An Exploratory Evaluation into the Effect of Data Availability and Indicator Coverage on Parameter Estimates using Hamiltonian Monte Carlo

[redacted]

[redacted]

Abstract

There are typically two important questions addressed via the model calibration process: (1) does the time series of the fitted model match to the historical data; and (2) can reliable parameter estimates be inferred that are bounded within credible intervals. The evolution of Markov Chain Monte Carlo (MCMC) methods provide powerful methodological and computational frameworks for parameter estimation, and recent studies confirm the value of the Hamiltonian Monte Carlo approach for system dynamics models. This paper addresses an important research question for the calibration process, namely: what is the impact of data availability and indicator coverage on parameter estimation. It presents a 3 x 7 factorial study based on an SEIR model with cases, hospitalisations, and deaths. An exploratory analysis is presented, where all models converge. Our results highlight differences for a number of posterior distributions calculated, depending on the data availablity, and the set of indicators used.

1 Introduction

In system dynamics, models must be grounded in data if modellers are to provide reliable advice to decision makers, and as part of this, robust parameter and confidence interval estimation is crucial [1]. There are a number of computational methods that can be used to fit models and estimate parameters. These include: (1) the use of numerical optimisation combined with bootstrapping for confidence interval estimation and hypothesis testing in system dynamics models [2]; and (2) Markov chain Monte Carlo (MCMC) approaches to estimate the posterior distribution for parameters, a statistical approach that has gained significant adoption in the system dynamics community in recent years [3, 4, 5, 6, 7, 8, 9].

Our approach in the paper is to fit an SEIR model (based on synthetic data) using Hamiltonian Monte Carlo (HMC) [5], in order to gain insights into the how the availability of data (through epochs and indicators) impacts the overall credible intervals ranges for model parameters. Our choice of the three indicators is motivated by the benefits of having more than one indicator to calibrate a model, for example, to help migitate against reporting biases that would be a factor in real-world implemntations [10].

As part of our analysis we also introduce a new evaluation method for comparing posterior distributions, known as overlapping, and this provides useful insights into the sensitivity of parameter estimation to the availability of data.

The structure of the paper is as follows:

- We introduce the SEIR model structure and overall experimental design, which involves 21 experiments;
- We present an open-source computational framework, mainly written in R, and also using the Stan HMC package;
- We present a detailed set of results, including: convergence tests, model fits, descriptive summaries of fits; 95% quantile analysis; and, overlapping analysis for all 21 experiments.
- The appendices include: (1) a sample stan file for inference of cases, hospitalisations, and deaths; (2) the main script file for running experiments; (3) the configuration file for the experiments; and (4) the code comparing the posterior distributions.

2 Model Structure and Experimental Design

2.1 Model Structure

The model used is an extension the well-known deterministic SEIR structure [11], and is shown in Figure 1. People start out as being susceptible to a novel pathogen, and with the introduction of *patient zero* to the infectious (I) stock, the contagion loop is activated. People then move to an exposed (E) stock, where they do not contribute to the force of infection (λ), before entering an infectious state. Once in an infectious state people contribute to the force of infection (λ), before exiting via a first order exponential delay structure. A certain proportion of those infected have symptoms (clinical fraction), and a proportion of infectious are hospitalised, while the remainder recover. Hospitalisation is modelled as a third order delay structure, and 90% of those hospitalised recover, while 10% do not and move to the deaths stock (D). Three stocks (TC, TH, and TD) are used to record the cumulative numbers of cases, hospitalisations, and deaths.

Equations (1-6) for the main SEIR structure are shown below. The parameters σ and γ represent the inverse of the latent and infectious delays, c is the clinical fraction, and h is the hospitalisation fraction. The model assumes that only those who show

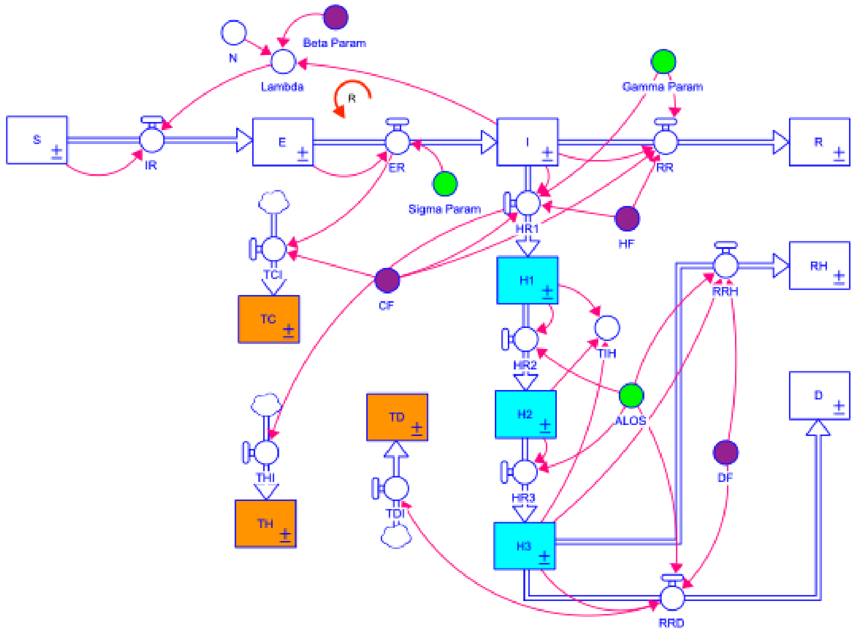


Fig 1. An SEIR Influenza Model with Hospitalisations and Deaths

symptoms can end up in the hospital stream. The force of infection λ is calculated based on product of the effective contact rate (β) with the number of infectious (I), divided by the total population (N).

$$\dot{S} = -\lambda S \tag{1}$$

$$\dot{E} = \lambda S - \sigma E \tag{2}$$

$$\dot{I} = \sigma E - (1 - ch)\gamma I - ch\gamma I \tag{3}$$

$$\dot{R} = (1 - ch)\gamma I \tag{4}$$

$$\lambda = \beta \frac{I}{N} \tag{5}$$

$$\beta = 1.0 \tag{6}$$

The hospitalisation stream is modelled via equations 7-13. It involves a straightforward sequence of stocks that model people staying in hospital, with the average length of stay (L) set to 10 days. The hospitalisation rate is governed by the fraction ch exiting the infectious stock, and therefore for this model, that will evaluate to $0.6 \times 0.1 = 0.06$. Upon exiting hospital, $(1 - d)$ move to the stock R_H, while the fraction d move to the stock D.

$$\dot{H_1} = ch\gamma I - \frac{H_1}{L_1} \tag{7}$$
$$\dot{H_2} = \frac{H_1}{L_1} - \frac{H_2}{L_2} \tag{8}$$
$$\dot{H_3} = \frac{H_2}{L_2} - d\frac{H_3}{L_3} - (1-d)\frac{H_3}{L_3} \tag{9}$$
$$\dot{R_H} = (1-d)\frac{H_3}{L_3} \tag{10}$$
$$\dot{D} = d\frac{H_3}{L_3} \tag{11}$$
$$L_1 = L_2 = L_3 = \frac{L}{3.0} \tag{12}$$
$$L = 10 \tag{13}$$

As part of the inference process, we need to generate incidence rates for cases, hospitalisations, and deaths, and these numbers are initially recorded as cumulative (stock) values (14-16).

$$\dot{T_C} = c\sigma E \tag{14}$$
$$\dot{T_H} = ch\gamma I \tag{15}$$
$$\dot{T_D} = d\frac{H_3}{L_3} \tag{16}$$

The difference values are calculated as follows (17-19), and these values are then used as part of the data fitting process.

$$X_C(t+1) = T_C(t+1) - T_C(t) \tag{17}$$
$$X_H(t+1) = T_H(t+1) - T_H(t) \tag{18}$$
$$X_D(t+1) = T_D(t+1) - T_D(t) \tag{19}$$

The statistical distributions (count data) used for the fitting process are shown in equations (20-22), and this is appropriate given that the synthetic data generated for the three indicators in based on the negative binomial distribution. Note that, depending on the experiment, a subset of these will be used.

$$Y_C \sim Nbin(X_C, \phi_1) \tag{20}$$
$$Y_H \sim Nbin(X_H, \phi_2) \tag{21}$$
$$Y_D \sim Nbin(X_D, \phi_3) \tag{22}$$

The model parameters for the experiments are shown below. Two of these are based on the literature relating to pandemic influenza [12], and the remaining values are arbitrary choices used as part of the experimentation process. Four of these parameters (β, c, d and h) will be estimated as part of the MCMC inference process.

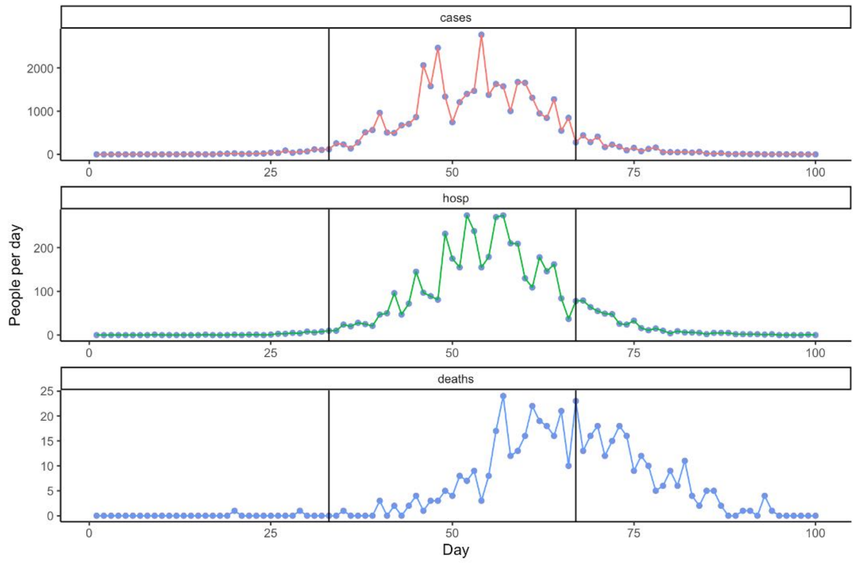


Fig 2. Sample synthetic data generated for the experiments

Name	Symbol	Value	Units	Source
Latent Duration	σ^{-1}	2.0	*Days*	Vynnycky et al. [12]
Infectious Duration	γ^{-1}	2.0	*Days*	Vynnycky et al. [12]
Effective Contact Rate	β	1.0	$Days^{-1}$	Model estimate
Clinical Fraction (CF)	c	0.60	*Dimn*	Model estimate
Death Fraction (DF)	d	0.10	*Dimn*	Model estimate
Hospitalisation Fraction (HF)	h	0.10	*Diml*	Model estimate
Average Length of Stay	L	10.0	*Days*	Model estimate

2.2 Experimental Design

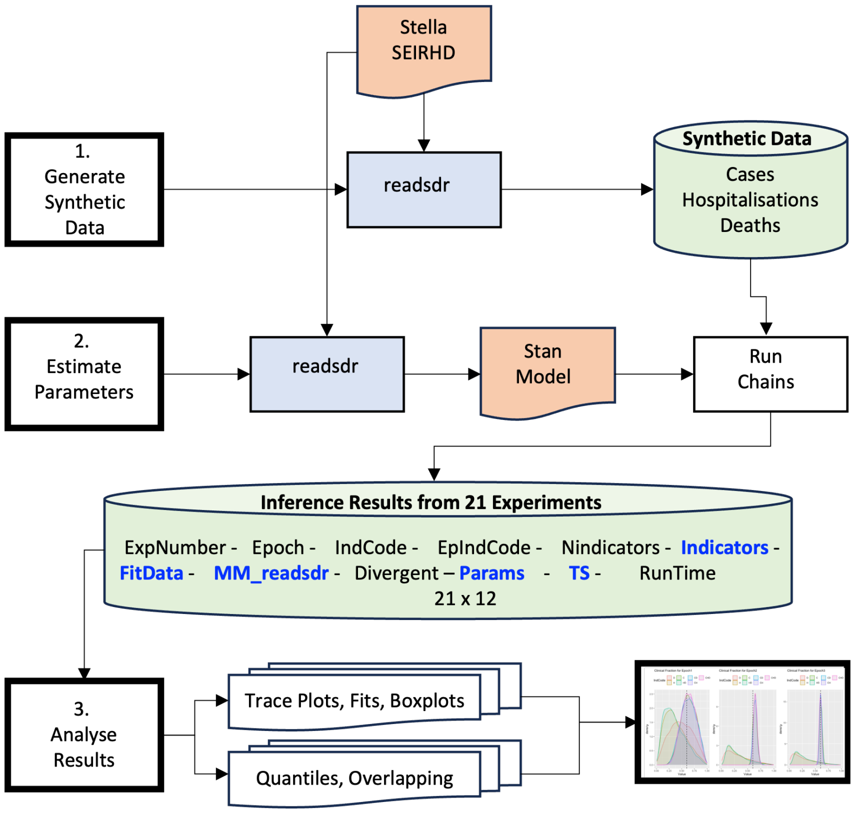
The overall aim of the experimental design is to explore the possible effect of data availability and indicator coverage on the fitting process. The SEIR model was run once, and synthetic data generated, as shown in Figure 2. This synthetic data was based on the SEIR model values generated (17-19), and different levels of variation were introduced via the negative binomial dispersion parameter, where values of 10, 20 and 40 were used respectively for cases, hospitalisations, and deaths.

With three data sets available, a factorial-type set of experiments was designed based on:

- Three epochs of equal length, one for the start of the epidemic, one for the middle duration, and one for the end. These are shown on the figure, and capture different dynamic phases (e.g. exponential growth, inflection point, peak, and decline) for each indicator.

- Seven combinations of indicators based on the indicators cases (C), hospitalisation (H), and deaths (D). This yielded the subsets: (CHD,CH,CD,HD,C,H,D).

The combination of epochs (3) and indicators (7) were then combined to run 21 experiments, which involve fitting the parameters to different data sources. The computation framework for this process is now described.



Stella SEIRHD
1. Generate Synthetic Data
readsdr
Synthetic Data
Cases Hospitalisations Deaths
2. Estimate Parameters
readsdr
Stan Model
Run Chains
Inference Results from 21 Experiments
ExpNumber - Epoch - IndCode - EpIndCode - Nindicators - Indicators - FitData - MM_readsdr - Divergent – Params - TS - RunTime
21 x 12
3. Analyse Results
Trace Plots, Fits, Boxplots
Quantiles, Overlapping

Fig 3. Overall framework for experiments

3 Computational Framework

Figure 3 captures the overall computational framework used to configure the experiments, generate the results, and produce the analysis. In order to execute this workflow, which is open-source and designed using R [13, 14], a number of additional components were required:

- Stan [15], a statistical modelling platform, which provides an interface to perform Bayesian inference via the No-U-Turn-Sampler (NUTS). Stan models can contain ordinary differential equations, and is used to perform inference for deterministic models of infectious disease [5, 6, 7]
- `cmdstanr` [16], which is a lightweight interface to Stan for R users.
- `readsdr` [17], a package that automatically converts XMILE files from Stella and Vensim to Stan code.
- R's tidyverse packages, specifically `dplyr` for data manipulation, `ggplot2` for visualisation and `tidyr` for nesting data frames, a feature that allows the results to be conveniently organised into a 21 by 12 table.

The overall framework is divided into three main functions.

- **Generate Synthetic Data**, which runs the SEIR model using the package `readsdr` for one instance, and is based on the differencing of the three indicator stocks (see equations 13-15). The negative binomial distribution is used to generate random count variables based on the model outputs. For this experiment, the dispersion parameters selected are (10, 20, 40) for (Cases, Hospitalisations, and Deaths). A sample run from the synthetic data generation process is shown in Figure 2.

- **Estimate Parameters**, which performs the fitting process. For each experiment, this will (1) use `readsdr` to generate a stan file, (2) run the stan file using the `cmdstanr` interface, and (3) prepare and store all of the results in a single multidimensional data structure. This structure will contain one row for each experiment, and encapsulate variables such as the number of indictors, the measurement model, the data used for the calibration, the posterior samples for each parameter (4,000 each), the time series output for each model run, and the duration of the run. Given the computational resources needed to run the fitting, an advantage of storing all the results in a database (RDS file used) is that the analysis stage can then be conducted independently.

- **Analyse Results**, which provides a number of scripts to generate a range of results to support analysis. These include: trace plots to explore convergence; time series showing the fits; boxplots highlighting the parameter distributions across all 21 experiments; quantile analysis to show the 95% credible intervals from the inference process; and overlap analysis to provide insights into how close the posterior distributions are for each combination of epoch and indicator.

The results are now presented in more detail.

4 Results

Overall, the inference process for the 21 experiments generates over 273MB of data, so there is a wide range of analysis that can be performed. Given that the overall goal is to explore possible differences in parameter estimates, our analysis comprises the following stages:

- Presentation of convergence tests for each parameter over the four MCMC chains, for each of the 21 data configurations.

- Exploring of the model fits, to confirm that the parameter estimates generate plausible dynamic behaviour for each model.

- Boxplots to highlight the distribution of inferred parameters, and to see how the estimates line up with the original values used to construct the synthetic data sets.

- Overlapping analysis, defined as the area intersected by two or more probability density functions [18, 19], to provide a measure of how close the estimates are across each of the 21 experiments.

4.1 Convergence Tests

Our first analysis, capture in Figure 4, displays the trace plots showing how the estimates of the parameter values change over the MCMC process. Four MCMC chains are run, with 1000 iterations for the warm-up phase, and 1000 for the sampling process. The goal is to ensure that a stationary distribution is achieved for all parameters [5], and this is achieved, and confirmed by analysing the output from stan, which showed no divergences within all the samples. This chain convergence is an important property of the inference method. However, the plots can also illustrate interesting properties of the fitted parameters. Through observation, we can see the different ranges of the estimates, for example:

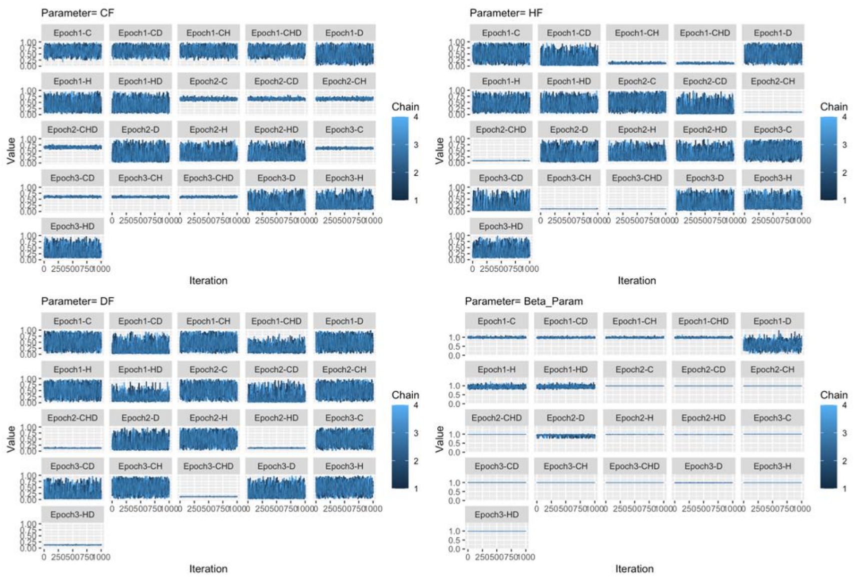


Fig 4. Exploring parameter convergence

- Experiments involving epoch one (the opening phase of the epidemic) typically have wider ranges, which is as expected, as there is less data available to inform the fitting process. For example, for the parameter `HF` (true value 0.1), the values for `Epoch1-C` seem to cover the interval from 0 to 1.

- Experiments where the three indicators were used tend to have narrower bands, which again is intuitive given that more information is available for the calibration process. Returning to parameter `HF`, we can see that its values for `Epoch1-CHD` remain much closer to the true value of 0.1.

A more detailed analysis of these parameter values will be explored through boxplots, quantile analysis and overlapping analysis.

4.2 Model fits

An important requirement for calibration is that the model provides a plausible representation of historical data, and in MCMC fitting we can also explore the time series quantiles returned as part of the posterior distribution. From a process perspective, having fits that align with historical data is needed to confirm that the model structure can generate the behaviour of interest, and these outputs also can be deployed as a confidence-building measure for modellers and clients. A sample of the fits are displayed in Figure 5, for the indicators *Cases* and *Deaths*, across three of the epochs. The mean value from the MCMC samples is also shown, and overall, on visual inspection, these look like good fits to the data.

An interesting aspect is what the fits do not show, which is the range of parameter values underling the generated time series. For example, if you take the set of plots in row 1 column 1 (epoch 1), and compare the fits for indicator *C* and indicators *CHD*, there is not an observable distinction between both, in that both fits look plausible. However, when we compare the parameters estimated for these two samples, differences emerge. This can first be explored using boxplots.

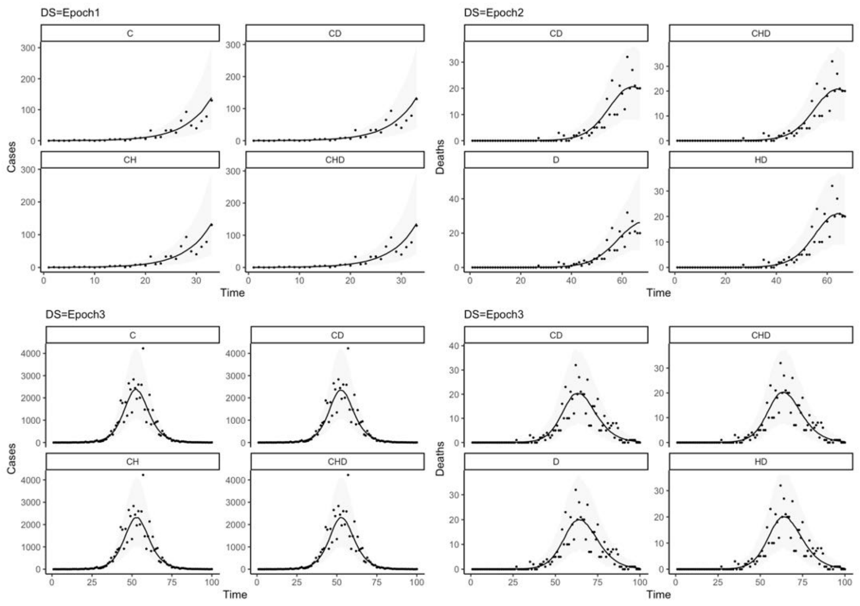


Fig 5. A sample of fits from the inference process

4.3 Boxplots of parameter estimates

The boxplot is a valuable method to summarise data, as it shows the median, the interquartile range (IQR), the location of values 1.5 times above and below the 75th and 25th percentiles, and outliers. We use the boxplot as a means to compare parameter values from all 21 experiments, and estimated across (1) each of the three epochs and (2) all of the seven combinations of data indicators. The 84 boxplots are presented in Figure 6. Our working assumption is that a narrower range indicates a more certain fit (the 95% credible intervals are shown in the following section).

We can initailly observe a number of patterns from these descriptive statistics:

- Across all parameters, the narrowest parameter estimate ranges seem to be those fits using the most indicators, and using the most data (i.e. epoch 3).
- The rate at which parameters "lose their narrowness" varies across both indicators, and epochs. For example, the parameter β within epoch 3 retains a narrow range for most indicator combinations, while the death fraction (DF) only retains a narrow range using three indicators across epoch 1 and 2.
- Therefore, this highlights a difference depending on the epochs used, and the indicators. For example, for parameter `DF` with indicators `CHD`, there is a large difference in median and the overall spread of data across the three epochs.

4.4 Quantile analyis of fitted parameters

The plots displayed in the preceding section are useful to gain an overall appreciation of the posterior distributions, and to supplement that analysis, it is important to show the 95% credible intervals. For the four parameters, this information is presented in Figure 7, and arranged in descending order by the difference between the upper and lower quantiles.

For each of the fitted parameters, we can make the following initial observations:

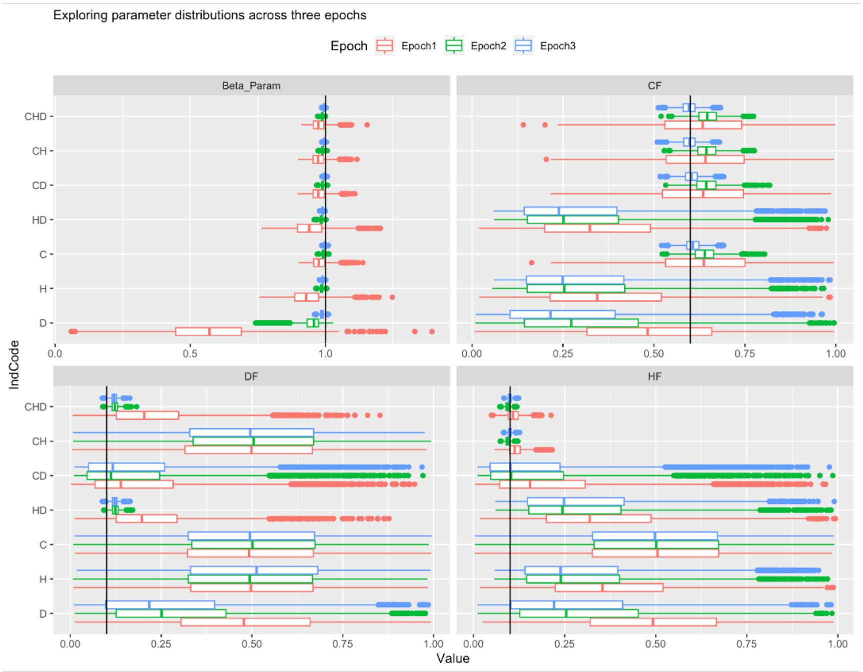
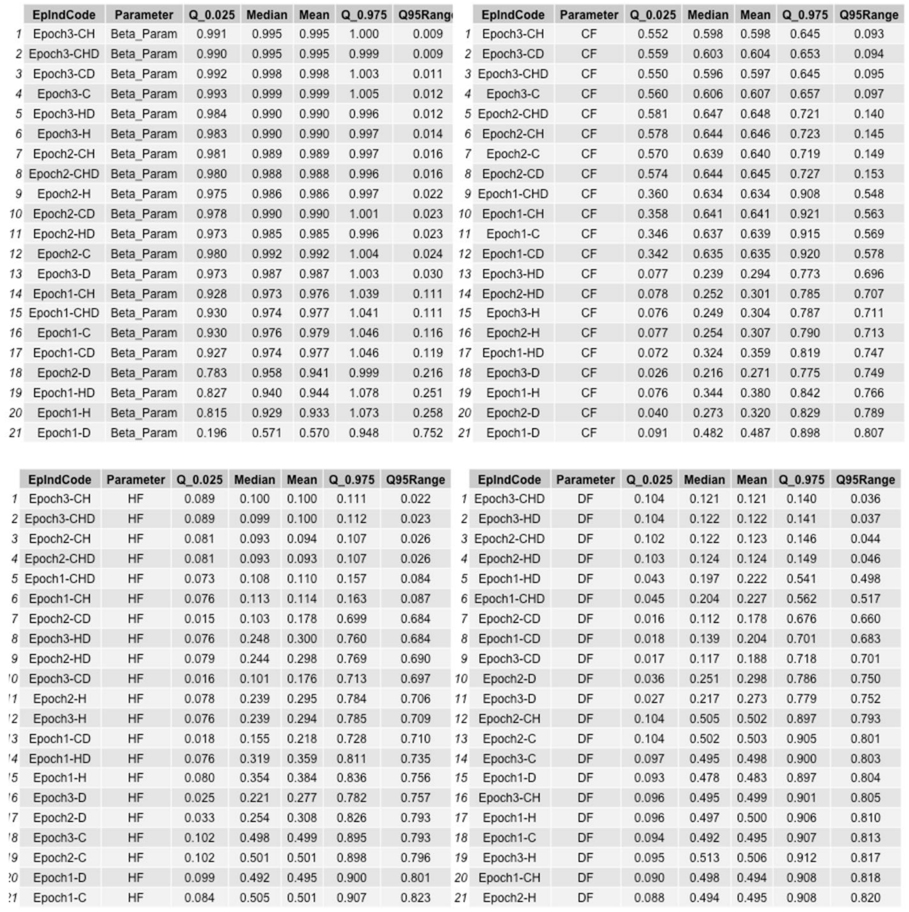


Fig 6. Box plot analysis for parameters by epoch and indicator source(s)

- *Beta_Param*, which models the effective contact rate within the population model. The true value is 1.0, and many of the 95% ranges contain this value. A shift in the range occurs for observation 14, and the experiment `Epoch1-CH`. From that point onwards, 7 of the 8 experiments relate to epoch 1, showing that epoch1 is prominent for fits with wider estimate ranges.
- *Clinical fraction (CF)*, which is the fraction of cases that are clinical, with a true value of 0.60. All of the estimates include this value, although a significant shift in the range happens with experiment 9 (`Epoch1-CHD`). Prior to that, of the eight experiments that have narrower bands, four are from epoch 2 and four from epoch 3. The four best are from epoch 3, which has the most data points used for the calibration process.
- *Hospitalisation fraction (HF)*, which is the fraction of cases hospitalised, with a true value of 0.10. All values contain the true value within their range, however, an observable widening of the range occurs from expertment 5 (`Epoch1-CHD`). Of the four experiments with the narrowest interval, two are from epoch 3, and two from epoch 2.
- *Death fraction (DF)*, which is the fraction of hospitalisations that die, with a true value of 0.10. Interestingly, the narrowest ranges (the first four experiments) do not cover the true value at the 2.5% level, although the difference is small (e.g. 0.002 for `Epoch2-CHD`).

4.5 Overlapping analysis of posterior densities

Visual inspection of the parameter densities can be performed for each of the 21 experiments. The aim is to identify fitted parameters that have different posterior distributions, and explore possible reasons why this might be so. In Figure 8 we explore the posterior densities for β and c (the clinical fraction). The original model parameter values are



	EpIndCode	Parameter	Q_0.025	Median	Mean	Q_0.975	Q95Range
1	Epoch3-CH	Beta_Param	0.991	0.995	0.995	1.000	0.009
2	Epoch3-CHD	Beta_Param	0.990	0.995	0.995	0.999	0.009
3	Epoch3-CD	Beta_Param	0.992	0.998	0.998	1.003	0.011
4	Epoch3-C	Beta_Param	0.993	0.999	0.999	1.005	0.012
5	Epoch3-HD	Beta_Param	0.984	0.990	0.990	0.996	0.012
6	Epoch3-H	Beta_Param	0.983	0.990	0.990	0.997	0.014
7	Epoch2-CH	Beta_Param	0.981	0.989	0.989	0.997	0.016
8	Epoch2-CHD	Beta_Param	0.980	0.988	0.988	0.996	0.016
9	Epoch2-H	Beta_Param	0.975	0.986	0.986	0.997	0.022
10	Epoch2-CD	Beta_Param	0.978	0.990	0.990	1.001	0.023
11	Epoch2-HD	Beta_Param	0.973	0.985	0.985	0.996	0.023
12	Epoch2-C	Beta_Param	0.980	0.992	0.992	1.004	0.024
13	Epoch3-D	Beta_Param	0.973	0.987	0.987	1.003	0.030
14	Epoch1-CH	Beta_Param	0.928	0.973	0.976	1.039	0.111
15	Epoch1-CHD	Beta_Param	0.930	0.974	0.977	1.041	0.111
16	Epoch1-C	Beta_Param	0.930	0.976	0.979	1.046	0.116
17	Epoch1-CD	Beta_Param	0.927	0.974	0.977	1.046	0.119
18	Epoch2-D	Beta_Param	0.783	0.958	0.941	0.999	0.216
19	Epoch1-HD	Beta_Param	0.827	0.940	0.944	1.078	0.251
20	Epoch1-H	Beta_Param	0.815	0.929	0.933	1.073	0.258
21	Epoch1-D	Beta_Param	0.196	0.571	0.570	0.948	0.752

	EpIndCode	Parameter	Q_0.025	Median	Mean	Q_0.975	Q95Range
1	Epoch3-CH	CF	0.552	0.598	0.598	0.645	0.093
2	Epoch3-CD	CF	0.559	0.603	0.604	0.653	0.094
3	Epoch3-CHD	CF	0.550	0.596	0.597	0.645	0.095
4	Epoch3-C	CF	0.560	0.606	0.607	0.657	0.097
5	Epoch2-CHD	CF	0.581	0.647	0.648	0.721	0.140
6	Epoch2-CH	CF	0.578	0.644	0.646	0.723	0.145
7	Epoch2-C	CF	0.570	0.639	0.640	0.719	0.149
8	Epoch2-CD	CF	0.574	0.644	0.645	0.727	0.153
9	Epoch1-CHD	CF	0.360	0.634	0.634	0.908	0.548
10	Epoch1-CH	CF	0.358	0.641	0.641	0.921	0.563
11	Epoch1-C	CF	0.346	0.637	0.639	0.915	0.569
12	Epoch1-CD	CF	0.342	0.635	0.635	0.920	0.578
13	Epoch3-HD	CF	0.077	0.239	0.294	0.773	0.696
14	Epoch2-HD	CF	0.078	0.252	0.301	0.785	0.707
15	Epoch3-H	CF	0.076	0.249	0.304	0.787	0.711
16	Epoch2-H	CF	0.077	0.254	0.307	0.790	0.713
17	Epoch1-HD	CF	0.072	0.324	0.359	0.819	0.747
18	Epoch3-D	CF	0.026	0.216	0.271	0.775	0.749
19	Epoch1-H	CF	0.076	0.344	0.380	0.842	0.766
20	Epoch2-D	CF	0.040	0.273	0.320	0.829	0.789
21	Epoch1-D	CF	0.091	0.482	0.487	0.898	0.807

	EpIndCode	Parameter	Q_0.025	Median	Mean	Q_0.975	Q95Range
1	Epoch3-CH	HF	0.089	0.100	0.100	0.111	0.022
2	Epoch3-CHD	HF	0.089	0.099	0.100	0.112	0.023
3	Epoch2-CH	HF	0.081	0.093	0.094	0.107	0.026
4	Epoch2-CHD	HF	0.081	0.093	0.093	0.107	0.026
5	Epoch1-CHD	HF	0.073	0.108	0.110	0.157	0.084
6	Epoch1-CH	HF	0.076	0.113	0.114	0.163	0.087
7	Epoch2-CD	HF	0.015	0.103	0.178	0.699	0.684
8	Epoch3-HD	HF	0.076	0.248	0.300	0.760	0.684
9	Epoch2-HD	HF	0.079	0.244	0.298	0.769	0.690
10	Epoch3-CD	HF	0.016	0.101	0.176	0.713	0.697
11	Epoch2-H	HF	0.078	0.239	0.295	0.784	0.706
12	Epoch3-H	HF	0.076	0.239	0.294	0.785	0.709
13	Epoch1-CD	HF	0.018	0.155	0.218	0.728	0.710
14	Epoch1-HD	HF	0.076	0.319	0.359	0.811	0.735
15	Epoch1-H	HF	0.080	0.354	0.384	0.836	0.756
16	Epoch3-D	HF	0.025	0.221	0.277	0.782	0.757
17	Epoch2-D	HF	0.033	0.254	0.308	0.826	0.793
18	Epoch3-C	HF	0.102	0.498	0.499	0.895	0.793
19	Epoch2-C	HF	0.102	0.501	0.501	0.898	0.796
20	Epoch1-D	HF	0.099	0.492	0.495	0.900	0.801
21	Epoch1-C	HF	0.084	0.505	0.501	0.907	0.823

	EpIndCode	Parameter	Q_0.025	Median	Mean	Q_0.975	Q95Range
1	Epoch3-CHD	DF	0.104	0.121	0.121	0.140	0.036
2	Epoch3-HD	DF	0.104	0.122	0.122	0.141	0.037
3	Epoch2-CHD	DF	0.102	0.122	0.123	0.146	0.044
4	Epoch2-HD	DF	0.103	0.124	0.124	0.149	0.046
5	Epoch1-HD	DF	0.043	0.197	0.222	0.541	0.498
6	Epoch1-CHD	DF	0.045	0.204	0.227	0.562	0.517
7	Epoch2-CD	DF	0.016	0.112	0.178	0.676	0.660
8	Epoch1-CD	DF	0.018	0.139	0.204	0.701	0.683
9	Epoch3-CD	DF	0.017	0.117	0.188	0.718	0.701
10	Epoch2-D	DF	0.036	0.251	0.298	0.786	0.750
11	Epoch3-D	DF	0.027	0.217	0.273	0.779	0.752
12	Epoch2-CH	DF	0.104	0.505	0.502	0.897	0.793
13	Epoch2-C	DF	0.104	0.502	0.503	0.905	0.801
14	Epoch3-C	DF	0.097	0.495	0.498	0.900	0.803
15	Epoch1-D	DF	0.093	0.478	0.483	0.897	0.804
16	Epoch3-CH	DF	0.096	0.495	0.499	0.901	0.805
17	Epoch1-H	DF	0.096	0.497	0.500	0.906	0.810
18	Epoch1-C	DF	0.094	0.492	0.495	0.907	0.813
19	Epoch3-H	DF	0.095	0.513	0.506	0.912	0.817
20	Epoch1-CH	DF	0.090	0.498	0.494	0.908	0.818
21	Epoch2-H	DF	0.088	0.494	0.495	0.908	0.820

Fig 7. Summary of quantiles for parameters across the 21 experiments.

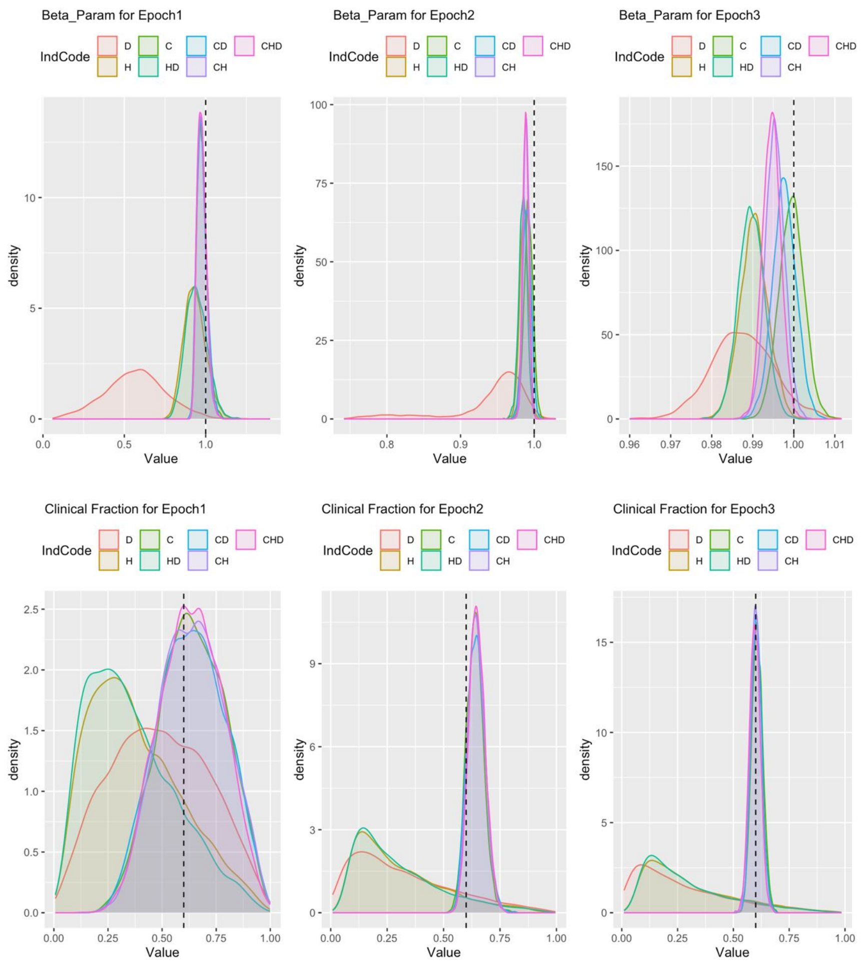


Fig 8. Posterior density distributions for beta and the clinical fraction

indicated by the dotted vertical lines, and the density plots are divided into three epochs. The purpose of these plots is to show the overall shape, however it should be noted that the scales on the x axis are different. For example, that parameter range for *Beta_Param* in epoch 3 is quite small, when compared to epochs 1 and 2. There seems to be more variation for the clinical fraction parameter with an epoch, and across the three epochs.

While the visual inspection method is a useful way to explore differences, more formal approaches can provide measurements to provide insights into the similarity between distributions. One such method is *overlapping*, which defined as the area intersected by two or more probability density functions [19]. The R package `overlapping` [18] is used to calculate this measure for each fitted parameter, with comparisons made between each of the 21 experiments. Output from an overlapping analysis of the parameter β is shown in Figure 10.

While we already mentioned that the range for epoch 3 is quite narrow for β estimates, the data provided through this analysis is interesting. For example, the highest overlapping value (98%) is between `Epoch1-CHD` and `Epoch1-CH`, where the only difference between these data sets is the used of the deaths indicator. In general, overlapping values of 90% or higher are associated with data sets from similar epochs, which is a useful test of the fitting process.The lowest overlapping values are associated with `Epoch1-D`, which would have the lowest amount of data (one indicator with 33 data

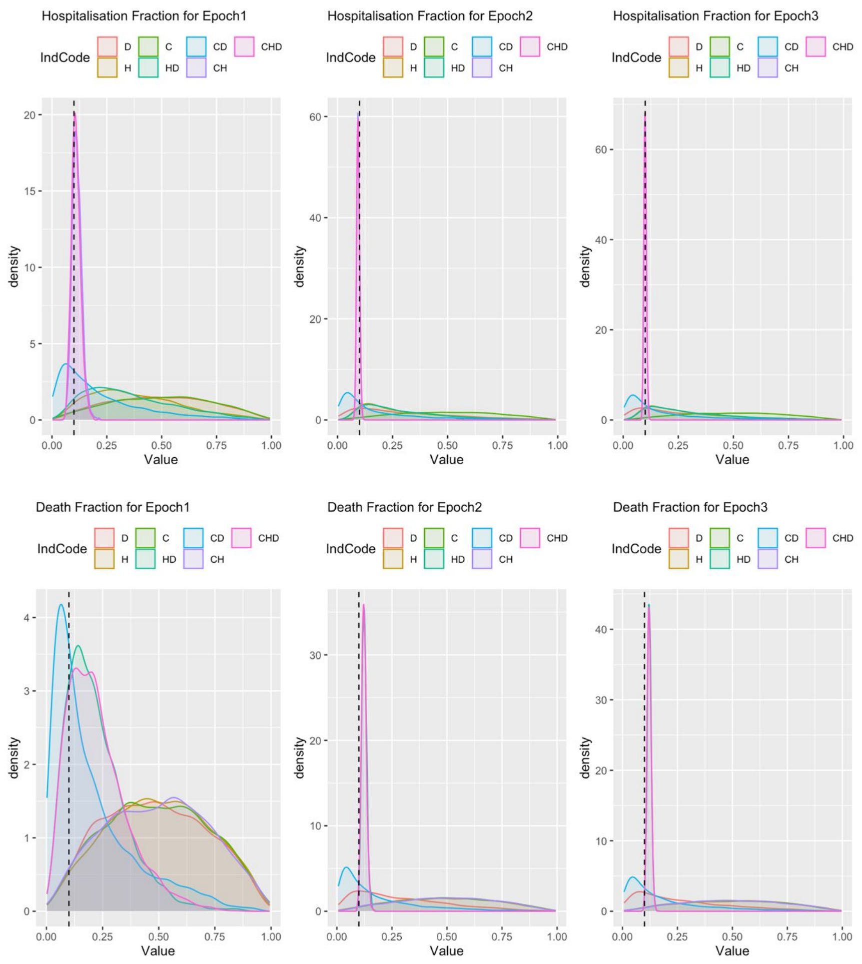


Fig 9. Posterior density distributions for hospitalisation and death fractions

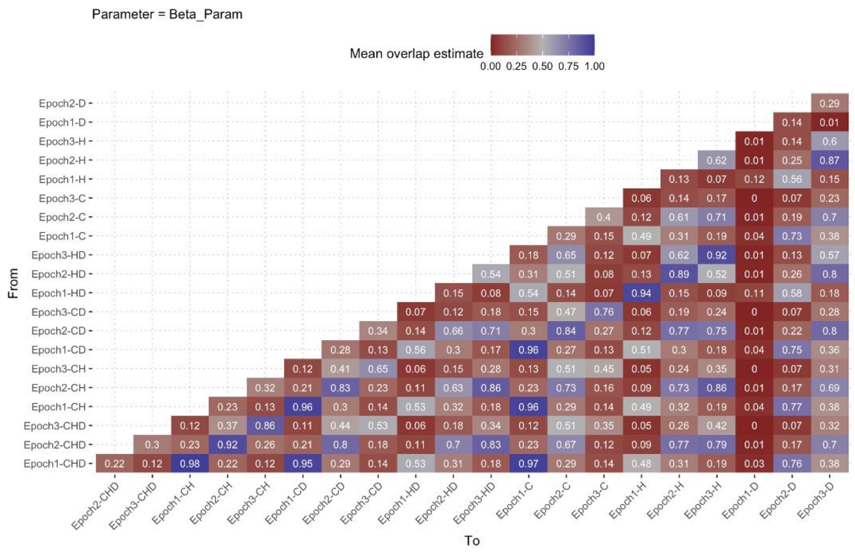


Fig 10. Overlapping analysis for beta

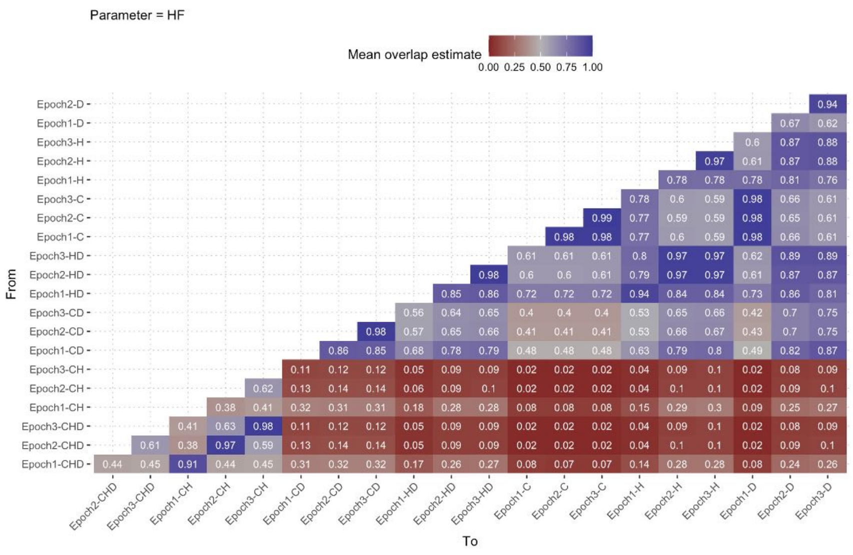


Fig 11. Overlapping analysis for hospitalisation fraction

points).

A second overlapping analysis for the hospitalisation fraction parameter is shown in Figure 11, and these show a range of values from a high overlapping fraction of 99% (`Epoch2-C` with `Epoch3-C`), and low values of 2%, for example, between `Epoch3-CH` and `Epoch1-C`. If we take `Epoch3-CHD` as a plausible experiment where you would expect good estimations (given that it maximises the amount of data for calibration), it is interesting to see that only one other experiment has a high level of overlap, namely, `Epoch3-CH`.

Overall, the value of this overlapping algorithm is that it provides a similarity measurement between the parameter posterior distributions. The measure indicates the impact of data availability, and choice of indicators, on the fitting process. Potentially, this approach could have value during the validation stages of parameter fitting, as a means to explore teh relationships between data availability and parameter estimates.

On reflection, it would also have been useful to highlight the prior distribution on the density graphs, just to confirm that the fitting process has been impacted by the available data. For example, some of the posterior densities for the death fraction (Epoch1), do look quite similar to the prior $Beta(2, 2)$ distribution that was specified as part of the stan inference model.

Further research could combine additional fitting metrics to bear on this process. For example, the potential scale-reduction factor ($\widehat{R}$) [5] which compares within-chain variance (stationarity) to between-chain variance (mixing). This could provide further insights into how the overlap percentage could be used to indicate an acceptable level of similarity between posterior distributions. Also, the use of additional unsupervised machine learning methods such as clustering could be benefiical, as a way to explore further patterns in the data.

5 Conclusion

In reflecting on the results, a valuable output (in additional to the visualisation and exploratory analysis), is the calculation of the overlapping metric, whic presents a similarity measure between the parameter estimates. The differences calculated indicate that the MCMC fitting process is sensitive to (1) the amount of data in a time series (e.g. whether it is epoch one, two, or three), and (2) the number of indicators available for the process (combinations of cases, hospitalisations, and deaths). On the broader point, it also suggests that modellers must reflect on parameter estimates, and how the ranges can vary depending on the amount of data available. Overall, despite impressive technical advances in computational inference processes, parameter estimation remains a challenging task in system dynamics; the following quote seems as relevant today as it was in the early 2000s.

> ... limitations in numerical data availability mean it is often impossible to estimate all parameters in a model. You must also develop the ability to estimate parameters judgmentally using expert opinion gleaned from interviews, workshops, archival materials, direct experience, and other methods.
>
> — John D. Sterman [20]

6 Appendix 1 - Stan model for CHD Indicators

The following stan code was automatically generated by the package `readsdr` [17] based on the Stella XMILE file. It embeds the system dynamics model (functions block), and also adds code to calculate the incidence data (`delta_x_1`, `delta_x_2`, and `delta_x_3`), namely the earlier equations (17-19). The `model` block below comprises the parameters to be fitted, and also specifies the distributions for cases (C), hospitalisations (H), and deaths (D). A separate stan file (seven in total) was created for each unique combination of indicator (i.e., CHD, CH, CD, HD, C, H, and D).

```
// Code generated by the R package readsdr v0.2.0.9014
// See more info at github https://github.com/jandraor/readsdr

// SEIR Model in stan, direct translation from XMILE via readsdr
functions {
  vector X_model(real time, vector y, array[] real params) {
    vector[12] dydt;
    real ER;
    real RR;
    real HR1;
    real HR2;
    real HR3;
    real RRH;
    real RRD;
    real Lambda;
    real TCI;
    real THI;
    real TDI;
    real TIH;
    real Checksum;
    real IR;
    ER = y[2]*0.5;
    RR = (1-params[1]*params[2])*y[3]*0.5;
    HR1 = params[1]*params[2]*y[3]*0.5;
    HR2 = y[5]/(10/3.0);
    HR3 = y[6]/(10/3.0);
    RRH = (1-params[3])*y[7]/(10/3);
    RRD = params[3]*y[7]/(10/3.0);
    Lambda = params[4]*y[3]/1e+05;
    TCI = params[1]*ER;
    THI = HR1;
    TDI = RRD;
    TIH = y[5]+y[6]+y[7];
    Checksum = y[1]+y[2]+y[3]+TIH+y[4]+y[8]+y[9];
    IR = y[1]*Lambda;
    dydt[1] = -IR;
    dydt[2] = IR-ER;
    dydt[3] = ER-RR-HR1;
    dydt[4] = RR;
    dydt[5] = HR1-HR2;
    dydt[6] = HR2-HR3;
    dydt[7] = HR3-RRH-RRD;
    dydt[8] = RRH;
    dydt[9] = RRD;
    dydt[10] = TCI;
    dydt[11] = THI;
    dydt[12] = TDI;
    return dydt;
  }
}

// Data for the calibration process
data {
  int<lower = 1> n_obs;
  array[n_obs] int C;
  array[n_obs] int H;
  array[n_obs] int D;
  array[n_obs] real ts;
  vector[12] x0;
}


// Parameters to be fitted.
parameters {
  real<lower = 0, upper = 1> CF;
  real<lower = 0, upper = 1> HF;
  real<lower = 0, upper = 1> DF;
  real<lower = 0> Beta_Param;
  real<lower = 0> inv_phi1;
  real<lower = 0> inv_phi2;
  real<lower = 0> inv_phi3;
}

// Calling the SEIR model and extracting cases
transformed parameters{
  array[n_obs] vector[12] x; // Output from the ODE solver
  array[4] real params;
  array[n_obs] real delta_x_1;
  array[n_obs] real delta_x_2;
  array[n_obs] real delta_x_3;
  real phi1;
  real phi2;
  real phi3;
  phi1 = 1 / inv_phi1;
  phi2 = 1 / inv_phi2;
  phi3 = 1 / inv_phi3;
  params[1] = CF;
  params[2] = HF;
  params[3] = DF;
  params[4] = Beta_Param;
  x = ode_rk45(X_model, x0, 0, ts, params);
  delta_x_1[1] =  x[1, 10] - x0[10] + 1e-5;
  delta_x_2[1] =  x[1, 11] - x0[11] + 1e-5;
  delta_x_3[1] =  x[1, 12] - x0[12] + 1e-5;
  for (i in 1:n_obs-1) {
    delta_x_1[i + 1] = x[i + 1, 10] - x[i, 10] + 1e-5;
    delta_x_2[i + 1] = x[i + 1, 11] - x[i, 11] + 1e-5;
    delta_x_3[i + 1] = x[i + 1, 12] - x[i, 12] + 1e-5;
  }
}

// The stan model, including priors.
model {
  CF ~ beta(2, 2);
  HF ~ beta(2, 2);
  DF ~ beta(2, 2);
  Beta_Param ~ lognormal(0, 1);
  inv_phi1 ~ exponential(5);
  inv_phi2 ~ exponential(5);
  inv_phi3 ~ exponential(5);
  C ~ neg_binomial_2(delta_x_1, phi1);
  H ~ neg_binomial_2(delta_x_2, phi2);
  D ~ neg_binomial_2(delta_x_3, phi3);
}

// Values generated by fitting process.
generated quantities {
  real log_lik;
  array[n_obs] int sim_C;
  array[n_obs] int sim_H;
  array[n_obs] int sim_D;

  // The negative binomial probability mass given location and precision.
  log_lik = neg_binomial_2_lpmf(C | delta_x_1, phi1)+
            neg_binomial_2_lpmf(H | delta_x_2, phi2)+
            neg_binomial_2_lpmf(D | delta_x_3, phi3);

  // Generate the three negative binomial variates
  sim_C = neg_binomial_2_rng(delta_x_1, phi1);
  sim_H = neg_binomial_2_rng(delta_x_2, phi2);
  sim_D = neg_binomial_2_rng(delta_x_3, phi3);
}
```

7 Appendix 2 - Overall script to run the inference process

This code is the main script used to run the inference process. Using R's `tidyverse` tools [14], it stores all the runs in an RDS file which can be used for subsequent analysis.

```
library(purrr)
library(glue)
library(lubridate)

source("R/02 estimate/Header.R")
get_stamp <- function(sep=" ")
{
  lubridate::ymd_hms(Sys.time()) %>%
    str_replace("UTC","") %>%
    str_trim() %>%
    str_replace(" ",sep)
}

# Get the data
epi <- get_data(config$G_DATA)

# Configure factorial design
exp <- configure_exp(config$EPOCHS,
                     config$MEAS_MODELS)

# Prepare readsdr stan measure models
exp <- exp %>%
        mutate(FitData=map(Epoch,~filter_data(epi,.x)),
               MM_readsdr=map(Indicators,~get_meas_model(.x)))

config$RUN_INFO <- vector(mode="list",length = nrow(exp))


# Run stan fits for each experiment, store in new column
exp <- exp %>%
        mutate(StanFit=pmap(list(ExpNumber,
                                 IndCode,
                                 Indicators,
                                 FitData,
                                 MM_readsdr),~{
          t1 <- Sys.time()
          start_time <- get_stamp()
          f <- fit_model(XMILE_FILE = config$G_MODEL,
                         STAN_FILE  = str_replace(config$G_STAN_MODEL,
                                                  "SEIRH.stan",
                                                  paste0("SEIRH_",..2,".stan")),
                         Exper      = ..1,
                         Indicators = ..3,
                         data       = ..4,
                         meas_model = ..5)
          diagnostic      <- f$cmdstan_diagnose()
          diagnostic_summ <- f$diagnostic_summary()
          finish_time <- get_stamp()
          config$RUN_INFO[[..1]] <<- list(ExpNo=..1,
                                          Indicators=..2,
                                          Obs=nrow(..3),
                                          Start_Time=start_time,
                                          Finish_Time=finish_time,
                                          Duration=Sys.time()-t1,
                                          Diagnostic=diagnostic,
                                          Diagnostic_Summary=diagnostic_summ)
          f
        }))
```

```
# Save in RDS file
STAMP <- get_stamp("#")

fits <- prepare_data1(exp,config)

rds_file  <- glue("data/estimates/{config$DESC}_{STAMP}_EP{length(config$EPOCHS)


saveRDS(fits,rds_file)
```

8 Appendix 3 - Header.R file 301

Thsi script configures the runs and specifies (1) the source scripts for the SEIR model 302
and data, (2) the epochs and (3) the indicator combinations. 303

```
source("R/02 estimate/GenerateSamples.R")
source("R/02 estimate/Data.R")
source("R/02 estimate/PrepareData.R")

G_DESC <- "TEST"



config <- list(G_DATA        = "data/SEIRH_Beta.xlsx",
               G_MODEL       = "models/SEIRH_Beta.stmx",
               G_STAN_MODEL  = "models/stan/SEIRH.stan",
               EPOCHS        = c("Epoch1","Epoch2","Epoch3"),
               MEAS_MODELS   = list(c("Cases","Hospitalisations","Deaths"),
                                    c("Cases","Hospitalisations"),
                                    c("Cases","Deaths"),
                                    c("Hospitalisations","Deaths"),
                                    c("Cases"),
                                    c("Hospitalisations"),
                                    c("Deaths")),
               DESC          = G_DESC)
```

9 Appendix 4 - Running the overlapping analysis 304

Here, we present one of the analysis scripts, which uses the `overlapping` package in R 305
to generate a similarity measure between the posterior distributions. 306

```
library(overlapping)
library(purrr)
library(dplyr)
library(tidyr)
library(ggpubr)

FILE   <- "data/estimates/TEST_2024-02-28#14:28:32_EP3_MM7_FITS.rds"
fits <- readRDS(FILE)

object_size(fits)
```

```r
all_params <- fits %>%
               select(EpIndCode,Params) %>%
               unnest(cols = "Params")

# Prepare the data for pair-wise analysis for each parameter

prep <- map_df(list("CF","HF","DF","Beta_Param"),~{
  x1 <- all_params %>%
          select(EpIndCode,SN,dplyr::matches(.x))

  cf <- pivot_wider(x1,names_from=EpIndCode,
                      values_from = .x) %>%
        mutate(Param=.x) %>%
        select(Param,everything())
  cf
}) %>% group_by(Param) %>% nest()

overlaps <- prep %>%
              mutate(Overlap=map(data,~{
                d <- select(.x,-SN)
                exps <- names(d)
                cbs <- combn(exps,2)
                inputs <- list(From=cbs[1,],
                               To=cbs[2,])

                over <- map2_df(inputs$From,inputs$To,~{
                  v1 <- d[,.x] %>% pull()
                  v2 <- d[,.y] %>% pull()
                  ol <- overlap(list(v1,v2))$OV
                  tibble(From=.x,To=.y,Overlap=ol)
                })

                over
              }))

oa <- overlaps %>%
       select(Param,Overlap) %>%
       unnest(cols="Overlap") %>%
       mutate(From=factor(From,levels=fits$EpIndCode),
              To=factor(To,levels=fits$EpIndCode))


s_oa <- oa %>%
          group_by(From,To) %>%
          summarise(Median=median(Overlap),
                    Mean=round(mean(Overlap),2),
                    Min=min(Overlap),
                    Max=max(Overlap))

p <- ggplot(s_oa,aes(x=To,y=From,fill=Mean))+geom_tile()+
  scale_fill_gradient2(midpoint = 0.5,mid="grey70",limits=c(0,1))+
  geom_text(aes(To, From, label=Mean), colour = "white", check_overlap = TRUE)+
  theme(legend.position = "top",
        axis.text.x = element_text(angle = 45,hjust = 1),
        panel.background = element_rect(fill="white"),
        panel.grid=element_line(colour="blue",linetype=3,linewidth = 0.3))+
  labs(fill="Mean overlap estimate")


sp_oa <- oa %>%
  group_by(From,To,Param) %>%
  summarise(Median=median(Overlap),
            Mean=round(mean(Overlap),2),
            Min=min(Overlap),
            Max=max(Overlap))

target <- "HF"
p <- ggplot(filter(sp_oa,Param==target),aes(x=To,y=From,fill=Mean))+geom_tile()
  scale_fill_gradient2(midpoint = 0.5,mid="grey70",limits=c(0,1))+
  geom_text(aes(To, From, label=Mean), size=3,colour = "white", check_overlap =
  theme(legend.position = "top",
        axis.text.x = element_text(angle = 45,hjust = 1),
        panel.background = element_rect(fill="white"),
        panel.grid=element_line(colour="grey",linetype=3,linewidth = 0.3))+
  labs(fill="Mean overlap estimate",
       subtitle=paste0("Parameter = ",target))


# Plot histograms to show the overlaps
plots <- prep %>%
          mutate(Plots=map2(Param,data,~{
            set.seed(100)
            d <- select(.y,-SN)
            rn <- sample(1:ncol(d),4)
            d <- d %>% select(rn)
            plots <- final.plot(as.list(d),pairs=T)+labs(subtitle=.x)
          }))

p1 <- ggarrange(plotlist = plots$Plots)
```

References

1. Struben J, Sterman J, Keith D. Parameter estimation through maximum likelihood and bootstrapping methods. Analytical methods for dynamic modelers. 2015; p. 3–38.

2. Dogan G. Bootstrapping for confidence interval estimation and hypothesis testing for parameters of system dynamics models. System Dynamics Review. 2007;23(4):415–436. doi:https://doi.org/10.1002/sdr.362.

3. Pierson K, Sterman JD. Cyclical dynamics of airline industry earnings. System Dynamics Review. 2013;29(3):129–156. doi:https://doi.org/10.1002/sdr.1501.

4. Osgood ND, Liu J. Combining Markov chain Monte Carlo approaches and dynamic modeling. Analytical methods for dynamic modelers. 2015; p. 125–169.

5. Andrade J, Duggan J. A Bayesian approach to calibrate system dynamics models using Hamiltonian Monte Carlo. System Dynamics Review. 2021;37(4):283–309. doi:https://doi.org/10.1002/sdr.1693.

6. Andrade J, Duggan J. An evaluation of Hamiltonian Monte Carlo performance to calibrate age-structured compartmental SEIR models to incidence data. Epidemics. 2020;33:100415. doi:https://doi.org/10.1016/j.epidem.2020.100415.

7. Andrade J, Duggan J. Anchoring the mean generation time in the SEIR to mitigate biases in R0 estimates due to uncertainty in the distribution of the epidemiological delays. Royal Society Open Science. 2023;10(8):230515. doi:10.1098/rsos.230515.

8. Rahmandad H, Lim TY, Sterman J. Behavioral dynamics of COVID-19: estimating underreporting, multiple waves, and adherence fatigue across 92 nations. System Dynamics Review. 2021;37(1):5–31. doi:https://doi.org/10.1002/sdr.1673.

9. Rahmandad H, Xu R, Ghaffarzadegan N. Enhancing long-term forecasting: Learning from COVID-19 models. PLOS Computational Biology. 2022;18(5):1–15. doi:10.1371/journal.pcbi.1010100.

10. Jit M, Ainslie K, Althaus C, Caetano C, Colizza V, Paolotti D, et al. Reflections On Epidemiological Modeling To Inform Policy During The COVID-19 Pandemic In Western Europe, 2020–23. Health Affairs. 2023;42(12):1630–1636. doi:10.1377/hlthaff.2023.00688.

11. Vynnycky E, White R. An introduction to infectious disease modelling. OUP oxford; 2010.

12. Vynnycky E, Edmunds W. Analyses of the 1957 (Asian) influenza pandemic in the United Kingdom and the impact of school closures. Epidemiology & Infection. 2008;136(2):166–179.

13. Duggan J. System dynamics modeling with R. Springer; 2016.

14. Duggan J. Exploring Operations Research with R. Chapman and Hall/CRC.; 2024. Available from: `https://doi.org/10.1201/9781003293910`.

15. Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, et al. Stan: A probabilistic programming language. Journal of statistical software. 2017;76.

16. Gabry J. cmdstanr: R Interface to'CmdStan'. (No Title). 2021;.

17. Andrade J. Readsdr: translate models from system dynamics software into R; 2021. Available from: `https://github.com/jandraor/readsdr`.

18. Pastore M. Overlapping: a R package for Estimating Overlapping in Empirical Distributions. Journal of Open Source Software. 2018;3(32):1023. doi:10.21105/joss.01023.

19. Pastore M, Calcagnì A. Measuring Distribution Similarities Between Samples: A Distribution-Free Overlapping Index. Frontiers in Psychology. 2019;10. doi:10.3389/fpsyg.2019.01089.

20. Sterman J. Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill; 2000.