SIR Model

Renee Brady[1], Andrew Marquis[1], Johnny T. Ottesen[2], and Mette S. Olufsen[1]

[1] *Department of Mathematics, North Carolina State University, Raleigh, NC*
[2] *Department of Science and Environment, Roskilde University, Roskilde, Denmark*

SIR Model

Under certain circumstances it may be assumed that the spread of disease by contact with an infected person can be described by the differential equations

$$\begin{aligned} \frac{dS}{dt} &= \delta(N-S) - \gamma kSI \qquad (1)\\ \frac{dI}{dt} &= \gamma kSI - (r+\delta)I, \qquad (2) \end{aligned}$$

where S is the susceptible population, I is the infectious population, and $N = 1000$ is the total population. The parameters of the model are $\theta = (\delta, \gamma, k, r)$, where δ is the net growth rate of susceptible population, γk is the rate of infection, and r is the rate at which subjects recover from the infection. The number of recovered individuals can be calculated directly from the assumption that the total population $N = S + I + R$ is constant. From this equation we get an expression for the recovered, given by

$$R = N - S - I. \qquad (3)$$

For this study, the "true" parameter values $\theta = (0.15, 0.2, 0.1, 0.6)$.

Data

We generated a set of pseudo-data for the population of infected by simulating the forward model using the true parameter values θ. Subsequently, we added 10% noise by drawing random values from the Gaussian distribution with mean 0 and variance $\hat{\sigma}^2$. The model and data are shown in Figure 1.

Model

This type of problem can be written as a general system of nonlinear differential equations of the form

$$\frac{dx}{dt} = f(t, x, \theta), \qquad x(t_0) = x_0, \qquad (4)$$

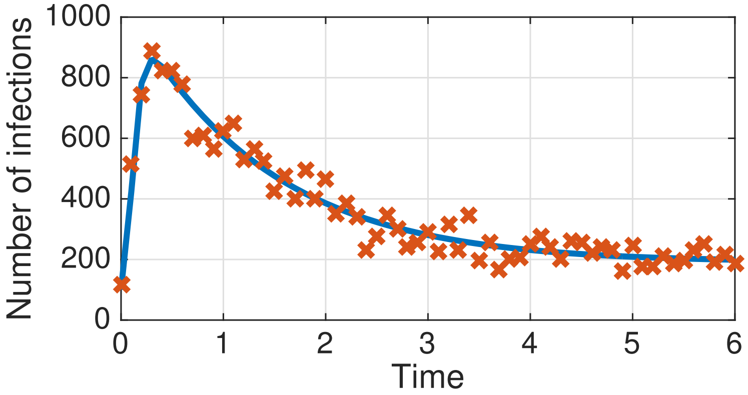

Figure 1: SIR Model (blue line) with simulated data (red dots).

where $x \in \mathbb{R}^n$ denotes the state vector, $t \in \mathbb{R}^T$ denotes time, $\theta \in \mathbb{R}^q$ denotes the parameter vector, and $x_0 \in \mathbb{R}^n$ is the associated initial conditions. For the SIR model, $x \in \mathbb{R}^2$ and $\theta \in \mathbb{R}^4$.

The model output $y \in \mathbb{R}^m$ can be expressed as

$$y = g(t, x, \theta). \tag{5}$$

For the SIR model, we choose $y = I \in \mathbb{R}$.

Given d observations of the model output at time $t = t_0, t_1, \ldots, t_d$, the data can be expressed as

$$Y_i = g(t_i, x(t_i), \theta) + \epsilon_i, \qquad i = 1, 2, \ldots, d, \tag{6}$$

where ϵ_i denotes the measurement noise, which we assume is i.i.d. $\mathcal{N}(0, \sigma^2)$ with some unknown variance σ^2.

Parameter optimization can be used to find a parameter set that gives a best fit to the available data, by minimizing the least squares error between the model predictions and data. This requires an initial guess for the parameter values θ, as well as the identification of a subset of sensitive and identifiable parameters to be estimated. A parameter is said to be *sensitive* if the model output is greatly affected in response to small perturbations of the model parameter. Otherwise, it is considered *insensitive.* A parameter is said to be *identifiable* if it is linearly independent of the other parameters [9].

Sensitivity Analysis

The sensitivity of the model output to the model parameters can be obtained from the sensitivity matrix defined by

$$\chi = \frac{\partial y}{\partial \theta} = \begin{bmatrix} \frac{\partial y_1(t_1)}{\partial \theta_1} & \cdots & \frac{\partial y_1(t_1)}{\partial \theta_q} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_1(t_d)}{\partial \theta_1} & \cdots & \frac{\partial y_1(t_d)}{\partial \theta_q} \\ \frac{\partial y_2(t_1)}{\partial \theta_1} & \cdots & \frac{\partial y_2(t_1)}{\partial \theta_q} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m(t_d)}{\partial \theta_1} & \cdots & \frac{\partial y_m(t_d)}{\partial \theta_q} \end{bmatrix}. \tag{7}$$

Sensitivities can be computed from solutions to sensitivity equations, which can be derived analytically. Alternatively, sensitivities can be computed numerically, e.g. using forward difference or central difference. Sensitivities computed using forward differences are first order accurate. They are computed from the first term in the Taylor series expansion, with respect to a single parameter, as

$$\chi(\theta_j, t_i) = \frac{g(t_i, x(t_i), \theta + he_j) - g(t_i, x(t_i), \theta)}{h}, \tag{8}$$

where h is an appropriate step based on the precision of the model output; if the relative error tolerance of the ordinary differential equation (ODE) is φ, then $h = \sqrt{\varphi}$. The variable e_j represents the unit vector in the j^{th} direction [11]. The advantage of forward differences is that they are easy to compute, requiring $q+1$ evaluations of the model output. On the other hand, sensitivities computed using a central difference scheme given by

$$\chi(\theta_j, t_i) = \frac{g(t_i, x(t_i), \theta + he_j) - g(t_i, x(t_i), \theta - he_j)}{2h} \tag{9}$$

are 2^{nd} order accurate, yet they require $2q$ evaluations of the model.

Since the parameter values and model output can have different units, it may be beneficial to instead compute the relative sensitivity matrix, given by

$$\tilde{\chi} = \frac{\partial y}{\partial \theta}\frac{\theta}{y}. \tag{10}$$

Given that sensitivities are evaluated at a fixed parameter value, typically the nominal or the optimal parameter values, the sensitivities are valid locally in a region near the parameter values. A global sensitivity may be more beneficial if the parameters exhibit a high level of uncertainty [5].

Once the sensitivity matrix has been computed, the sensitivities can be ranked by averaging over time, e.g. using the two-norm, and the parameters can be ordered from most to least sensitive. Using a cutoff of $10\sqrt{\varphi}$, the parameters can be separated into the p sensitive and $q-p$ insensitive parameters.

The parameters of the SIR model have been ranked using the two-norm after computing the relative sensitivities with the forward difference approximation. Based on the results of this analysis, shown in Figure 2, it can be concluded that all four parameters are sensitive with respect to a single parameter.

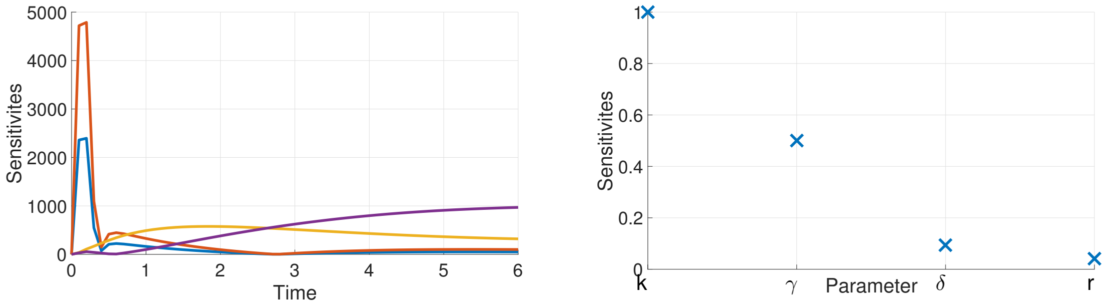

Figure 2: (Left) Relative sensitivity of each parameter with respect to time. (Right) Ranked relative sensitivities

Subset Selection

Although all four parameters of the SIR model are sensitive, they may not all be identifiable. Hence, the sensitive parameters must be analyzed for pairwise correlations. One way to do this is the correlation method [10], which predicts pairwise parameter correlations from evaluation of the correlation matrix evaluated at the nominal parameter values. The correlation matrix is defined as

$$c_{ij} = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}}, \tag{11}$$

where $C = (\chi^T\chi)^{-1}$ is the covariance matrix. In general, a parameter pair (i, j) is correlated if $|c_{ij}| > \gamma$ for $\gamma \to 1$.

Note: The correlation matrix cannot be computed if $\chi^T\chi$ is singular or close to singular. The condition number of χ gives a measure of how close the matrix is to singular. If the condition number is large, then the matrix is ill-conditioned and thus, close to singular. Recall that,

$$\kappa(\chi) = \frac{\sigma_1}{\sigma_n},$$

where σ_1 and σ_n is the largest singular value of χ. The smallest singular value is given by $\sigma_n(\chi) = \sqrt{\lambda_n(\chi^T\chi)}$, where λ_n is the smallest eigenvalue of χ. So if λ_n is small, then $\kappa(\chi)$ will be large. If this is the case, then correlations will have to be manually removed by analyzing the relationships between the parameters of the model and then removing columns of the sensitivity matrix χ corresponding to the correlations.

Returning to the SIR model in equation 1, the ranked sensitivities (from most to least sensitive) are $\theta = (k, \gamma, \delta, r)$ and subset selection reveals that k and γ are correlated. Since γ is less sensitive than k, γ will be fixed while the other three parameters will be optimized.

Parameterization

Sensitivity analysis and subset selection identifies a set of parameters that can be optimized. There are numerous optimization techniques available, both local and global. For this study, we compare two local methods: the Nelder-Mead simplex method [12] and the Levenberg-Marquardt algorithm [4, 6, 7].

Nelder-Mead

Let $R = (r_1, r_2, \ldots, r_d)^T$, where $r_i = Y_i - g(t_i, x(t_i), \theta)$, and define the least-squares cost by

$$J = R^T R. \tag{12}$$

The Nelder-Mead simplex method aims to minimize the cost through direct search. A simplex, a triangle in h dimensions, is formed using the initial guess for the h parameters that are to be optimized. The algorithm is a pattern search that compares the values of the function at the three vertices of the triangle. The worst vertex, that is where the cost is the largest, is rejected and replaced with a new vertex [8].

This process is repeated and terminated once the value of the cost at each of the vertices converges to a single value. Since it does not depend on numerical or analytic gradients, it does not return the sensitivity matrix, covariance matrix, or mean squared error. This process can be executed in MATLAB using *fminsearch.*

Levenberg-Marquardt

The least-squares cost, J can be expressed as

$$\begin{aligned} J(\theta) &= R^T R \\ &= (Y - y)^T (Y - y) \\ &= Y^T Y - 2Y^T y + y^T y. \end{aligned} \tag{13}$$

The Levenberg-Marquardt method uses a combination of the gradient descent method and the Gauss-Newton method [2]. Its aim is to find a perturbation h to the parameters θ that will reduce J. The gradient descent method updates the parameters in the direction of steepest descent, while the Gauss-Newton Method assumes that the least squares function is locally quadratic and then finds the minimum of the quadratic. The Levenberg-Marquardt method finds the value of h by solving

$$[S^T S + \lambda I]h = S^T (Y - y). \tag{14}$$

where small values of λ result in a Gauss-Newton update and large values of λ result in a gradient descent update. The parameter λ is initialized to be large so that first updates are small steps in the steepest-descent direction. As the solution improves, λ is decreased, the Levenberg-Marquardt method approaches the Gauss-Newton method, and the solution typically accelerates to the local minimum [2].

The steepest descent method is a general minimization method that updates parameters in the direction opposite to the gradient of the objective function, J, where the gradient is given by

$$\begin{aligned}\frac{\partial}{\partial\theta}J(\theta) &= 2(Y-y)^T\frac{\partial}{\partial\theta}(Y-y)\\ &= -2(Y-y)^T\left[\frac{\partial y}{\partial\theta}\right]\\ &= -2(Y-y)^T\chi.\end{aligned}$$

The parameter update h that moves the parameters in the direction of steepest descent is given by

$$h_{gd} = \alpha\chi^T(Y-y), \tag{15}$$

where $\alpha > 0$ determines the length of the step in the direction of steepest descent.

The Gauss-Newton method assumes that the objective function is approximately quadratic in the parameters near the optimal solution. The model output evaluated with a slightly perturbed parameter set, $y(t,x,\theta+h)$, can be locally approximated by

$$y(t,x,\theta+h) \approx y(t,x,\theta) + \frac{\partial y}{\partial\theta}h = y + \chi h. \tag{16}$$

Substituting $y + Sh$ for y in Equation 13, we obtain

$$J(\theta+h) \approx Y^TY + y^Ty - 2Y^Ty - 2(Y-y)\chi h + h^T(\chi^T\chi)h.$$

To find the parameter update h that minimizes J, we need $\frac{\partial J}{\partial h} = 0$.

$$\frac{\partial}{\partial h}J(\theta+h) \approx -2(Y-y)\chi h + 2h^T(\chi^T\chi) = 0. \tag{17}$$

Solving for h, we obtain

$$(\chi^T\chi)h = \chi^T(Y-y). \tag{18}$$

Typically, the optimal values for the Nelder-Mead and Levenberg-Marquardt methods are different. However, for the SIR model shown here, both optimizers produce the same results, shown in Table 1.

Table 1: Optimization Results

Parameter	**Nominal**	**Optimal $\pm\Delta\theta$**
k	0.1000	0.1083 ± 0.0147
δ	0.1500	0.1562 ± 0.0321
r	0.6000	0.6112 ± 0.0470

Nominal and optimal parameter values, along with the parameter confidence intervals.

Uncertainty Quantification

Uncertainty quantification is the process of determining parameter uncertainties from uncertainties in model formulation and experimental measurements. The *parameter confidence interval* provides information on the extent of uncertainty involved in estimating the true parameter set [1]. The *Prediction interval* is used to predict where the model response well be at a given data point, not included in the data. It provides information about the distribution of the model output values, not the uncertainty associated with determining the population mean. The *confidence interval* is used to measure the precision of the model in predicting the mean response. It is used to determine how much variation is expected comparing the true model response and the estimated response. Since the prediction interval takes into account the tendency of the model to fluctuate from its mean value, it is wider than the confidence interval.

Parameter Confidence Interval

To compute the parameter confidence interval, recall that the covariance matrix is defined as $C = (\chi^T \chi)^{-1}$, where χ is defined in Equation 7 and the least-squares cost defined in Equation 12. From Equation 6, we know that ϵ_i are i.i.d. $\mathcal{N}_q(0, \sigma^2)$. Then $\hat{\theta} \sim \mathcal{N}_q(\theta, \sigma^2 (S^T S)^{-1})$, where θ is the unknown true parameter set and $\hat{\theta}$ is the (optimal) estimate of θ. The confidence interval for the r^{th} element of $\hat{\theta}$ is given by

$$\hat{\theta} \pm t_{N-q}^{\alpha/2} s \sqrt{C_{jj}} = \hat{\theta} \pm \Delta\theta,$$

where N is the total number of data points, q is the number of parameters (estimated), $t_{N-q}^{\alpha/2}$ is the t-value for the $1 - \alpha/2$ quartile with $N - q$ degrees of freedom, and

$$s^2 = \frac{1}{N-q} J(\hat{\theta})$$

is an estimator of the variance σ^2. The parameter confidence intervals for the optimal parameter values are in Table 1.

Prediction Interval

To obtain a prediction interval for y at $t = t_1, t_2, \ldots, t_d$, let

$$y_i = g(t_i, x(t_i), \theta) + \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2).$$

Then an obvious estimate of y_i is $\hat{y}_i = g(t_i, x(t_i), \hat{\theta})$. For large N, $\hat{\theta}$ is close to the true value θ, so we can use the Taylor series expansion

$$g(t_i, x(t_i), \hat{\theta}) \approx g(t_i, x(t_i), \theta) + g_i^T (\hat{\theta} - \theta),$$

where

$$g_i^T = \left(\frac{\partial g(t_i, x(t_i), \theta)}{\partial \theta_1}, \quad \frac{\partial g(t_i, x(t_i), \theta)}{\partial \theta_2}, \quad \ldots, \quad \frac{\partial g(t_i, x(t_i), \theta)}{\partial \theta_q} \right).$$

Thus,

$$y_i - \hat{y}_i \approx y_i - g(t_i, x(t_i), \theta) - g_i^T(\hat{\theta} - \theta) = \epsilon_i - g_i^T(\hat{\theta} - \theta).$$

This implies that

$$\mathbb{E}[y_i - \hat{y}_i] \approx \mathbb{E}[\epsilon_i] - g_i^T \mathbb{E}(\hat{\theta} - \theta) \approx 0.$$

Moreover,

$$\begin{aligned} \text{var}[y_i - \hat{y}_i] &\approx \text{var}[\epsilon_i] + \text{var}(g_i^T(\hat{\theta} - \theta)) \\ &\approx \sigma^2 + \sigma^2 g_i^T(\chi^T\chi)^{-1} g_i \\ &= \sigma^2(1 + v_0), \end{aligned}$$

where $v_0 = g_i^T(\chi^T\chi)^{-1}g_i$. Thus, $y_i - \hat{y}_i$ is asymptotically $\mathcal{N}(0, \sigma^2(1 + v_0))$. s^2 is independent of y_i and is asymptotically independent of $\hat{\theta}$, so that s^2 is asymptotically independent of $y_i - \hat{y}_i$. Hence, asymptotically

$$\frac{y_i - \hat{y}_i}{s\sqrt{1 + v_0}} \sim t_{N-q}$$

has a t-distribution with $N - q$ degrees of freedom. So the prediction interval for y at $t = t_i$ is given by

$$\hat{y}_i \pm t_{N-q}^{\alpha/2} s(1 + g_i^T(\chi^T\chi)^{-1}g_i)^{1/2}. \tag{19}$$

Confidence Interval

To obtain a confidence interval for the mean response, note that

$$\hat{y}_i = g(t_i, x(t_i), \hat{\theta})$$

is an estimation of the mean response. Then

$$\mathbb{E}[\hat{y}_i] = \bar{y}_i,$$

where $\bar{y}_i$ is the mean response, and

$$\text{var}[\hat{y}_i] = \sigma^2 g_i^T(\chi^T\chi)^{-1}g_i.$$

This implies that

$$\frac{\hat{y}_i - \bar{y}_i}{s\sqrt{v_0}} \sim t_{N-q}$$

has a t-distribution with $N - q$ degrees of freedom. So the confidence interval for y at $t = t_i$ is given by

$$\hat{y}_i \pm t_{N-q}^{\alpha/2} s(g_i^T(\chi^T\chi)^{-1}g_i)^{1/2}. \tag{20}$$

For further details of the derivation, please see [12]. The prediction and confidence intervals are shown in Figure 3.

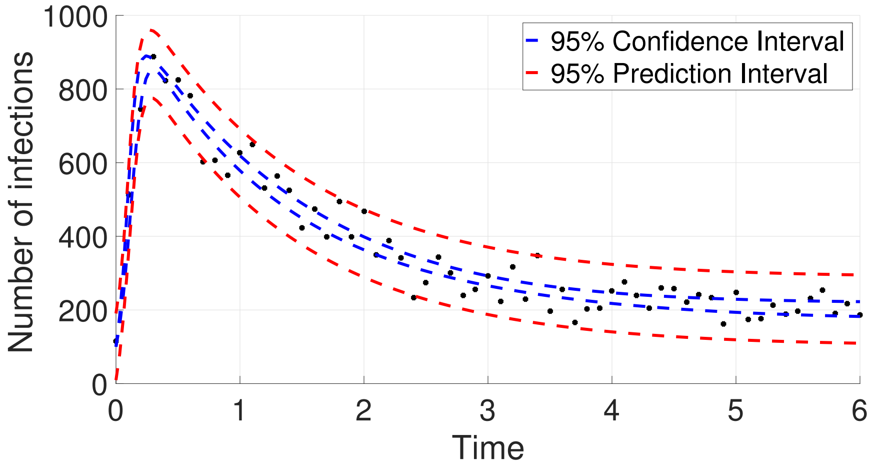

Figure 3: Prediction and confidence intervals

Bayesian Uncertainty Propagation

The methods described above are frequentist uncertainty propagation methods that are relatively computationally inexpensive. More precise results can be achieved through Bayesian methods. This can be attributed to the fact that in general, many frequentist statistics force their uncertainty distributions into a Gaussian shape, while Bayesian methods are more adaptive in how the parameter distributions can vary across different data sets.

To calculate Bayesian credible and prediction intervals we use a sampling method. The prerequisite condition to using a sampling method is to have defined uncertainty distributions for the model inputs (parameters). We use the parameter chains that come from the Delayed Rejection Adaptive Metropolis (DRAM) parameter estimator to meet this condition. More about the nuances of these sampling methods can be found in [13, 3]. Furthermore, the MCMC Matlab toolbox by Haario et al. contains code to implement DRAM and performs Bayesian uncertainty propagation.

DRAM

Frequentist methodology is fundamentally rooted in quantifying uncertainty in terms of repeating the data generating procedure. However, in a Bayesian context, inference is conditioned on the single data set that is observed; this allows for uncertainty about parameters to be expressed with probability distributions. Thus in the Bayesian framework θ refers to a vector of random variables and rather than searching for point estimate, we want to estimate the distribution associated with the random variables. Given the observations $Y = [Y_1, Y_d]$, Bayes formula,

$$\pi(\theta|Y) = \frac{\pi(Y|\theta)\pi(\theta)}{\pi(Y)}$$

is used to describe the relationship between the prior parameter density $\pi(\theta)$, the posterior density $\pi(\theta|Y)$, and the likelihoood $\pi(Y|\theta)$ of observing the data Y for the model given θ. $\pi(Y)$ is the marginal density of the data but in practice really only functions as the a normalization factor and can be determined by $\int \pi(Y|\theta)\pi(\theta)d\theta$

Assuming the statistical hypothesis in equation (6) it follows that

$$\pi(Y|\theta) = \frac{e^{-J(\theta)/2\sigma^2}}{(2\pi\sigma^2)^{d/2}},$$

where $J(\theta)$ is typically defined as the sum of squares error, however it can be replaced with any cost function that satisfies can be derived by assuming (6). With the likelihood function $\pi(Y|\theta)$ given, it is possible to estimate the posterior density $\pi(\theta|Y)$ given a prior $\pi(\theta)$. A direct approach to estimate the posterior density $\pi(\theta|Y)$ would involve estimating the integral $\pi(Y) = \int \pi(Y|\theta)\pi(\theta)d\theta$. While this route is theoretically possible, the evaluation of high-dimensional integrals is a difficult and expensive task and is currently an active research area (see sparse grid and quasi-Monte Carlo methods).

A more practical alternative is to randomly sample directly from the density $\pi(Y|\theta)$, thus the usage of Markov Chain Monte Carlo (MCMC) methods. DRAM combines two methods for improving efficiency of Metropolis-Hastings type MCMC algorithms: Delayed Rejection and Adaptive Metropolis. Metropolis algorithms are acceptance-rejection algorithms that accept new parameter samples only if the likelihood of the new candidate is higher than the current sample. Delayed rejection allows the algorithm to try additional proposals per step if the initially proposed step is not accepted. This increases the acceptance rate and thereby mixing of the chain, which results in better estimates of the posterior densities. Adaptive metropolis allows the metropolis algorithm to update the covariance matrix based on the history of the chain. This helps the algorithm make better proposals, and move to the correct posterior distribution faster; reducing what is called the “burn-in” period.

Note that updating the proposal function using history of the chain breaks the Markov property, and other properties need to be established for guaranteed sampling from the posterior distribution [13, 3].

Recall that correlation analysis can identify pairwise correlations among parameters. Results from Bayesian sampling methods such as DRAM accomplish this quite naturally and capture the entirety of the nonlinear interactions between parameters by representing them as a joint density of random variables. The plots shown in Figure 4 show pairwise correlations between γ and k, as well as between δ and r. While correlation analysis is far less computationally expensive than DRAM, it only provides a first-order linearization of parameter interactions. Thus correlation analysis may fail to capture the nuance of parameter interactions in models with complicated nonlinear parameterizations.

On the other hand however, running DRAM typically requires performing some form of frequentist parameter estimation to set a good starting point of the sampling to prevent the algorithm from getting stuck in an undesirable local minimum of the predefined cost function $J(\theta)$. This is to say that tools such as correlation analysis might be a better option over DRAM in the early stages of model calibration. The real benefit and power of using Bayesian MCMC algorithms like DRAM comes from the

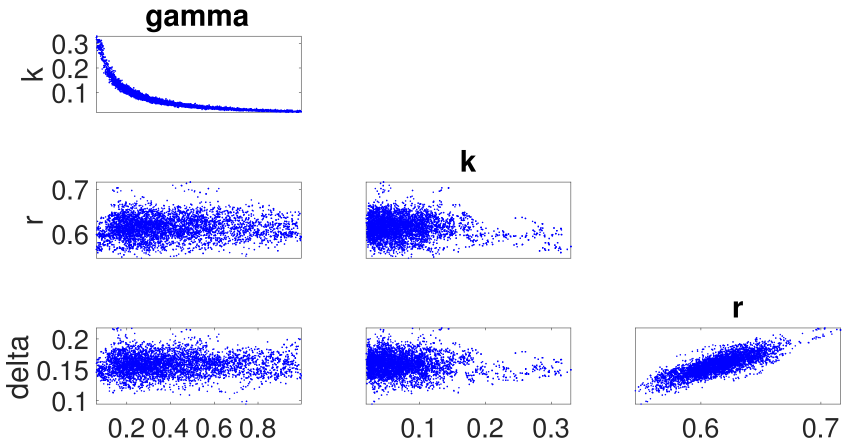

Figure 4: DRAM correlations

ease in propagating the parameter densities through the model the calculate Bayesian credible and prediction intervals to quantify the uncertainty of our model predictions. The prediction and confidence intervals for the SIR model produced by DRAM are shown in Figure 5.

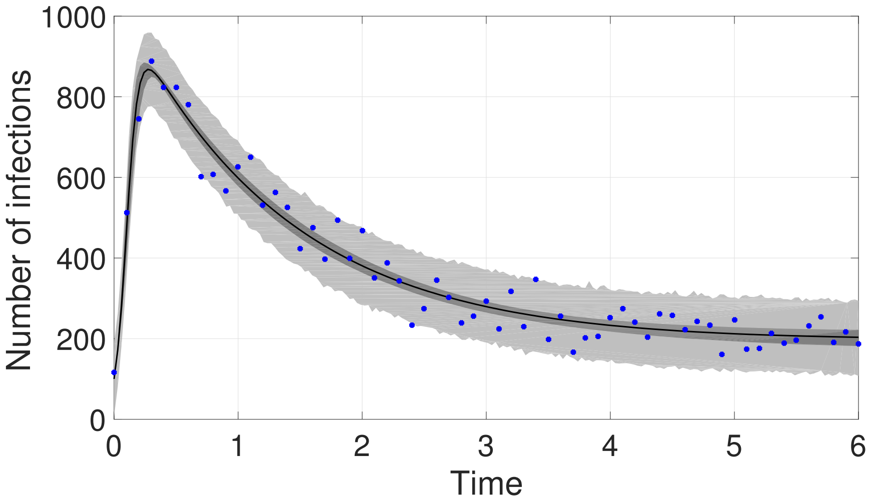

Figure 5: Prediction and confidence intervals produced by DRAM

References

[1] H.T. Banks and H.T. Tran. *Mathematical and Experimental Modeling of Physical and Biological Processes*. CRC Press, Boca Raton, FL, 2009.

[2] A. Bjorcl. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

[3] H. Haario, M. Laine, A. Mira, and E. Saksman. Dram: Efficient adaptive mcmc. *Statistics and Computing*, 16:339–354, 2006.

[4] C.T. Kelley. *Iterative Methods for Optimization.* SIAM, Philadelphia, PA, 1999.

[5] A. Kiparissides, S. S. Kucherenko, A. Mantalaris, and E. N. Pistikopoulos. Global sensitivity analysis challenges in biological systems modeling. *Industrial & Engineering Chemistry Research*, 48:7168–7180, 2009.

[6] K. Madsen, H. B. Nielsen, and O. Tingleff. *Methods for Nonlinear Least Squares Problems.* Denmark, 2004.

[7] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM*, 11:431–441, 1963.

[8] J. H. Mathews and K. D. Fink. *Numerical Methods Using MATLAB.* Simon & Schuster, 1998.

[9] H. Miao, X. Xia, A. S. Perelson, and H. Wu. On identifiability of nonlinear ODE models and applications in viral dynamics. *SIAM Rev Soc Ind Appl Math*, 53:3–39, 2011.

[10] M. S. Olufsen and J. T. Ottesen. A practical approach to parameter estimation applied to model predicting heart rate regulation. *J Math Biol*, 67:39–68, 2013.

[11] S. R. Pope, L. M. Ellwein, C. L. Zapata, V. Novak, C. T. Kelley, and M. S. Olufsen. Estimation and identification of parameters in a lumped cerebrovascular model. *Math Biosci Eng*, 6:93–115, 2009.

[12] G. A. F. Seber and C. J. Wild. *Nonlinear Regression.* John Wiley & Sons, Inc., Hoboken, NJ, 2003.

[13] R. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications.* SIAM, 2013.