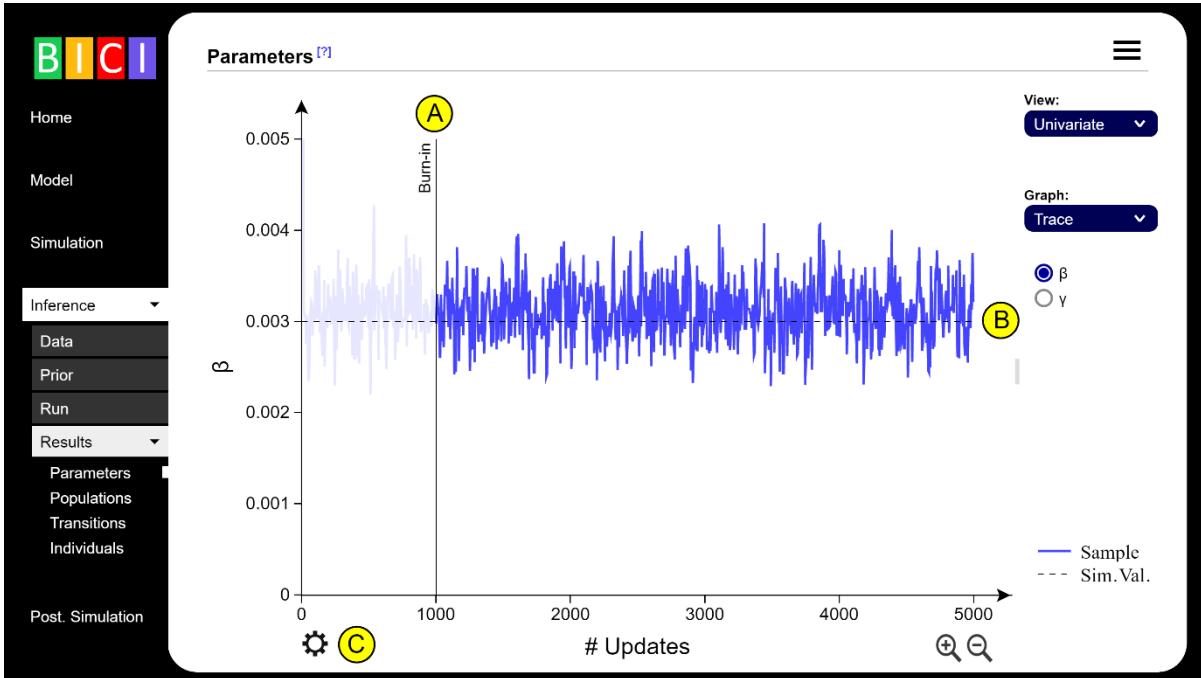


Figure C1 – Trace plot. MCMC works by iteratively performing “updates” and drawing posterior samples. The number of updates is shown on the x-axis and a selected parameter on the y-axis. Ideally, samples would be randomly sampled from the posterior, but in reality they are correlated (which manifests itself as structure within these trace plots). This example is for just a single MCMC chain, but multiple chains can be represented by lines with different colours. A: This vertical line represents the end of the burn-in period, B: The horizontal dashed line indicates the parameter value used to simulate the data (if applicable), C: Under settings, the burn-in period can be altered or the dashed line removed.

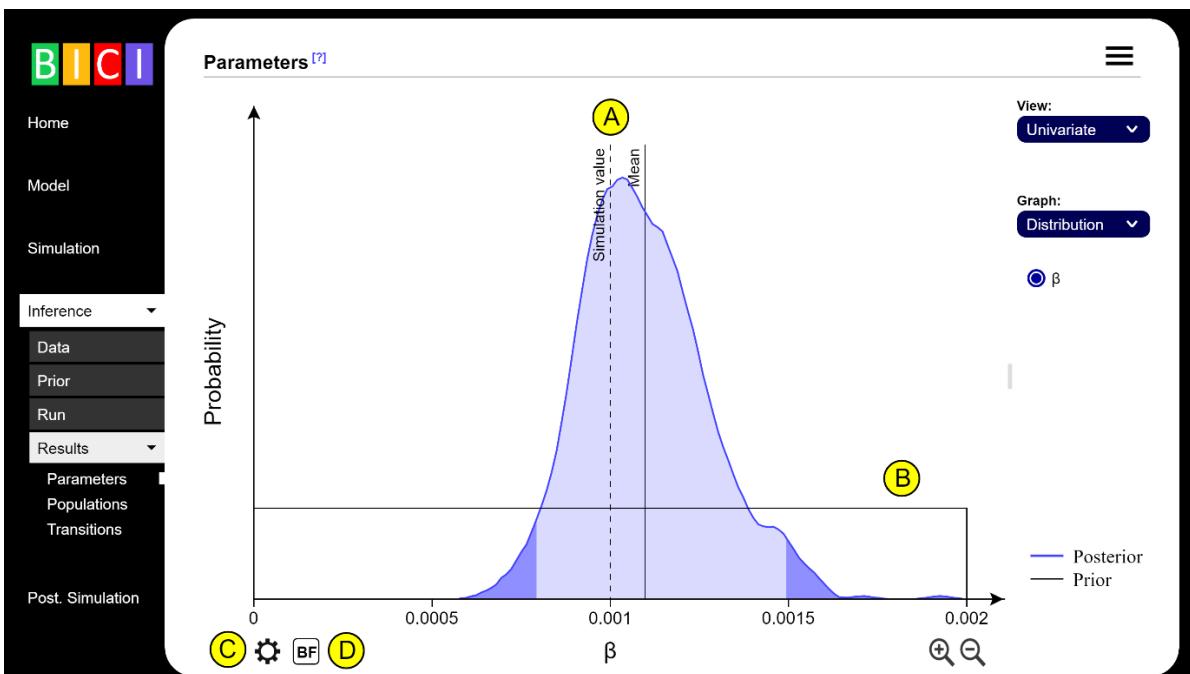


Figure C2 – Distribution plot. This shows a posterior distribution plot for a selected parameter (the darker blue areas indicate the 2.5% tails). A: The value used to simulate the data (if applicable), B: The prior distribution (solid black line), C: Under settings, either kernel density estimation (with a triangular kernel of specified size), or binning (with specified bin number) can be selected, D: Calculate the Bayes factor (see §4.4.6).

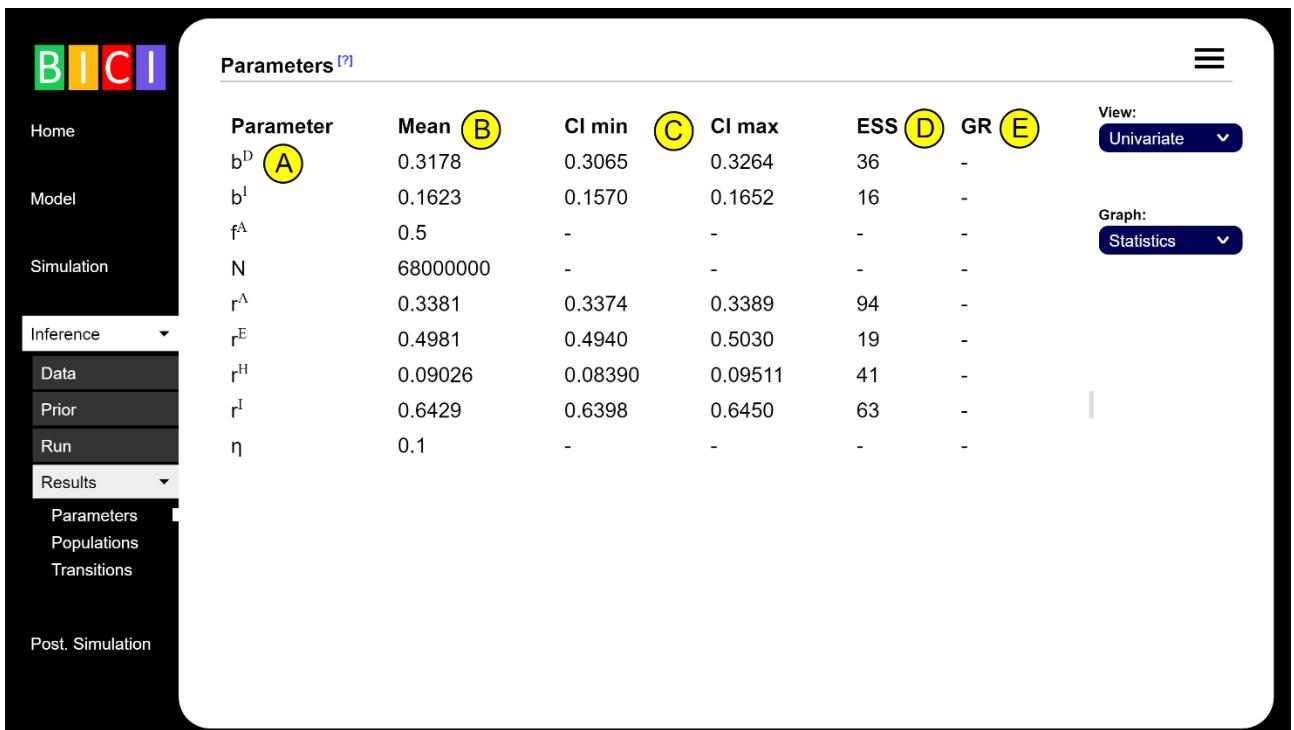


Figure C3 – Statistics. Provides posterior statistics for model parameters. A: Parameter name (this example shows univariate parameters, but for multivariate parameters different rows correspond to different elements), B: Posterior mean, C: 95% credible interval (note, this is unspecified for constant parameters), D: Effective sample size (ESS), E: Gelman-Rubin statistic (GR), not shown in this particular case because only a single MCMC chain was run. See §4.4.7 for details on ESS and GR.

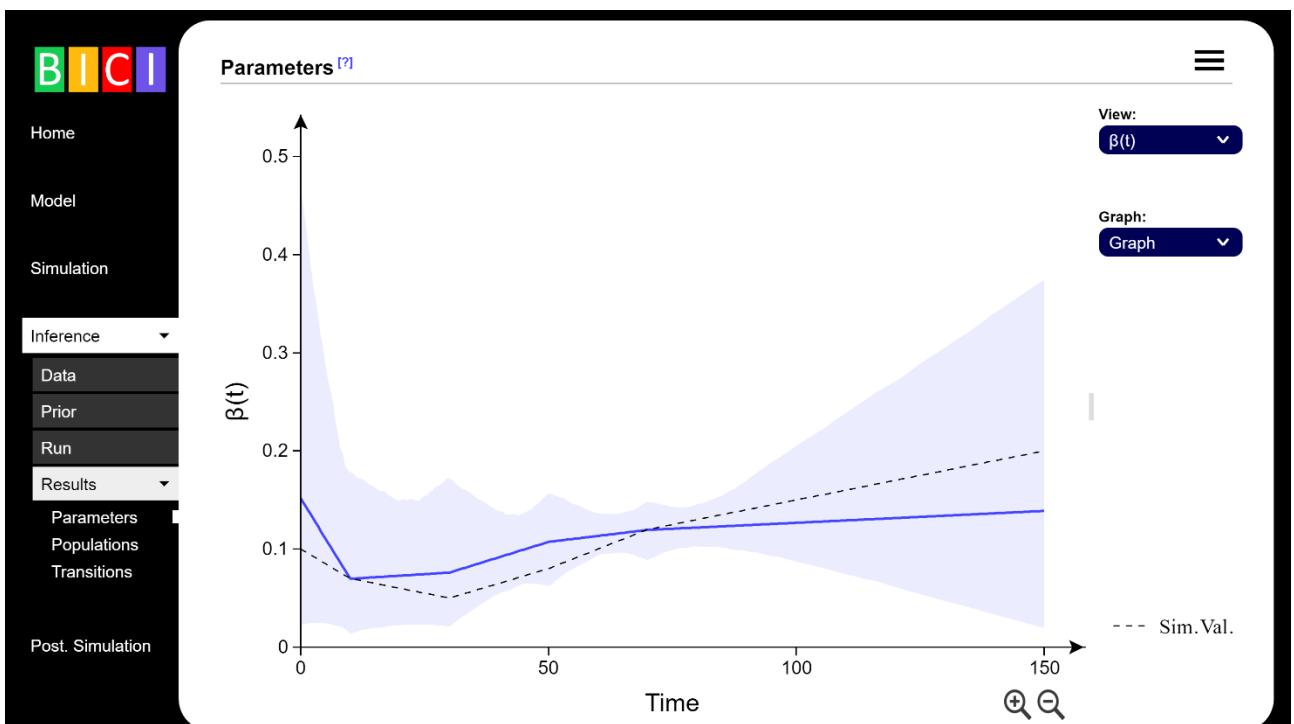


Figure C4 – Graph. This shows a graph representing time variation in spline parameter $\beta(t)$. The dashed line gives the values used for simulation. The blue line provides the posterior mean (with shading representing 95% credible intervals).

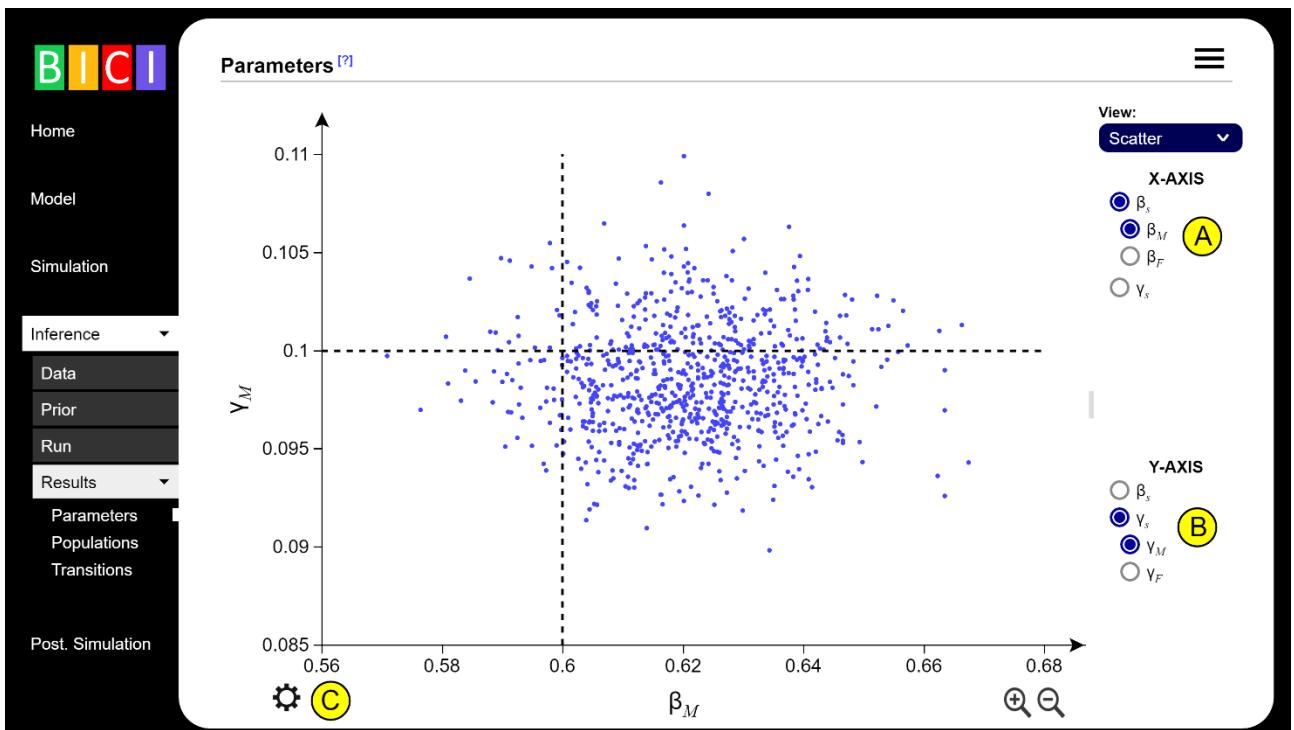


Figure C5 – Scatter plot. This shows a scatter plot for two model parameters, based on different posterior samples. A: Select one parameter on the x-axis, B: Select another parameter on the y-axis, C: Under settings, the simulated values (shown by the dashed lines) can be switch on or off.

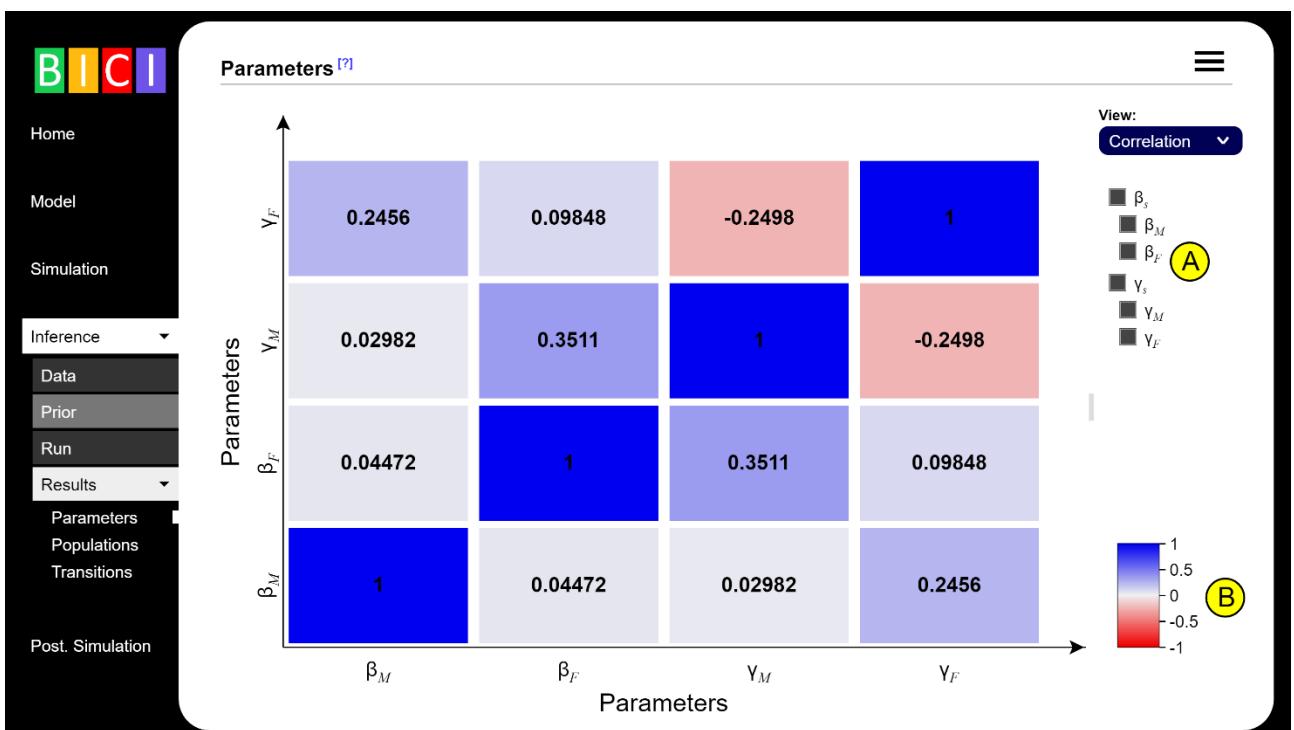


Figure C6 – Correlation matrix. This shows posterior correlations between different parameters in the model (as measured by the Pearson correlation coefficient) 1=完全 correlated, -1=完全 anti-correlated. A: Select which parameters appear in the matrix, B: Key indicating level of correlation (blue for correlated and red for anti-correlated).

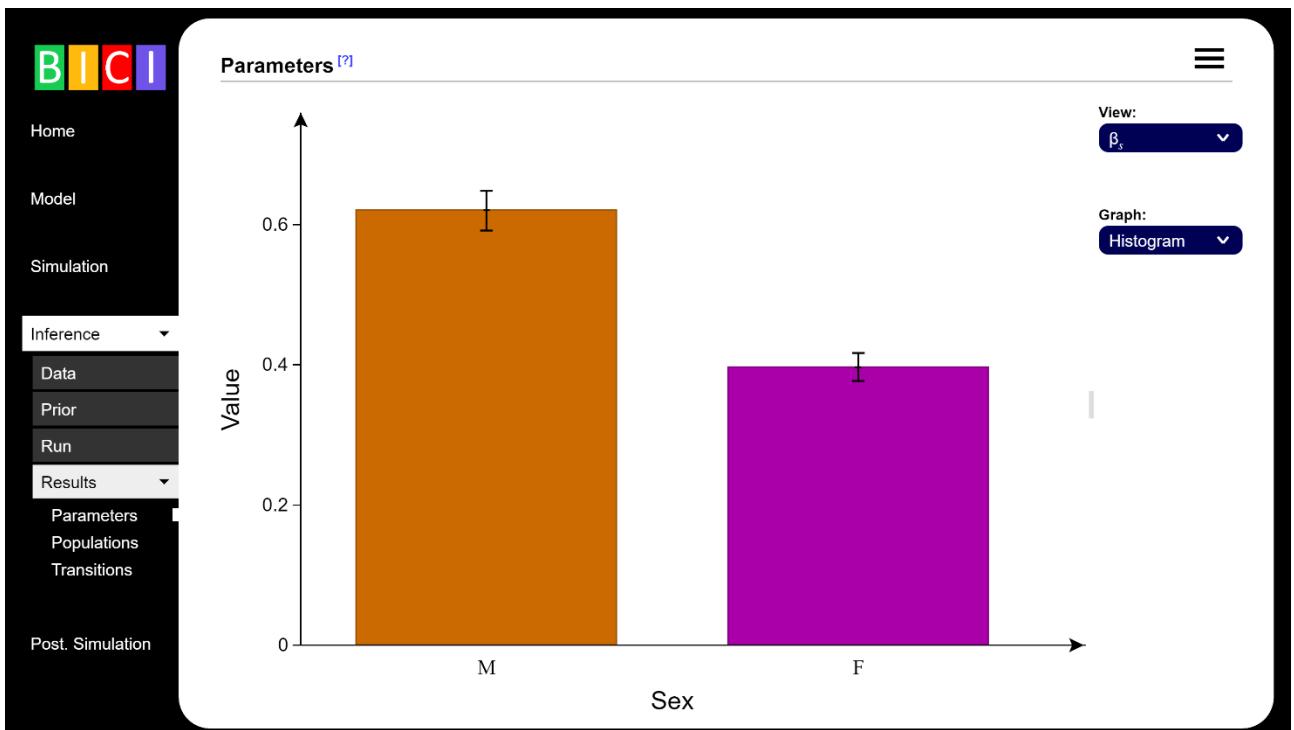


Figure C7 – Histogram. This can be used to visualise vectors, where each bar corresponds to a compartment within a classification. The error bars represent 95% credible intervals in the posterior. Time-varying histograms can also be plotted, with animation controls as in Fig. D3.

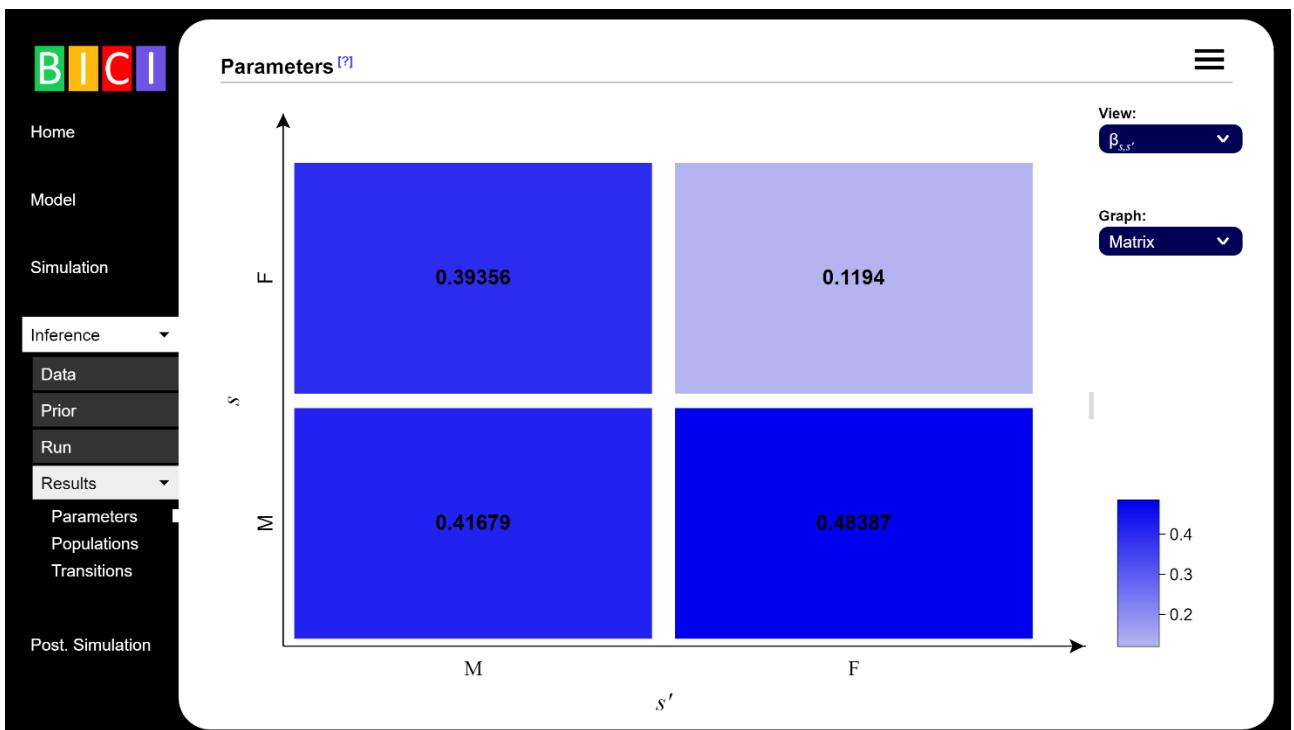


Figure C8 – Matrix. This shows the posterior mean for elements of a matrix. Time-varying matrices can also be plotted, with animation controls as in Fig. D3.

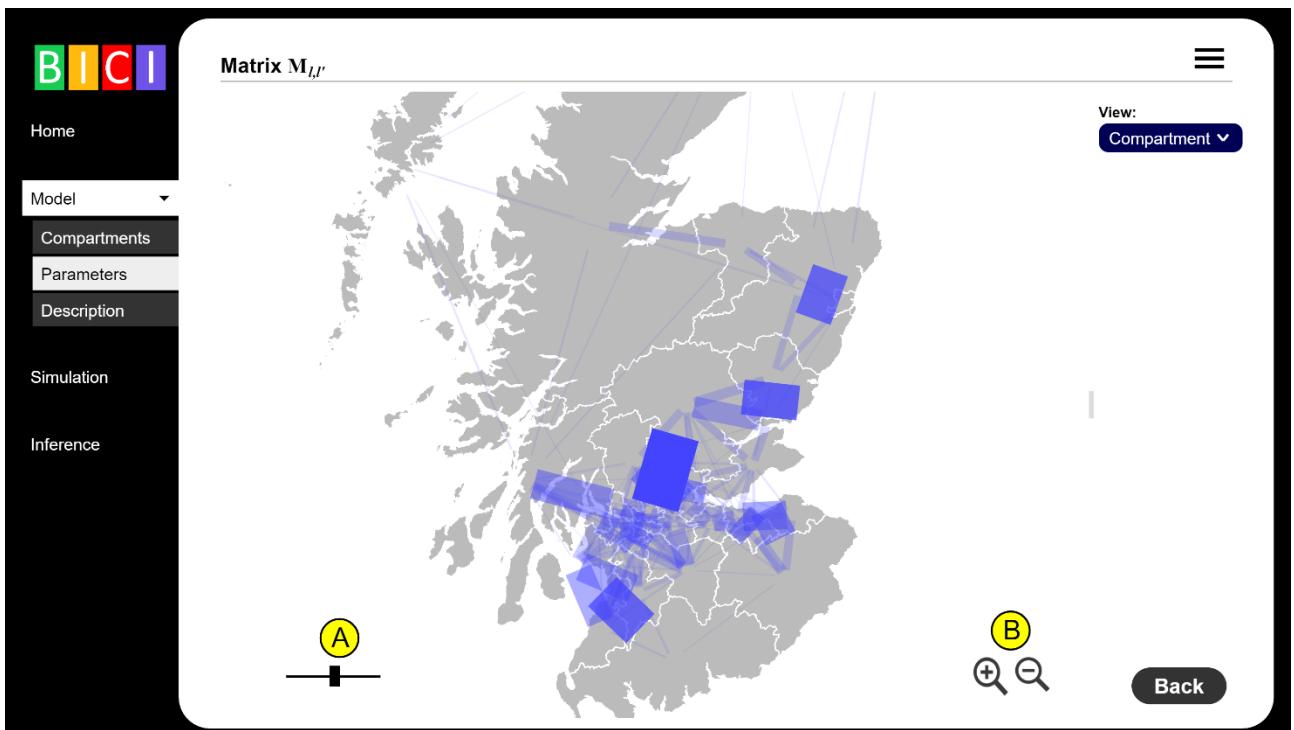


Figure C9 – Geographical matrix. When both matrix indices correspond to a geographical classification, the matrix can be plotted on the map. Here the strength of the bars indicates the size of the corresponding matrix elements. A: A slider can be used to vary the size of the bars (to help clarity) , B: The map can be zoomed in or out to see fine details. Time-varying matrices can also be plotted, with animation controls as in Fig. D3.

Appendix D: Population visualisations

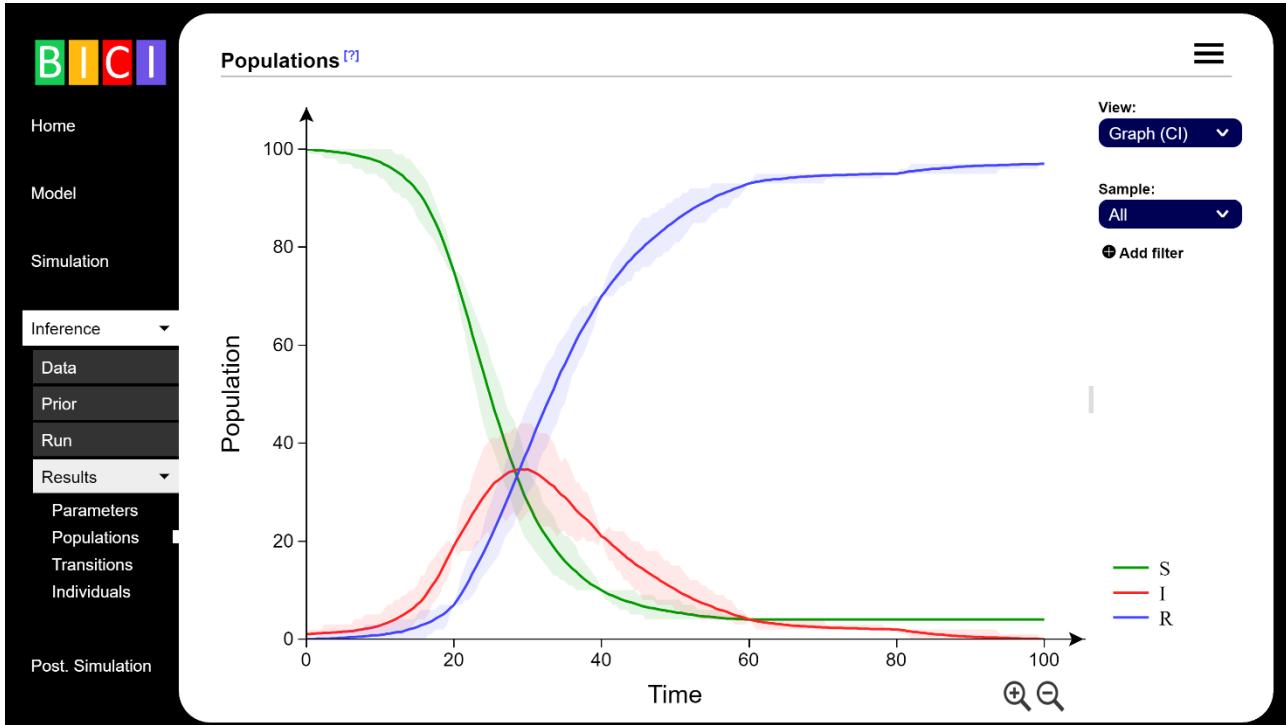


Figure D1 – Population graph. Shows the posterior distribution for the populations in different compartments as a function of time (solid lines give posterior means and shading gives 95% credible intervals). Various options on the right-hand menu are described in §4.4.2.

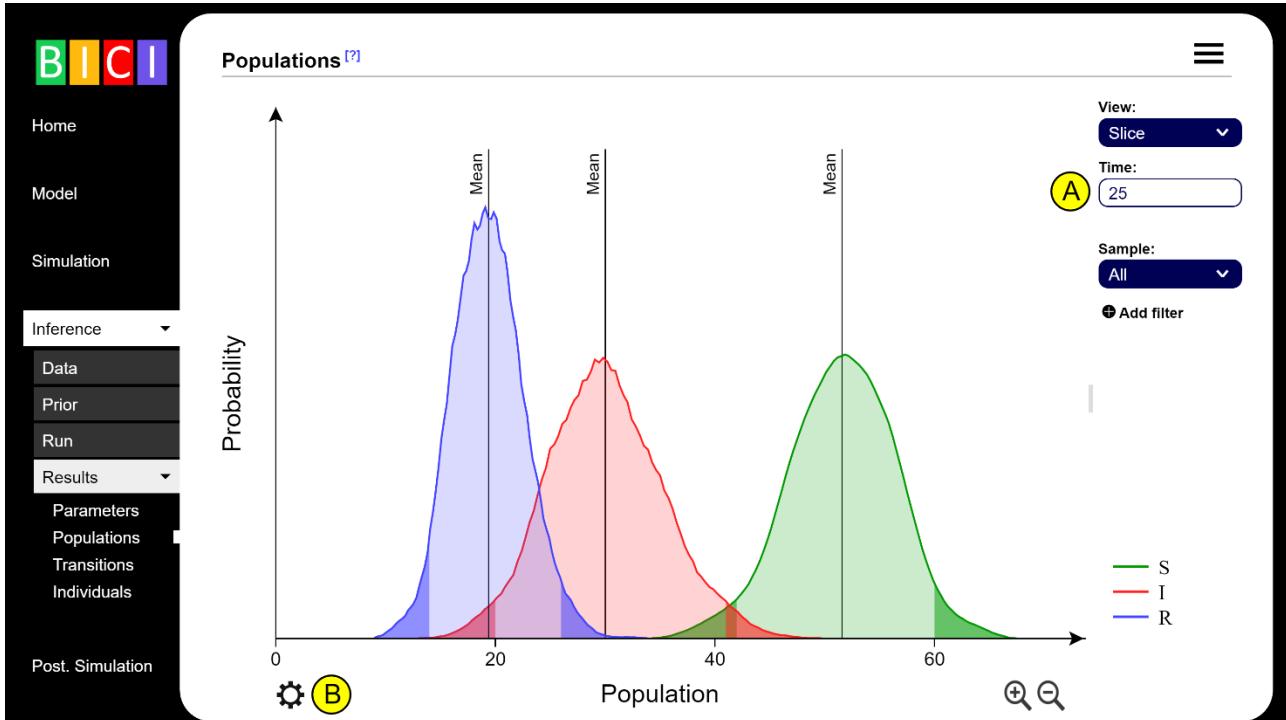


Figure D2 – Time slice. Provides the posterior distribution for populations at a specified time slice. A: Select the time, B: Under settings, either kernel density estimation (with a triangular kernel of specified size), or binning (with specified bin number) can be selected. Various options on the right-hand menu are described in §4.4.2.

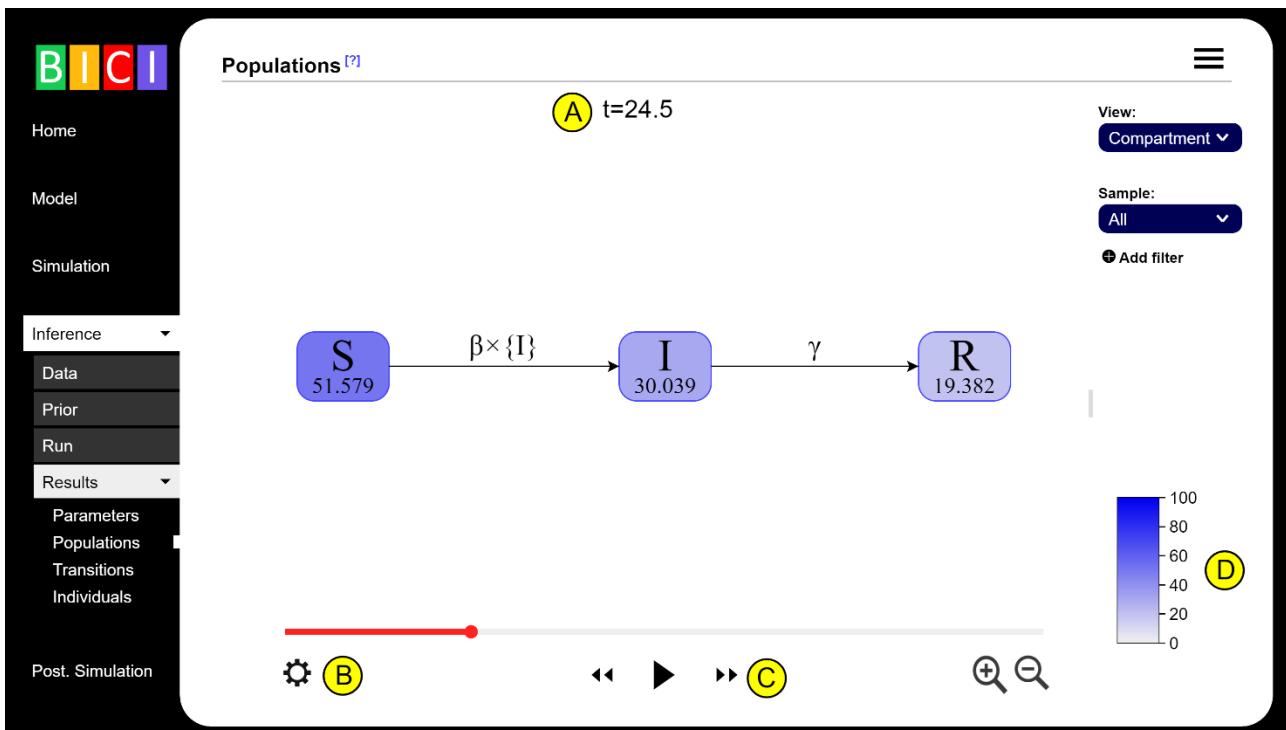


Figure D3 – Compartment plot. Shows an animation giving the posterior mean for the compartmental populations as a function of time. A: Shows the current time, B: Settings allow for the animation speed to be altered, C: The animation can be played, or stepped forwards or backwards, one frame at a time, D: Key for colour on compartments. Various options on the right-hand menu are described in §4.4.2.

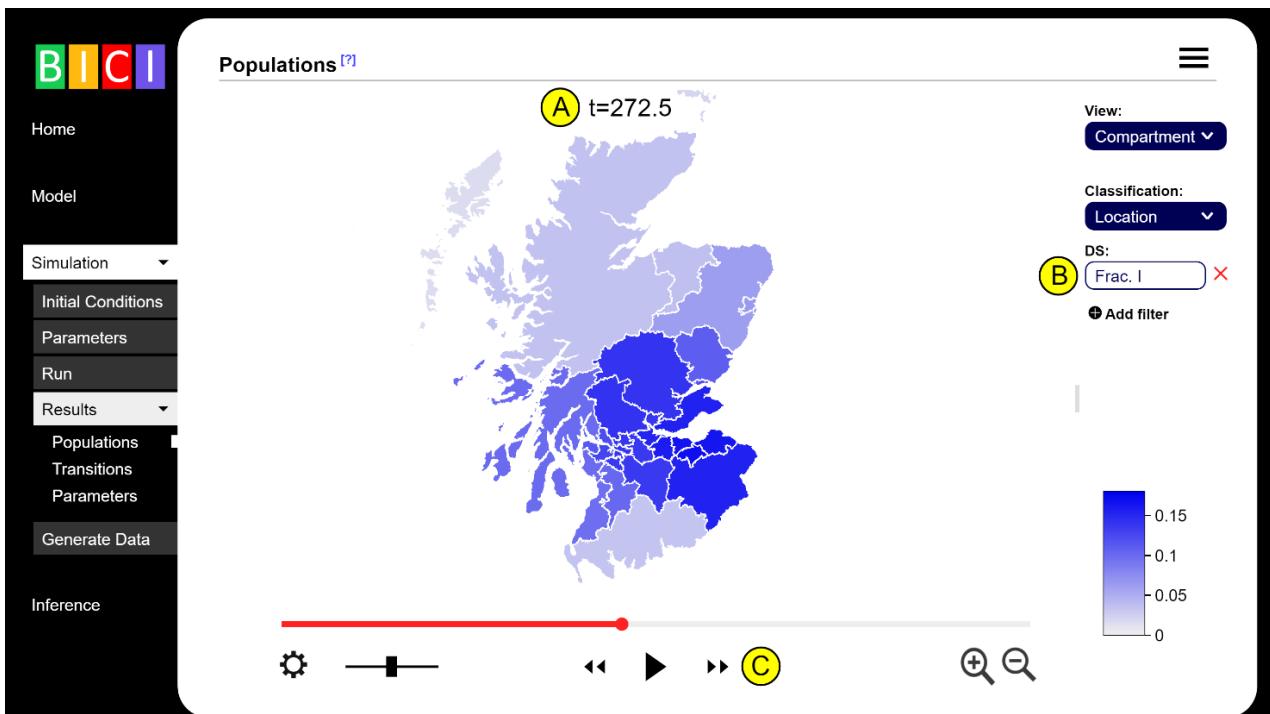


Figure D4 – Geographical compartment plot. When compartmental plots are made using geographical coordinates, they can be shown on maps. This shows an animation giving the posterior mean for the compartmental populations as a function of time. A: Shows the current time, B: This filter shows the fraction of infected individuals (*i.e.* the prevalence), C: The animation can be played, or stepped forwards or backwards, one frame at a time. Various options on the right-hand menu are described in §4.4.2.

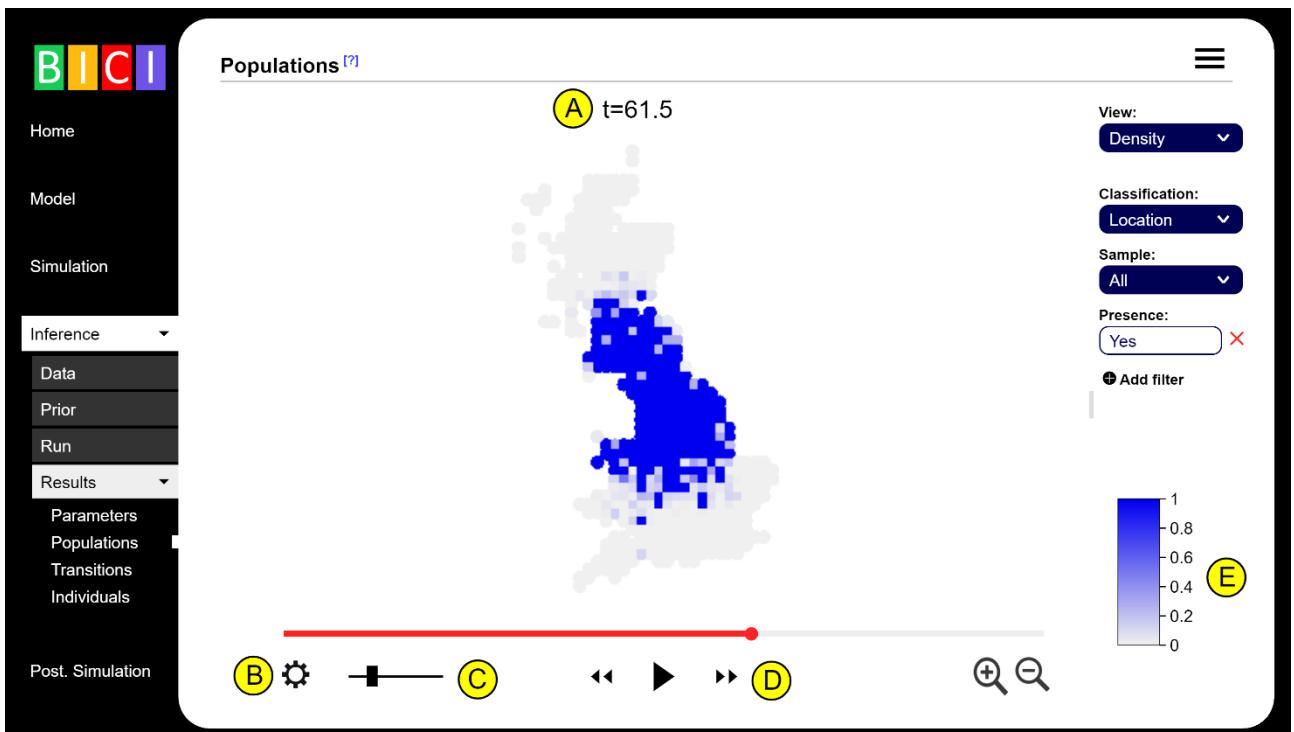


Figure D5 – Density plot. A density plot is like a compartment plot, but here circles are used (when overlapping they are clipped using Voronoi tessellation). This type of plot can be good for viewing the spread of disease over geographical points (*e.g.* farms). A: Shows the current time, B: Under settings the animation speed can be changed, C: This slider changes the size of the circles (which can be useful to improve the spatiotemporal visualisation), D: The animation can be played, or stepped forwards or backwards, one frame at a time, E: Key which relates the circle colours to population size. Various options on the right-hand menu are described in §4.4.2.

Appendix E: Transition visualisations

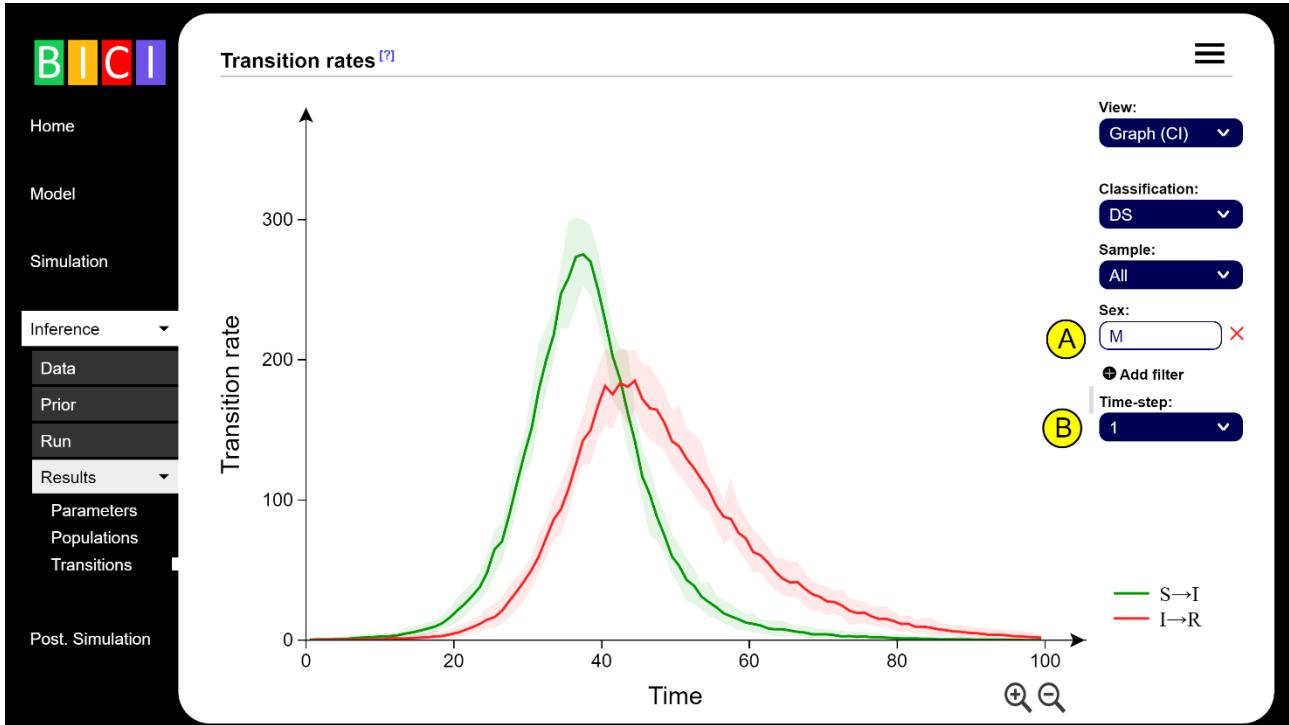


Figure E1 – Transition rate plot. This plots transition rates as a function of time (solid lines represent posterior means and shading represents 95% credible intervals). A: Population filters can be applied, in this case specifying males only (see §4.4.2 for details), B: The time-step can be altered (making this larger results in smoother plots, but reduces temporal resolution).

Appendix F: Individual visualisations

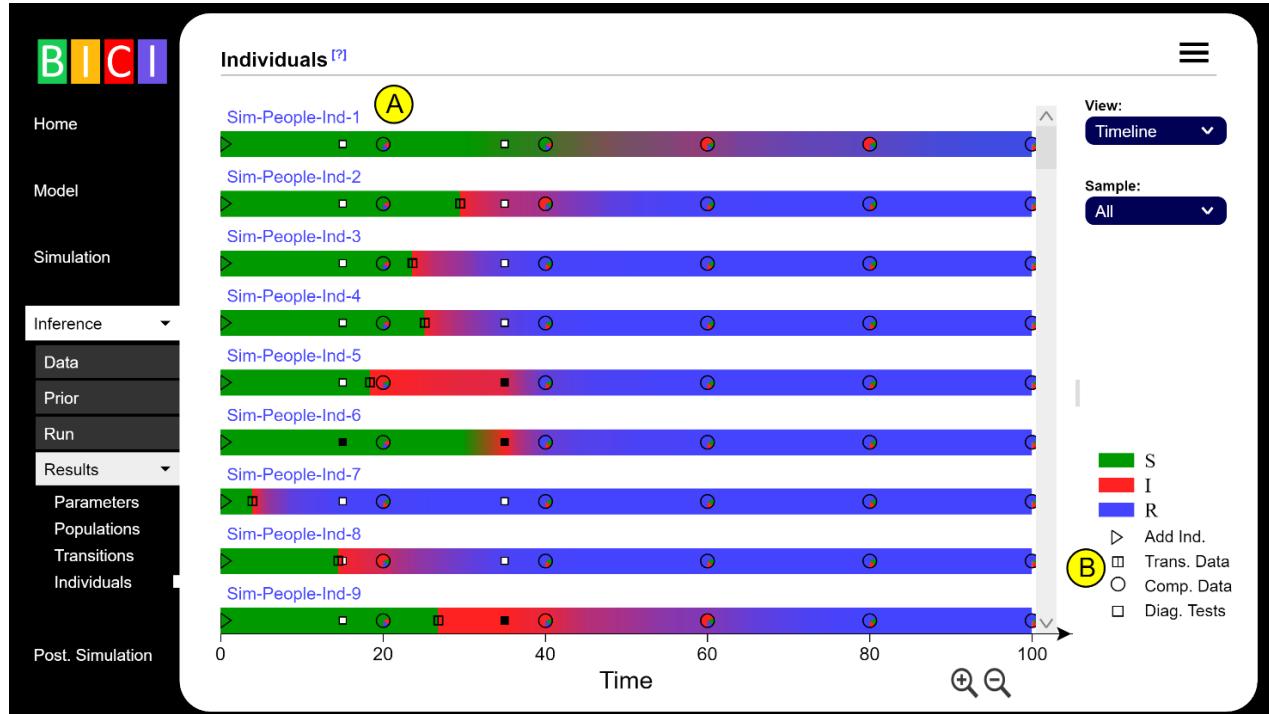


Figure F1 – Individual timelines. This summarises individual posterior state information. Each individual in the system has its own timeline. Compartments are colour coded (S=green, I=red and R=blue) according to a key. The gradations in colour reflect the mixing of compartmental colours in proportion to the posterior mean (e.g. a strongly red colour indicates that the individual is highly likely to be in the ‘I’ compartment). A: Timeline for individual with ID “Sim-People-Ind-1”, B: Different symbols represent different individual data types: right triangles show the addition of an individual, double rectangles denote transition data, circles represent compartmental data (where the small pie chart indicates the probability of being in the different compartments) and squares represent diagnostic test results (black/white indicating +ve/-ve).

The figure shows the BICI software interface with the 'Individuals' tab selected. The left sidebar includes 'Add Ind.' and 'Compartment: DS' columns. The 'Add Ind.' column lists times for each individual, and the 'Compartment: DS' column lists compartmental definitions. A legend on the right defines symbols: right triangles for 'Add Ind.', double rectangles for 'Trans. Data', circles for 'Comp. Data' (with a pie chart indicating compartmental probabilities), and squares for 'Diag. Tests' (black for +ve, white for -ve). A yellow circle highlights the 'Add Ind.' column.

ID	Add Ind.	Compartment: DS	Diag. Test: DS S
Sim-People-Ind-1	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-2	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-3	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-4	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-5	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:+
Sim-People-Ind-6	t=0	t=20 t=40 t=60 t=80 t=100	t=15:+ t=35:+
Sim-People-Ind-7	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-8	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-9	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:+
Sim-People-Ind-10	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-11	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-12	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:+
Sim-People-Ind-13	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-14	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-15	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-
Sim-People-Ind-16	t=0	t=20 t=40 t=60 t=80 t=100	t=15:- t=35:-

Figure F2 – Table. All individual data and properties are collected together into a table.

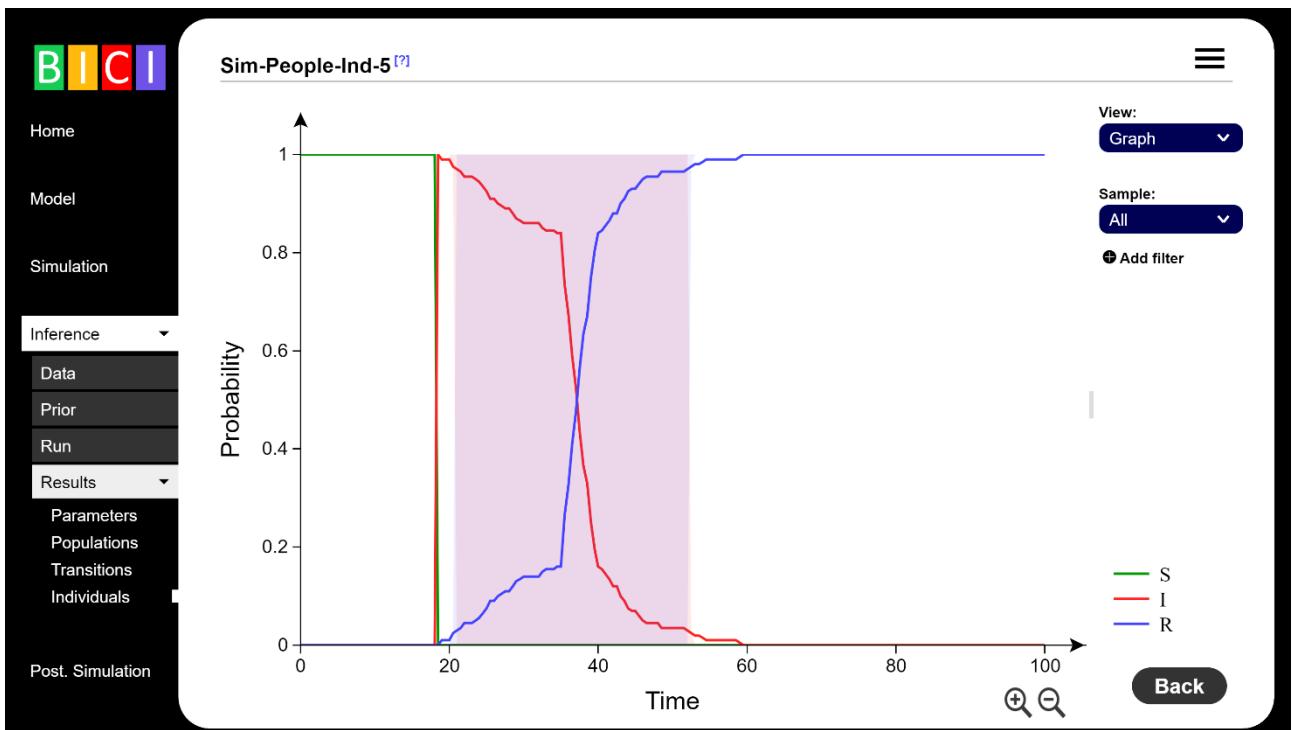


Figure F3 – Graph of individual compartmental probabilities. After an individual has been selected (Fig. F1A), this graph shows how the probability the individual is in the different compartments changes as a function of time (solid lines represent posterior means and shading represent 95% credible intervals).

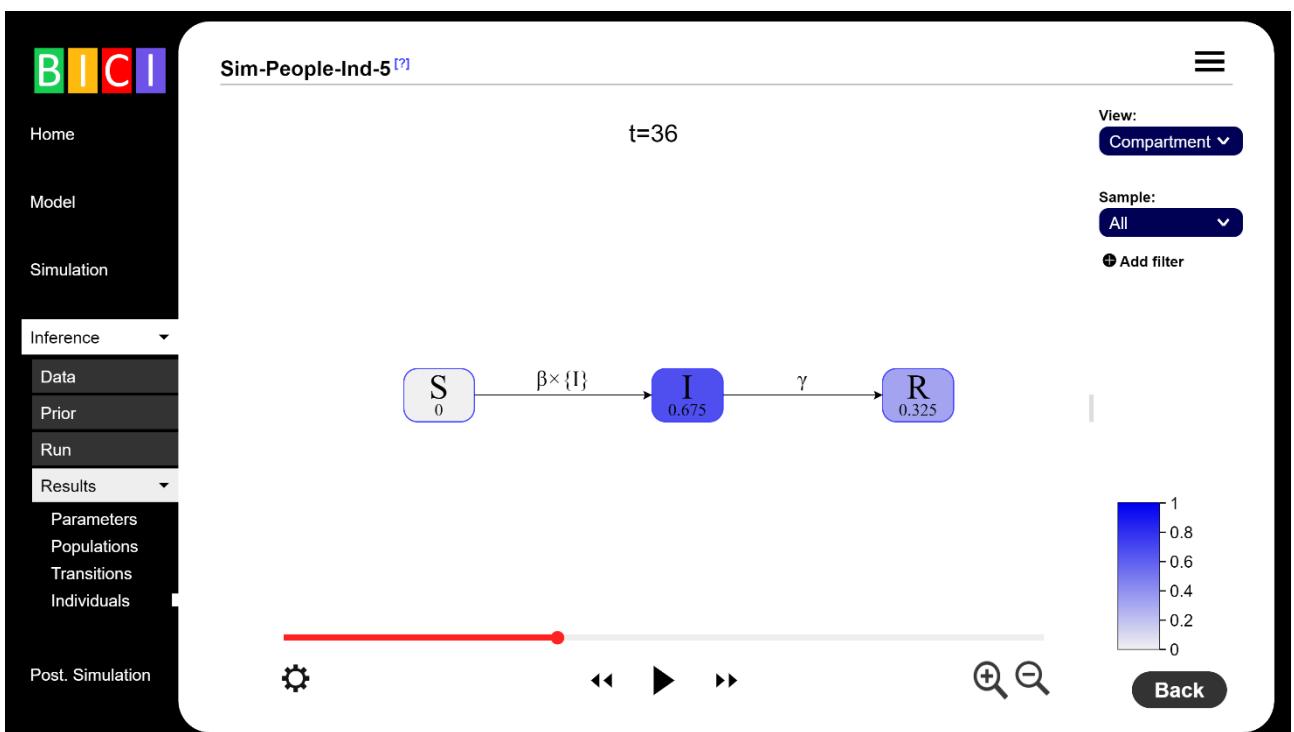


Figure F4 – Individual compartment probabilities. After an individual has been selected (Fig. F1A), this animation shows how the mean posterior probability that the individual is in the various compartments changes as a function of time. An analogous “density” plot is also available (see Fig. D5).

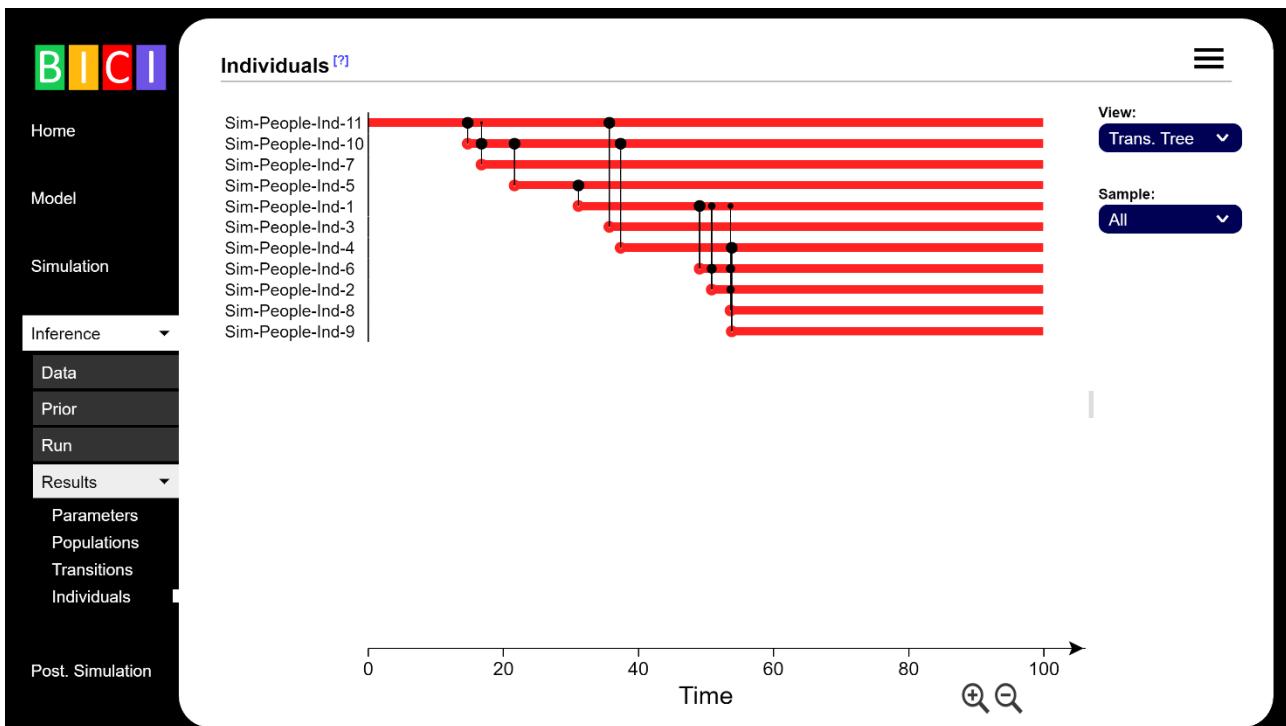


Figure F5 – Transmission tree. This shows who infects whom. Each individual has a time line. The red lines indicate the time period over which individual are infected. The black bars indicate the most likely individual and timing that caused the infection (with the size of the black circles indicating certainty).

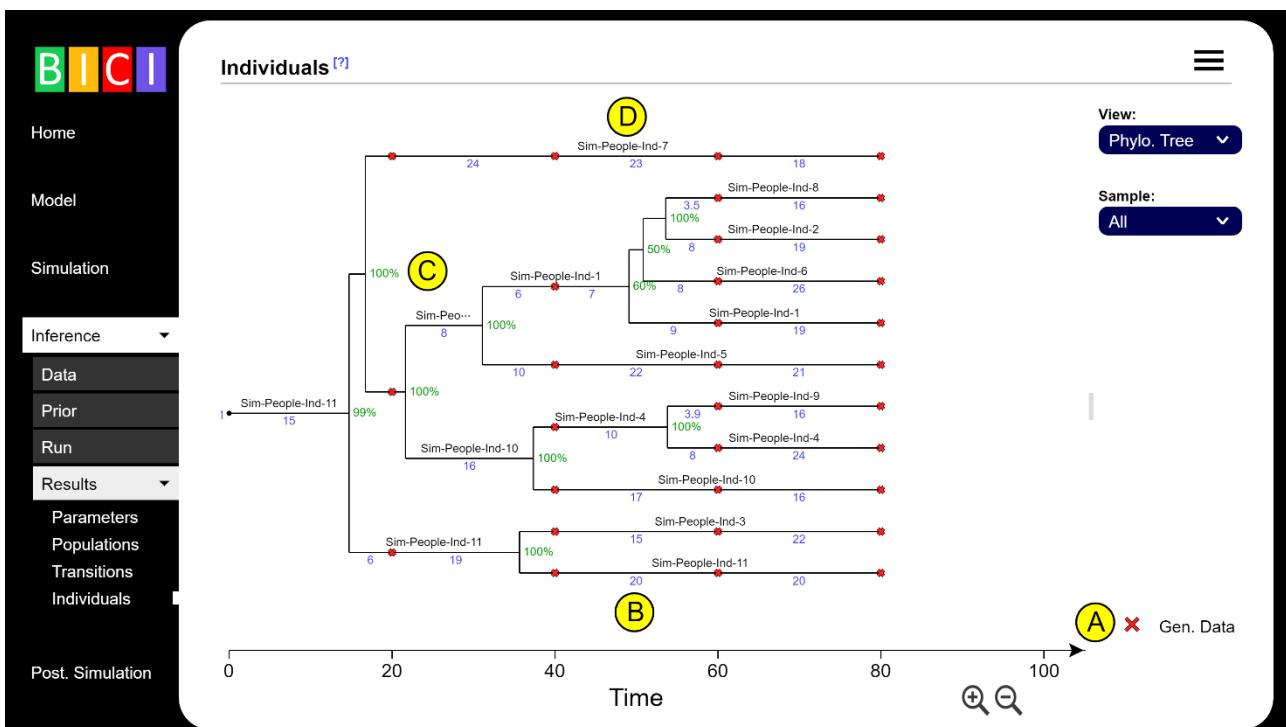


Figure F6 – Phylogenetic tree. This shows the relationship between the observed genetic data. A: The red crosses indicate the genetic observations of the pathogen, B: The blue figures represent the mean posterior number of mutations down a line segment, C: Green percentages show the posterior probability that this particular branching point occurs, D: Labels indicate the most likely pathogen host individual.