Reconstruction of the full transmission dynamics of COVID-19 in Wuhan

Hui Li, Yige Li, Jonathan Luu

1 Abstract

COVID-19 was detected in Wuhan in December 2019. Wuhan's high population density, along with increased social activity and travel stemming from pre-Chinese New Year celebrations, catalysed the outbreak. To contain the spread of the disease, Chinese authorities implemented a city-wide quarantine along with a plethora of non-pharmaceutical interventions. These interventions, along with resources and deployment of healthcare personnel from all over the country, drastically lowered the epidemic curve and reduced the attack rate in Wuhan. Hao et al. [1] were interested in the full transmission dynamics of SARS-CoV-2, the causative agent of the epidemic. Using epidemiological data from Wuhan, they delineated the full dynamics of the spread by extending the susceptible-exposed-infectious-recovered (SEIR) model to include presymptomatic infectiousness (P), unascertained cases (A), and case isolation in the hospital (H) – a model they termed SAPHIRE.

After reconstructing the dynamics over 5 periods defined by events and interventions, they identified two key features of the outbreak: high covertness and high transmissibility. They estimated 87% of the infections prior to March 8, 2020 were unascertained with a basic reproduction number (R_0) of 3.54 (95% credible interval 3.40-3.67) in the early outbreak. Compared to other outbreaks including severe acute respiratory syndrome (SARS) and Middle East respiratory syndrome (MERS), SARS-CoV-2's R_0 was much higher. However, the interventions utilized by Chinese authorities had considerable success on controlling the outbreak, reducing the R_0 to 0.28 (95% credible interval 0.23-0.33). Lastly, Hao et. al explored the probability of resurgence if all interventions were lifted after 14 consecutive days of no ascertained infections. Their results highlight the risk of lifting control measures if the number of unascertained cases is high. Furthermore, the results provide insight when considering surveillance and intervention strategies to eventually contain the COVID-19 outbreak.

2 Introduction

2.1 Method being studied

Hao et al.[1] proposed an extended susceptible-exposed-infectious-recovered (SEIR) model to include presymptomatic infectiousness (P), unascertained cases (A), and case isolation in the hospital (H), generating a model they named SAPHIRE. They developed this model to analyze the full transmission dynamics of COVID-19 in Wuhan with particular interest in the ascertainment rate and transmission rate. They modeled the outbreak over 5 different time periods, defined on the basis of key events and interventions in 2020. These include January 1-9 (before Chunyun, a period of high travel around Chinese New Year), January 10-22 (Chunyun), January 23-February 1 (city-wide quarantine), February 2-16 (centralized isolation and quarantine), and February 17-March 8 (community screening). They assumed that the transmission and ascertainment rate did not change in the first two periods, as few interventions were implemented during this time. However, these rates were allowed to vary in later periods to reflect the strength of the interventions. In order to estimate these rates across periods, they utilized Markov Chain Monte Carlo and then converted these transmission rates into the effective reproduction number (R_e).

2.2 Problem being addressed

After the confirmation of human-to-human transmission, Chinese authorities implemented a city-wide quarantine to contain the geographical spread of the disease. They also introduced many non-pharmaceutical interventions, including suspension of all intra- and inter-city transportation, mandatory mask wearing in public places, banning of social gatherings, and home quarantine of individuals with presumed infections. Furthermore, on February 2, they implemented a strict stay-at-home policy for all residents, and the centralized isolation and quarantine of all patients, as well as individuals suspected to have contracted the virus and their close contacts. A city-wide door-to-door universal survey of symptoms was also conducted between February 17-19 to identify previously undetected symptomatic cases. This combination of interventions along with redeployed healthcare workers and resources from all around the country were what reduced the epidemic curve and attack rate in Wuhan.

The authors were interested in reconstructing the transmission dynamics of COVID-19 to analyze the effects these interventions had, accounting for pre-symptomatic infectiousness, time-varying ascertainment rates, transmission rates and population movements. Furthermore, they could utilize

this model to estimate the number of unascertained infections in the population, the reproduction number of the disease, and the probability of resurgence if these interventions are lifted.

2.3 Why this problem is interesting

COVID-19 is still an ongoing pandemic with different countries utilizing different control measures in an effort to lower the epidemic curve. By better understanding the transmission behavior, experts can take this into consideration when developing surveillance methods to monitor the spread of the disease. These results can also help determine which non-pharmaceutical interventions are effective in controlling the spread of the disease, especially as more individuals receive vaccinations in the coming months. Estimating the number of unascertained individuals, potentially including asymptomatic and mildly symptomatic individuals, can also guide resource management and deployment of vaccines to areas with a higher probability of transmission. Furthermore, recent studies have revealed important transmission features of COVID-19 such as the infectiousness of presymptomatic and asymptomatic individuals, so modeling both ascertained and unascertained cases is important for interpreting transmission dynamics and epidemic trajectories.

Another interesting aspect of this paper is the timing of its publication. The results of the paper began circulating around April 2020, early on in the pandemic. It highlighted the importance of unascertained cases, an important factor that public policies and vaccine distributions efforts consider.

2.4 Method background

One of the most common approaches to study the dynamics of an infectious disease is the Susceptible-Exposed-Infectious-Recovered model (SEIR) model. The idea behind this type of model is to assign individuals in a population to different compartments which correspond to different key stages of the disease. The transitions between compartments are then governed by a set of ordinary differential equations (ODE) that involve parameters as targets of estimation.

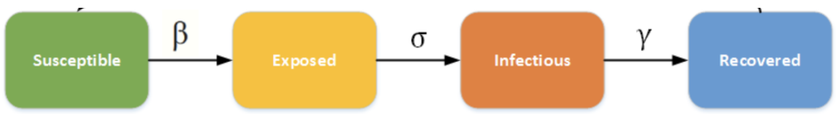

Figure 1: Illustration of the SEIR model. Susceptible (S), Exposed (E), Infected (I), and Recovered (R).

In a classic SEIR model, let S denote the susceptible population, E denote the exposed population, I denote the infectious population, and R denote the recovered population [2]. The infection rate, β, controls the rate of spread by changing the probability of transmitting the disease between a susceptible and infectious individual. The incubation rate, $\sigma = 1/D_e$, is the rate of latent individuals becoming infectious (average duration of incubation is D_e). Lastly, the recovery rate, $\gamma = 1/D_i$, is determined by the average duration of infection.

When considering the SEIR model, a factor that can be introduced is vital dynamics, or births and deaths. Enabling vital dynamics can change how individuals interact within the model. For example, births introduce more susceptible individuals into the population, potentially sustaining an epidemic by allowing new individuals to spread the disease. In a realistic population like this, disease dynamics will reach a steady state. Where μ and ν represent the birth and death rates, respectively, and are assumed to be equal to maintain a constant population, the ODE then becomes:

$$\begin{aligned}
\frac{dS}{dt} &= \mu N - \frac{\beta SI}{N} - \nu S \\
\frac{dE}{dt} &= \frac{\beta SI}{N} - \frac{E}{D_e} - \nu E \\
\frac{dI}{dt} &= \frac{E}{D_e} - \frac{I}{D_i} - \nu I \\
\frac{dR}{dt} &= \frac{I}{D_i} - \nu R
\end{aligned}$$

where $N = S + E + I + R$.

3 Method

This paper proposed a novel extension of the SEIR model to more accurately capture the full-spectrum dynamics of the COVID-19 outbreak in Wuhan, by incorporating three additional compartments: pre-symptomatic infectious individuals (P), unascertained cases (A) and cases isolated in hospitals (H). Figure 14 (panel a) illustrates the proposed SAPHIRE model as well as the schematic disease course of symptomatic individuals. The most important feature of this method is its ability to *jointly* estimate transmission rate *and* ascertainment rate, by explicitly distinguishing between the latent period and incubation period (panel b).

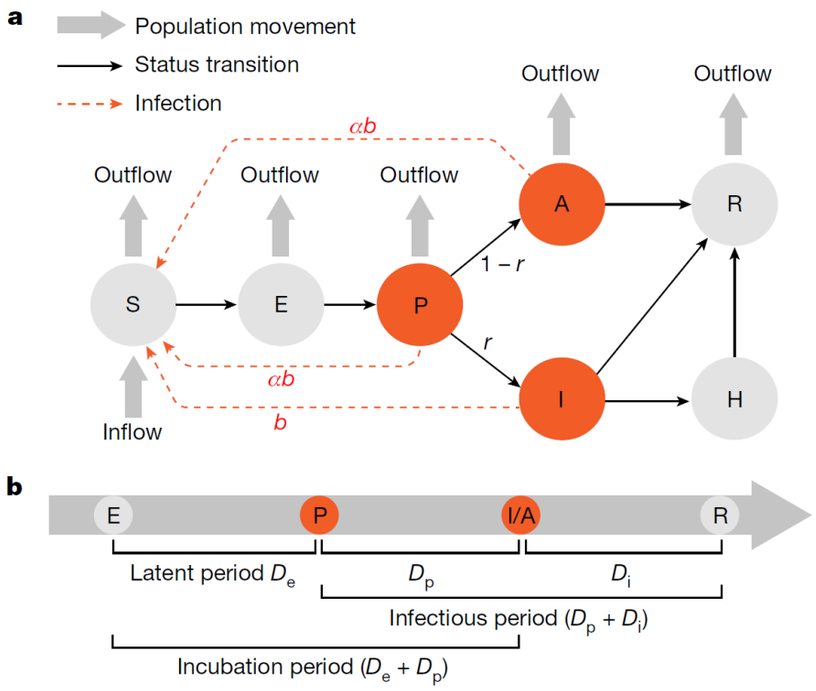

Figure 2: Illustration of the SAPHIRE model. Highlighted in red are the new compartments compared to the SEIR model. Susceptible (S), Exposed (E), Presymptomatic infectious (P), Ascertained infectious (I), Unascertained infectious (A), Isolation in hospital (H) and Removed (R).

In this method, the observed number of ascertained cases in which individuals experienced symptom onset on day d were modeled as random variables that follow a Poisson distribution:

$$x_d \sim Poisson(\lambda_d), \quad \lambda_d = rP_{d-1}D_p^{-1}$$

where P_{d-1} is the expected number of pre-symptomatic individuals on day $(d-1)$. Transmission rates b and ascertainment rates r were jointly modeled in the likelihood function through λ_d's:

$$L(b, r|x_1...x_D) = \prod_{d=1}^{D} \frac{e^{-\lambda_d}\lambda_d^{x_d}}{x_d!}.$$

The ordinary differential equations that characterize the transition rates between the compartments are specified in the following:

$$\frac{dS}{dt} = n - \frac{bS(\alpha P + \alpha A + I)}{N} - \frac{nS}{N}$$

$$\frac{dE}{dt} = \frac{bS(\alpha P + \alpha A + I)}{N} - \frac{E}{D_e} - \frac{nE}{N}$$

$$\frac{dP}{dt} = \frac{E}{D_e} - \frac{P}{D_p} - \frac{nP}{N}$$

$$\frac{dA}{dt} = \frac{(1-r)P}{D_p} - \frac{A}{D_i} - \frac{nA}{N}$$

$$\frac{dI}{dt} = \frac{rP}{D_p} - \frac{I}{D_i} - \frac{I}{D_q}$$

$$\frac{dH}{dt} = \frac{I}{D_q} - \frac{H}{D_h}$$

$$\frac{dR}{dt} = \frac{A+I}{D_i} + \frac{H}{D_h} - \frac{nR}{N}$$

where n is the number of daily inbound and outbound individuals; N is the constant population size; α is the ratio of the transmission rate of *unascertained* cases to that of the ascertained cases; D_X is the average duration of an individual in compartment X. The main parameters of interest are the transmission rate for the ascertained cases (b) and the ascertainment rate (r).

Parameter	Meaning	Jan 1-9	Jan 10-22	Jan 23-Feb 1	Feb 2-16	Feb 17-Mar 8
b	Transmission rate of ascertained cases	b_{12}	b_{12}	b_3	b_4	b_5
r	Ascertainment rate	r_{12}	r_{12}	r_3	r_4	r_5
α	Ratio of transmission rate for unascertained over ascertained cases	0.55	0.55	0.55	0.55	0.55
D_e	Latent period	2.9	2.9	2.9	2.9	2.9
D_p	Presymptomatic infectious period	2.3	2.3	2.3	2.3	2.3
D_i	Symptomatic infectious period	2.9	2.9	2.9	2.9	2.9
D_q	Duration from illness onset to isolation	21	15	10	6	3
D_h	Isolation period	30	30	30	30	30
N	Population size	10,000,000	10,000,000	10,000,000	10,000,000	10,000,000
n	Daily inbound and outbound size	500,000	800,000	0	0	0

Figure 3: Parameter values for the five periods in the main analysis of Wuhan data

The parameters were estimated by MCMC, and in particular, using the delayed rejection adaptive metropolis algorithm implemented in the R package BayesianTools. Convergence was assessed using trace plots and the multivariate Gelman-Rubin diagnostic [3]. Non-informative flat priors of $Unif(0, 2)$ were used for transmission rates; an informative prior of Beta(7.3, 24.6) was used for ascertainment rate.[1] In estimation, r_3, r_4, r_5 were reparametrized as:

$$
\begin{aligned}
logit(r_3) &= logit(r_{12}) + \delta_3 \\
logit(r_4) &= logit(r_3) + \delta_4 \\
logit(r_5) &= logit(r_4) + \delta_5
\end{aligned}
$$

where $\delta_3, \delta_4, \delta_5$ are the parameters to be estimated, with a $N(0, 1)$ prior.

To generate the 95% credible interval of the predicted epidemic curve, the authors used stochastic simulations - for each set of parameter values estimated from MCMC, they performed multinomial random sampling to simulate the flow of individuals across compartments. The stochastic simulations were repeated for 10,000 sets of parameter values sampled by MCMC to construct the 95% credible interval of the epidemic curve by the 2.5 and 97.5 percentiles at each time point.

4 Simulation

For our project, we chose to implement the adaptive-delayed rejection Metropolis Hastings (ARMH) algorithm. The framework for the project (github.com/chaolongwang/SAPHIRE) provides all the code for running the data analysis and simulation work found in the paper. It contains the dataset of the COVID-19 data from Wuhan over the 5 periods, with information on the case date, daily new ascertained cases, and the cumulative sum of these cases. It also contains several R scripts for implementing the SAPHIRE model, fitting the model, and generating descriptive plots.

4.1 Setup

A majority of the ARMH algorithm depends on the BayesianTools library to run. Rather than depending on the full library, we extracted some functions from the library itself and placed them in fun_BTSetup_BST249.R.

These functions were used to create the prior distribution, generate the appropriate likelihood function, create the posterior distribution, generate a common set-up class that encapsulates the values used in the simulation, and a proposal generating function used to move to the next step in the ARMH algorithm. These functions are mainly used in the algorithm to generate the Metropolis-Hastings ratio, generate a proposal step to move to, updating the adaptive correlation matrix for the proposal step generator, and the delayed rejection step.

4.2 Implementing the algorithm

All of the code for implementing the ARMH algorithm is in fun_SEIRfitting_BST249.R within the SEIRfitting function. Two parameters were added to this function: DRlevels to specify the number of delayed rejection chances, and adaptionInterval to specify how often the adaptation step is run.

We follow very similar steps to the original code, initializing all parameters accordingly and generating a prior using the beta, uniform, and normal distribution to generate lower, upper, and best value estimates for each of the eight parameters in our model. We utilize the helper function extracted from BayesianTools to generate an object that contains the likelihood, prior, and posterior functions that we can utilize in the ARMH algorithm.

[1] The parameters of 7.3 and 24.6 were inferred by matching the first two moments of the estimate using exported cases to Singapore.

When implementing the algorithm, it first checks the number of delayed rejection steps. If this value is set to one, it is run like a traditional MH algorithm, generating a proposal step from the proposal generator, calculating the MH ratio between the current and proposed steps, and then generating a number from a Uniform(0,1) to choose whether to move to the proposed step. For our project, we implemented delayed rejection levels up to 4 using the algorithm presented in "Delayed Rejection schemes for efficient Markov chain Monte Carlo sampling of multimodal distributions" [4]. Trias et al. provided an algorithm to generate a new proposal distribution while maintaining the same stationary distribution in the delayed rejection context:

$$\alpha_i(\vec{\lambda}, \vec{\beta_1}, ..., \vec{\beta_i}) = 1 \wedge \left\{ \frac{\pi(\vec{\beta_i}) q_1(\vec{\beta_i}, \vec{\beta}_{i-1}) q_2(\vec{\beta_i}, \vec{\beta}_{i-1}, \vec{\beta}_{i-2}) ... q_i(\vec{\beta_i}, \vec{\beta}_{i-1}, ..., \vec{\lambda})}{\pi(\vec{\lambda}) q_1(\vec{\lambda}, \vec{\beta_1}) q_2(\vec{\lambda}, \vec{\beta_1}, \vec{\beta_2}), ... q_i(\vec{\lambda}, \vec{\beta_1}, ..., \vec{\beta_i})} \right\}$$
$$\frac{[1 - \alpha_1(\vec{\beta_1}, \vec{\beta}_{i-1})][1 - \alpha_2(\vec{\beta_i}, \vec{\beta}_{i-1}, \vec{\beta}_{i-2})] ... [1 - \alpha_{i-1}(\vec{\beta_i}, ..., \vec{\beta_1})]}{[1 - \alpha_1(\vec{\lambda}, \vec{\beta_1})][1 - \alpha_2(\vec{\lambda}, \vec{\beta_1}, \vec{\beta_2})] ... [1 - \alpha_{i-1}(\vec{\lambda}, ..., \vec{\beta}_{i-1})]}$$

where α_i is the probability of accepting the proposed step, q_i is the proposal distribution, $\pi(x)$ is the target distribution, λ is the initial state ($d = 1$), and β_i are the delayed states ($d > 1$). Functions from the fun_BTSetup_BST249.R script were used to generate all of these values. In particular, the proposal distributions were multivariate normal where both the mean and covariance values were updates based on previous steps.

After these jump probabilities α_i are generated and the randomly generated Uniform(0,1) number is used to determine acceptance, the corresponding value and its posterior density are stored. Based on the burn-in and thinning parameters, the chain (history of values) is updated. If we reject the proposed move, we update the chain with the previous value. If we accept the proposed move, we update the chain with the proposed value. Lastly, an adaptive step is done based on two parameters: adaptationDepth and adaptationInterval. These two parameters influence the covariance matrix in the proposal generator function by changing how often it is updated.

4.3 Profiling

After implementing the algorithm and verifying its results with the paper, we attempted to profile the algorithm to locate sections that could speed it up. Using the R package profvis, we analyzed the runtime of the code over 100,000 iterations. Overall, it took 144 seconds to run, with a majority of the time (122 seconds) being spent in calculating the likelihood density. This step incorporates the SAPHIRE model to generate predicted values using the ordinary differential equations discussed previously. The second longest step was calculating the ratio used in the Metropolis-Hastings algorithm, taking around 15 seconds. All other steps took minimal time in comparison to these two steps.

In an attempt to speed up the likelihood calculation, we looked at the SEIRpred.R file provided. A majority of the time was spent in calculating the updated values for each stage of SAPHIRE, and using these values to generate an expected onset value. When examining these values, the numbers in certain steps got very large (up to 200 million), so we tried manipulating the order of operations to keep the calculation numbers small. We also tried the log-exp trick in attempt to minimize runtime while maintaining accuracy. However, these attempts barely affected the runtime of the algorithm.

Parallelization of the code was another option, but a majority of the time is spent within a for loop with dependencies on the previous iterated value. Due to the dependent nature of the algorithm, we could not implement parallelization for this function. One potential idea would be incorporating a C++ script to calculate these values rather than relying on R, but we did not attempt this speed-up method.

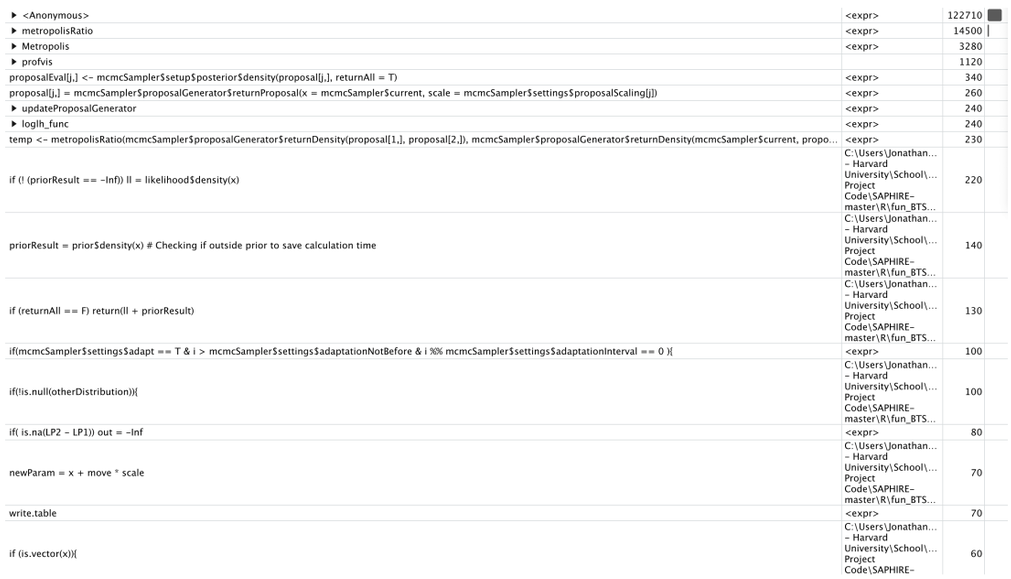

Figure 4: Profiling results - Overall

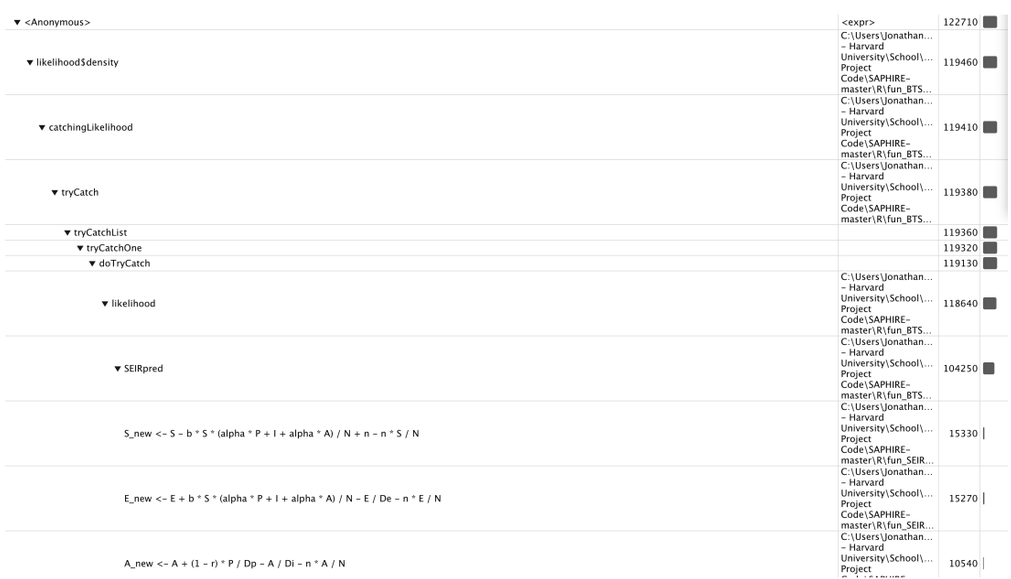

Figure 5: Profiling results - Bottleneck step

4.4 DRAM parameters

After implementing the delayed-rejection and adaptive sections within the Metropolis-Hastings code, we wanted to see how tuning these parameters would affect the results. With the original paper as our baseline, we tried 4 new scenarios: changing the delayed rejection to 3 and 4 (baseline is 2), and changing the adaptationInterval to 1000 and 2000 (baseline is 500). After changing these parameters and using the original dataset, we found very minimal differences in both the plots and estimated parameters.

Table 1: Estimated parameters for each scenario

	Original	DR=3	DR=4	adaptI=1000	adaptI=2000
b12	1.31 (1.25 - 1.38)	1.31 (1.24 - 1.37)	1.31 (1.25 - 1.38)	1.31 (1.25 - 1.38)	1.31 (1.25 - 1.37)
b3	0.40 (0.38 - 0.43)	0.40 (0.38 - 0.43)	0.40 (0.37 - 0.43)	0.40 (0.38 - 0.42)	0.40 (0.38 - 0.43)
b4	0.18 (0.16 - 0.19)	0.17 (0.16 - 0.19)	0.17 (0.16 - 0.19)	0.17 (0.16 - 0.19)	0.17 (0.16 - 0.19)
b5	0.10 (0.08 - 0.12)	0.10 (0.08 - 0.11)	0.10 (0.08 - 0.12)	0.10 (0.08 - 0.11)	0.10 (0.08 - 0.11)
r12	0.15 (0.13 - 0.17)	0.15 (0.13 - 0.17)	0.15 (0.12 - 0.17)	0.15 (0.13 - 0.17)	0.15 (0.13 - 0.17)
delta3	-0.06 (-0.15 - 0.04)	-0.05 (-0.15 - 0.05)	-0.06 (-0.16 - 0.05)	-0.05 (-0.15 - 0.05)	-0.06 (-0.15 - 0.04)
delta4	-0.40 (-0.49 - -0.32)	-0.40 (-0.49 - -0.31)	-0.40 (-0.49 - -0.3)	-0.40 (-0.49 - -0.31)	-0.4 (-0.48 - -0.31)
delta5	0.57 (0.44 - 0.72)	0.58 (0.44 - 0.73)	0.58 (0.42 - 0.74)	0.57 (0.44 - 0.73)	0.58 (0.44 - 0.73)

The original parameter setting for 180,000 iterations took 520 seconds to run. Changing the adaptationInterval did not affect the runtime. However, increasing the delayed rejection to 3 increased the runtime to 750 seconds, and increasing the delayed rejection to 4 increased the runtime

to 955 seconds. Intuitively, this made sense as we are allowing more chances for a particular proposal step to be accepted, so we will have to calculate the appropriate α_i for each additional chance.

4.5 Simulating new datasets

For the previous sections, we chose to use the provided dataset of the actual Wuhan cases. To evaluate the validity of the model and performance of the estimators, we simulated new datasets. Specifically, we considered two generative models, one that matches the model used for estimation (i.e. SAPHIRE) and the other one that does not exactly match the estimation model, where we removed the pre-symptomatic cases and the asymptomatic cases (i.e. SEIRH). We stochastically simulated the dataset using multinomial random sampling. Given a set of initial values $(S^0, E^0, P^0, A^0, I^0, H^0, R^0)$ and the vector of true parameter values $\{b_{12}, b_3, b_4, b_5, r_{12}, r_3, r_4, r_5\}$, we can sequentially update the states of each compartments using the following sampling probabilities.

For a generative model that matches SAPHIRE exactly, we have:

$$
\begin{aligned}
(\Delta_{S^t\to E^t}, \Delta_{S^t\to O^t}, \Delta_{S^t\to S^{t+1}}) &\sim MN\left(S^{t-1}, \frac{b(\alpha P + I + \alpha A)}{N}, \frac{n}{N}, 1 - \frac{b(\alpha P + I + \alpha A)}{N} - \frac{n}{N}\right)\\
(\Delta_{E^t\to P^t}, \Delta_{E^t\to O^t}, \Delta_{E^t\to E^{t+1}}) &\sim MN\left(E^{t-1}, \frac{1}{D_e}, \frac{n}{N}, 1 - \frac{1}{D_e} - \frac{n}{N}\right)\\
(\Delta_{P^t\to I^t}, \Delta_{P^t\to A^t}, \Delta_{P^t\to O^t}, \Delta_{P^t\to P^{t+1}}) &\sim MN\left(P^{t-1}, \frac{r}{D_p}, \frac{1-r}{D_p}, \frac{n}{N}, 1 - \frac{r}{D_p} - \frac{1-r}{D_p} - \frac{n}{N}\right)\\
(\Delta_{I^t\to H^t}, \Delta_{I^t\to R^t}, \Delta_{I^t\to I^{t+1}}) &\sim MN\left(I^{t-1}, \frac{1}{D_q}, \frac{1}{D_i}, 1 - \frac{1}{D_q} - \frac{1}{D_i}\right)\\
(\Delta_{A^t\to R^t}, \Delta_{A^t\to R^t}, \Delta_{A^t\to A^{t+1}}) &\sim MN\left(A^{t-1}, \frac{1}{D_i}, \frac{n}{N}, 1 - \frac{1}{D_i} - \frac{n}{N}\right)\\
(\Delta_{H^t\to R^t}, \Delta_{H^t\to H^{t+1}}) &\sim MN\left(H^{t-1}, \frac{1}{D_h}, 1 - \frac{1}{D_h}\right)\\
(\Delta_{R^t\to O^t}, \Delta_{R^t\to R^{t+1}}) &\sim MN\left(R^{t-1}, \frac{n}{N}, 1 - \frac{n}{N}\right).
\end{aligned}
$$

To generate data following the SEIRH model, we have:

$$
\begin{aligned}
(\Delta_{S^t\to E^t}, \Delta_{S^t\to O^t}, \Delta_{S^t\to S^{t+1}}) &\sim MN\left(S^{t-1}, \frac{bI}{N}, \frac{n}{N}, 1 - \frac{bI}{N} - \frac{n}{N}\right)\\
(\Delta_{E^t\to P^t}, \Delta_{E^t\to O^t}, \Delta_{E^t\to E^{t+1}}) &\sim MN\left(E^{t-1}, \frac{1}{D_e}, \frac{n}{N}, 1 - \frac{1}{D_e} - \frac{n}{N}\right)\\
(\Delta_{I^t\to H^t}, \Delta_{I^t\to R^t}, \Delta_{I^t\to I^{t+1}}) &\sim MN\left(I^{t-1}, \frac{1}{D_q}, \frac{1}{D_i}, 1 - \frac{1}{D_q} - \frac{1}{D_i}\right)\\
(\Delta_{H^t\to R^t}, \Delta_{H^t\to H^{t+1}}) &\sim MN\left(H^{t-1}, \frac{1}{D_h}, 1 - \frac{1}{D_h}\right)\\
(\Delta_{R^t\to O^t}, \Delta_{R^t\to R^{t+1}}) &\sim MN\left(R^{t-1}, \frac{n}{N}, 1 - \frac{n}{N}\right).
\end{aligned}
$$

We used the same initial values for the compartments as specified in Extended Data Table 1 of [1]. The series of $\{\Delta_{P^t\to I^t}\}_{t=1}^{T}$ where T is set to be 60 days (as in the original paper) were taken as the observed data. These represent the daily ascertained cases in which individuals experienced symptom onset on day t, and are modeled as random variables following a Poisson distribution. 100 datasets were simulated under each setting.

In our preliminary analysis, we discovered that the MCMC chains fails to converge (Fig. 6,8). We determined that the poor mixing is related to the mis-specified duration parameters (e.g. latent period, infectious period) which are fixed in the estimation. In contrast, we obtained valid parameter estimates for the same scenario parameterization using data generated under the SAPHIRE model (Fig. 7,9).[2] We assessed convergence using both the trace plots and the Gelman-Rubin diagnostics [3]. Table 2 suggests that convergence is poor when the underlying generative model is SEIRH.

[2]We ran more Markov chains after the presentation, and found that the convergence of the MCMC chain also depends on the seed, so there is some stochasticity associated with the performance. Although we found that this issue seems to arise for both the SEIRH and SAPHIRE model, MCMC converges for the majority of time ($> 90\%$) when the generative model is SAPHIRE whereas failures to converge occur much more frequently when the generative model is SEIRH ($> 90\%$). Hou et al. took account of the stochasticity issue by running three Markov chains with different initial values (i.e. two additional chains after estimation) to ensure convergence. Our conclusion about the performance of SEIRH vs SAPHIRE still holds.

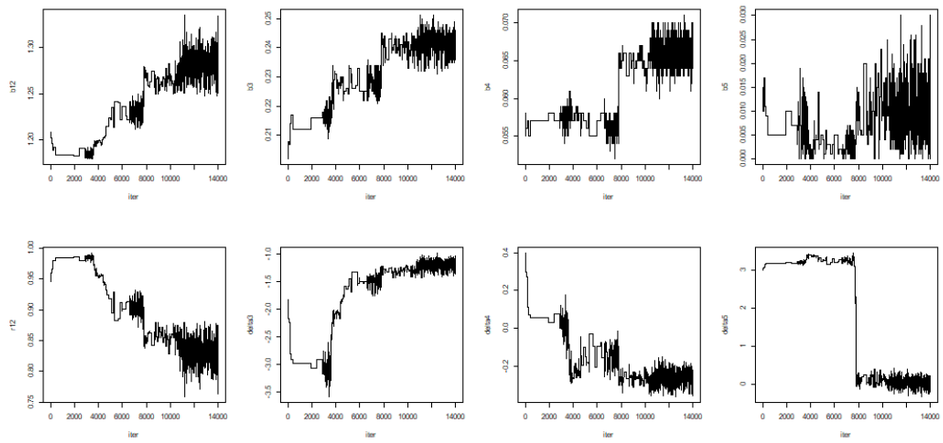

Figure 6: Trace plot of a MCMC chain for each parameter. Generative model: SEIRH.

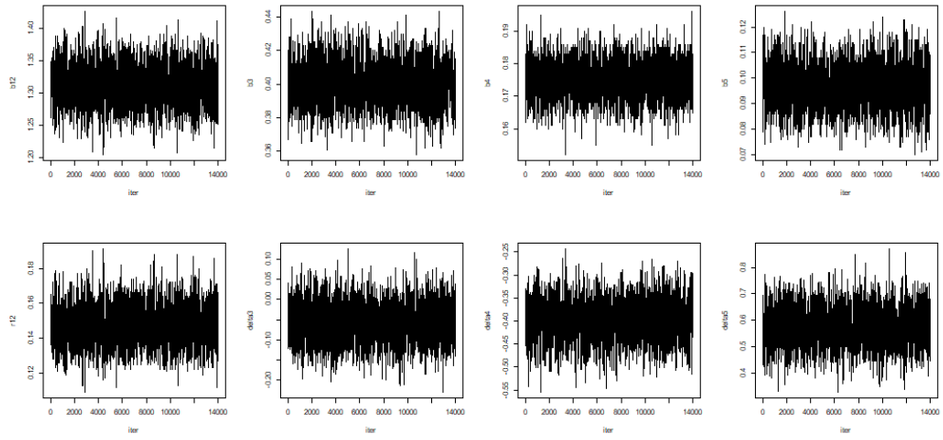

Figure 7: Trace plot of a MCMC chain for each parameter. Generative model: SAPHIRE.

Table 2: Potential scale reduction factor (PSRF) based on Gelman-Rubin diagnostics

	adaptI = 500	adaptI = 1000	adaptI = 2000	DR = 3	DR = 4
SAPHIRE					
b12	1.18	1.23	1.25	1.19	1.16
b3	1.02	1.01	1.01	1.02	1.01
b4	1.03	1.04	1.04	1.03	1.03
b5	1.4	1.35	1.37	1.35	1.32
r12	1.69	1.76	1.79	1.67	1.61
delta3	1.33	1.33	1.3	1.28	1.27
delta4	1.06	1.04	1.05	1.05	1.04
delta5	1.04	1.04	1.03	1.03	1.05
Multivariate	2.79	2.79	2.8	2.61	2.55
SEIRH					
b12	2.4	5.37	1.4	·	·
b3	1.72	7	7.58	·	·
b4	1.5	2.1	1.05	·	·
b5	1.31	1.88	1.48	·	·
r12	1.55	3.27	2.2	·	·
delta3	1.59	1.29	1.73	·	·
delta4	1.31	4.05	3.26	·	·
delta5	1.12	1.6	1.46	·	·
Multivariate	12.8	19	18.9	·	·

Note: When the true generative model is SEIRH, we continued to get errors, even after we changed seeds.

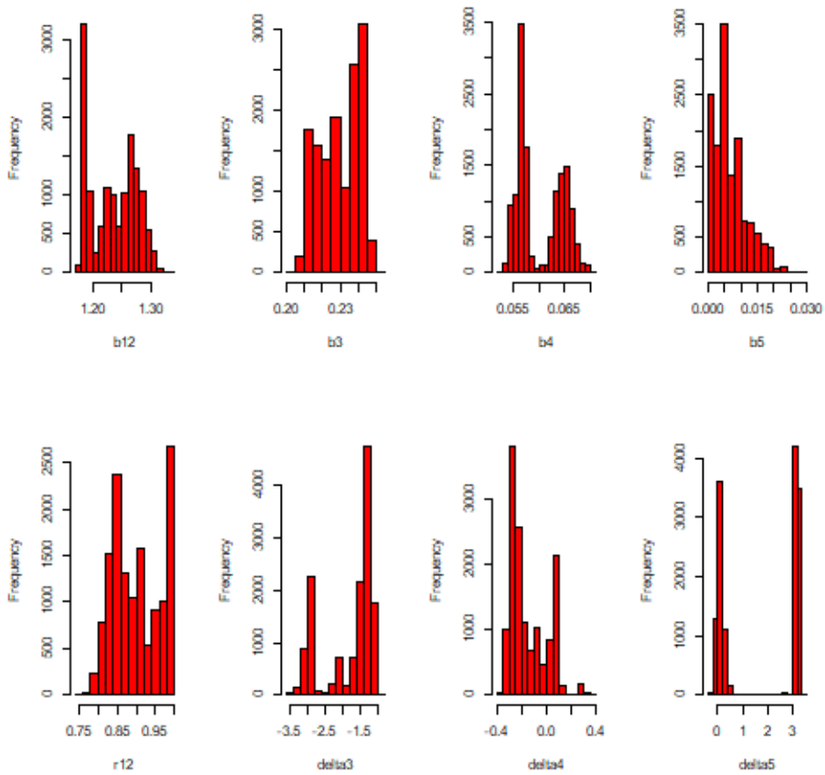

Figure 8: Histogram of MCMC samples for each parameter. Generative model: SEIRH.

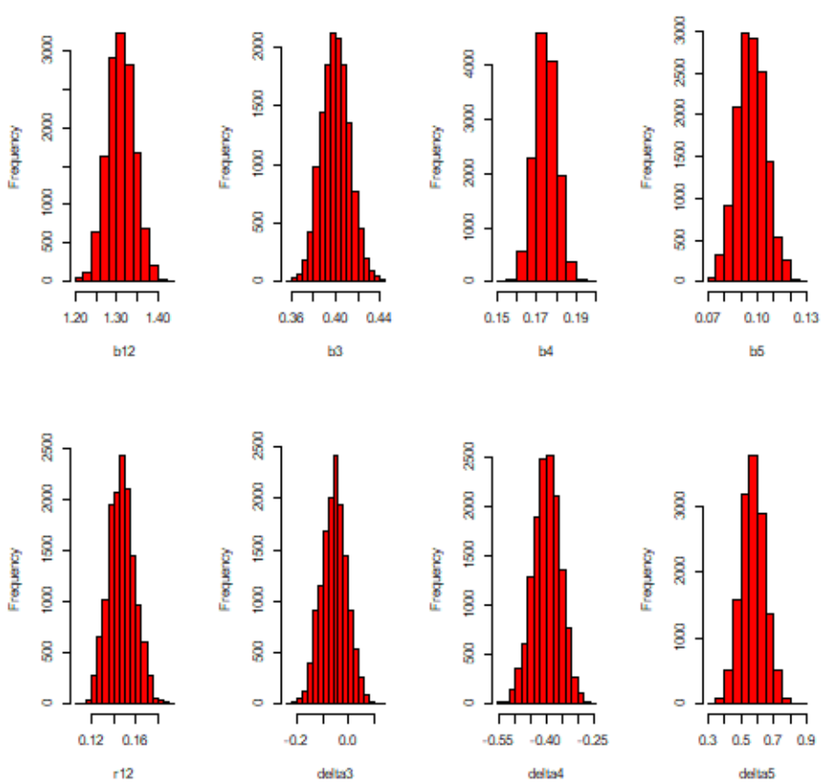

Figure 9: Histogram of MCMC samples for each parameter. Generative model: SAPHIRE.

4.6 Results

Table 3: Absolute bias, RMSE, coverage and length of the 95% confidence interval, run time (mean [Q1, Q3]) for each approach among 100 replicates

	adaptI = 1000	DR = 3	DR = 4	adaptI = 500	adaptI = 2000
Bias					
b12	0.01	0.01	0.01	0.00	0.01
b3	0.00	0.00	0.00	0.00	0.00
b4	0.00	0.00	0.00	0.00	0.00
b5	0.00	0.00	0.00	0.01	0.00
r12	0.00	0.01	0.00	0.00	0.01
delta3	0.01	0.02	0.01	0.00	0.01
delta4	0.00	0.00	0.00	0.01	0.01
delta5	0.01	0.01	0.01	0.02	0.01
RMSE					
b12	0.03	0.04	0.04	0.04	0.04
b3	0.02	0.01	0.01	0.04	0.02
b4	0.01	0.00	0.00	0.01	0.00
b5	0.01	0.01	0.01	0.06	0.01
r12	0.02	0.02	0.02	0.01	0.02
delta3	0.07	0.11	0.11	0.09	0.08
delta4	0.06	0.04	0.06	0.08	0.07
delta5	0.20	0.13	0.12	0.20	0.20
Coverage					
b12	1.00	1.00	1.00	1.00	1.00
b3	1.00	1.00	1.00	0.99	1.00
b4	1.00	1.00	1.00	1.00	1.00
b5	1.00	1.00	1.00	0.99	1.00
r12	1.00	1.00	1.00	1.00	1.00
delta3	1.00	1.00	1.00	0.99	1.00
delta4	1.00	1.00	1.00	0.99	1.00
delta5	1.00	1.00	1.00	0.99	1.00
Length					
b12	0.16	0.15	0.15	0.16	0.15
b3	0.06	0.06	0.06	0.07	0.06
b4	0.03	0.03	0.03	0.03	0.03
b5	0.05	0.04	0.05	0.05	0.05
r12	0.06	0.06	0.06	0.06	0.06
delta3	0.26	0.28	0.27	0.25	0.26
delta4	0.24	0.25	0.24	0.22	0.23
delta5	0.51	0.46	0.36	0.37	0.41
Run time	593 [577, 601]	862 [804, 853]	1103 [1031, 1072]	614 [587, 637]	595 [577, 615]

We chose to focus on evaluating the performance of the method when the true underlying dynamics of transmission matches the SAPHIRE model. As shown in Table 3, all approaches are comparable in terms of bias and coverage, with slightly more differences in RMSE and length. When the levels of delayed rejection increases to 3 or 4 while keeping adaptation interval constant at 1000 (columns 1, 2, 3), prediction error is lower for b_3, δ_5 but higher for b_{12} and δ_3. This suggests that the additionally retained samples were more noisy for certain parameters than for the other.

Comparing across various adaptation intervals while keeping delayed rejection equal to 2 (column 1, 4, 5), we found that when adaptation interval is low at 500, the accuracy of the estimates is lower. Setting adaptation interval to 1000 seems to yield the lowest prediction errors. All five approaches are conservative as the 95% confidence interval covers the true parameter value nearly all of the time. The lengths of the confidence intervals are comparable except for the δ_5 parameter. Generally, the performance of the estimators for $\delta_3, \delta_4, \delta_5$ are more sensitive to the configuration of approach, which may be related to the logit scaling of these parameters (see Section 3). As discussed in Section 4.4, the computational runtime differs across the five estimation procedures. In particular, the levels of delayed rejection has a large impact on runtime – when we included 3 (or 4) layers, runtime increases by approximately 40% (or 80%) compared to the baseline. In comparison, changing adaptation intervals has little effect on the runtime compared to baseline.

We assessed the convergence by both the trace plot and the multivariate Gelman-Rubin diagnostic. Table 2 indicates that varying DR and adaptation interval does not significantly improve convergence (lower multivariate PRSF suggests better convergence). The trace plots of randomly selected Markov chains from each of the estimation configuration are included in the Appendix.

5 Discussion

Hao et al.[1] revealed two critical features of the SARS-CoV-2 virus – its high covertness and high transmissibility. Importantly, these epidemiological findings were enabled by methodological advances in modeling an infectious disease. By explicitly accounting for the non-perfect ascertainment rate of the infectious individuals, the newly proposed SAPHIRE model more accurately delineated the dynamics of COVID-19 outbreak in Wuhan. Moreover, the additional compartments enabled analyses of targeted intervention measures and the outputs can be used to evaluate the effectiveness of containment measures implemented at various points in time. However, one important limitation of the model is that it assumes homogeneous transmission within the population. Although allowing for heterogeneity between demographically different groups will increase the model complexity, it may more realistically reflect the transmission dynamics of the disease.

In terms of estimation, the DRAM algorithm used in this paper has been demonstrated to be robust in extreme situations where other MCMC algorithms fail. For our project, we focused on extending the estimation algorithm implemented by Hao et al. to allow for more than two levels of delayed rejection and varying adaption intervals for model fitting. We found minimal differences in the trace plots and the estimated parameters between runs that utilize different DRAM parameters. This suggests that the chosen number of delayed rejection levels and adaptation intervals in the original method are ideal. In simulations, we considered different generative models for the observed case rates. The performance of the Markov chains is stochastic and depends on seed, but convergence was reached for majority of the time when the generative model is SAPHIRE whereas failures to converge occur much more frequent when we removed the pre-symptomatic and asymptomatic compartments. One potential explanation could be the estimation results are sensitive to the values of the fixed parameters, which requires prior knowledge. In the absence of information such as duration of latent period and infectious period, the method does not always produce valid results when there is a severe mismatch between the generative model and the inference model. Furthermore, when looking at the PSRF statistics for both SAPHIRE and SEIRH simulated data, we obtained values that suggest issues with convergence. For the SAPHIRE model, we obtained multivariate PSRF estimates ranging from 2.55 to 2.80. For the SEIRH model, we obtained much large estimates (> 12.8). Compared to the typical convergence cutoff of 1.1, the model seems to struggle with both simulated datasets. In the original paper using the Wuhan dataset, their model had a PSRF value of 1 using three randomly selected Markov chains, indicating good convergence.

Overall, in the case of the Wuhan dataset, their method provides valid results that accurately predicted both transmission and ascertainment rates. Their algorithm was well optimized, but results were highly dependent on their fixed parameters. This makes their algorithm less generalizable and more difficult to adapt as tuning parameters must be chosen on a case-by-case basis.

References

[1] Hao, X. *et al.* Reconstruction of the full transmission dynamics of covid-19 in wuhan. *Nature* **584**, 420–424 (2020).

[2] Seir and seirs models. `https://docs.idmod.org/projects/emod-hiv/en/latest/model-seir.html`. Accessed: 2021-03-27.

[3] Brooks, S. P. & Gelman, A. General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics* **7**, 434–455 (1998).

[4] Trias, M., Vecchio, A. & Veitch, J. Delayed rejection schemes for efficient markov-chain monte-carlo sampling of multimodal distributions (2009). `0904.2207`.

6 Appendix

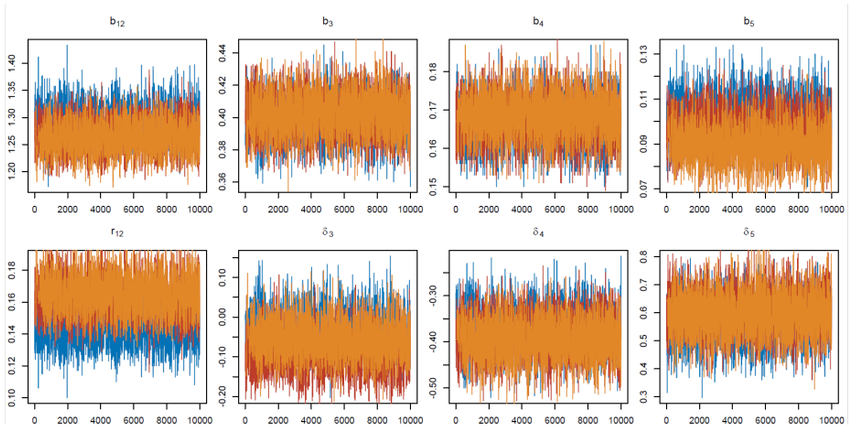

Figure 10: Trace plot of three randomly selected Markov chains for each parameter. Generative model: SAPHIRE. Adaption interval = 1000. DR = 2.

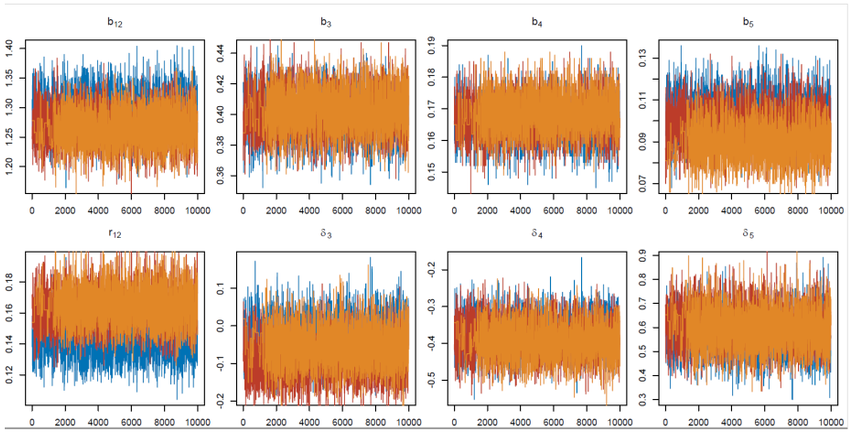

Figure 11: Trace plot of three randomly selected Markov chains for each parameter. Generative model: SAPHIRE. Adaption interval = 1000. DR = 3.

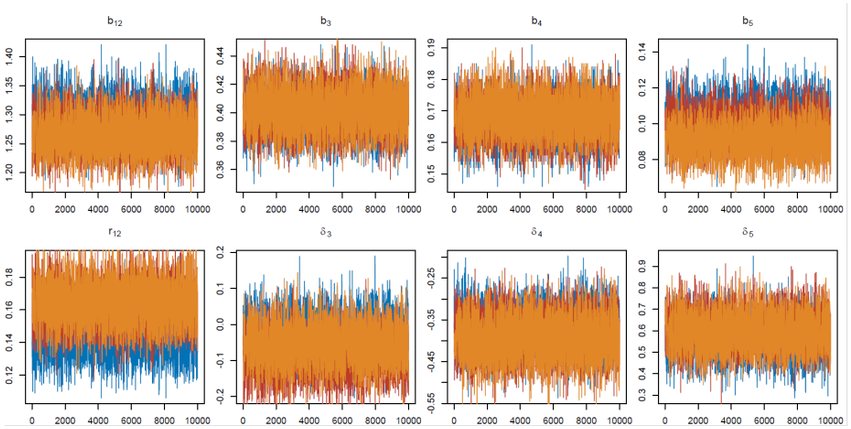

Figure 12: Trace plot of three randomly selected Markov chains for each parameter. Generative model: SAPHIRE. Adaption interval = 1000. DR = 4.

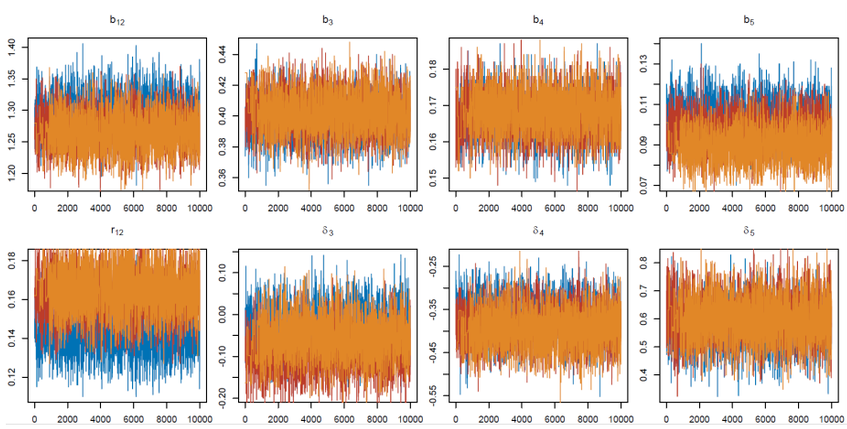

Figure 13: Trace plot of three randomly selected Markov chains for each parameter. Generative model: SAPHIRE. Adaption interval = 500. DR = 2.

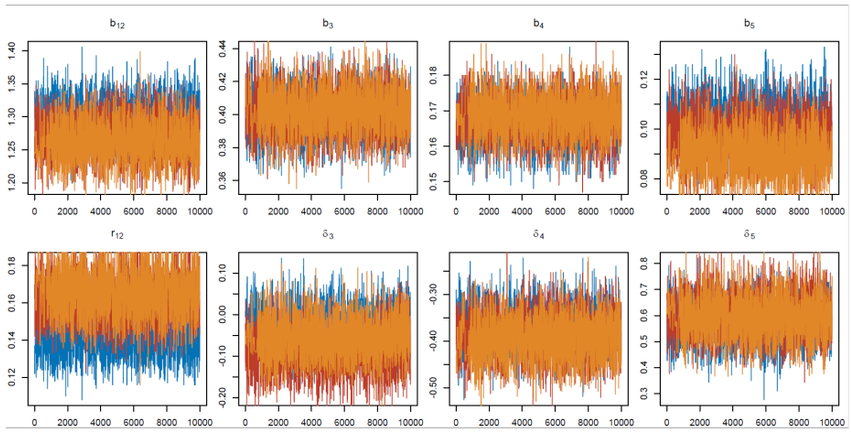

Figure 14: Trace plot of three randomly selected Markov chains for each parameter. Generative model: SAPHIRE. Adaption interval = 2000. DR = 2.