MCMC calibration of a receptor-ligand interaction model

Polly Jeffrey

February 2022

1 Introduction

This document provides some information about calibrating a deterministic ordinary differential equation mathematical model using Markov Chain Monte Carlo in Python. Here, the Metropolis-Hastings algorithm is used. The biological system of interest is an experimental setup in which cells are treated with ligand and the ligand can bind reversibly to receptor molecules on the surface of the cell. It is assumed that the concentration of ligand-bound receptors can be observed over time. Initially there is a concentration of 0 ligand-bound receptors. The reactions underlying the mathematical model are depicted in Figure 1, where the ligand binds to the receptor with rate k_f and dissociates the receptor with rate k_r. Assuming that the concentration of ligand-bound receptor is measured in units of nM, the rate k_f has units of nM^{-1}s^{-1} and the rate k_r has units of s^{-1}.

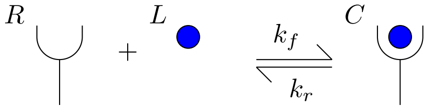

Figure 1: A diagrammatic representation of a single receptor, R, binding with a ligand, L, to form a receptor ligand complex, C.

It is assumed that we know the initial concentrations of receptor and ligand, and that they are $R^T = R(0) = 0.09$ nM and $L^T = L(0) = 0.1$ nM, respectively (values taken from [1]). Given that there are no synthesis or degradation reactions in the model, the total concentrations of receptor and ligand are conserved over time and hence we can write

$$\begin{aligned} R^T &= R(t) + C(t) \implies R(t) = R^T - C(t), \\ L^T &= L(t) + C(t) \implies L(t) = L^T - C(t), \end{aligned}$$

meaning that $R(t)$ and $L(t)$ are implicitly tracked over time by tracking only $C(t)$. Under the assumption of mass action kinetics, the differential equation describing the time evolution of $C(t)$ is given by

$$\frac{dC}{dt} = k_f(R^T - C)(L^T - C) - k_r C, \tag{1}$$

for all $t \geq 0$ but where time has been ommited in the species notation for ease of reading.

2 Data generation

For this toy system, some data can be generated synthetically, which we will then attempt to fit using MCMC for parameter inference. The model is simulated for 100 seconds using the parameter values $k_f = 0.092$ nM^{-1}s^{-1} and $k_r = 0.02$ s^{-1}, taken from [1] in which the authors mathematically model a system involving the epidermal growth factor receptor. For the time points $t \in T = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, a data point is generated by adding to $C(t)$ a random value simulated from a normal distribution with mean 0 and standard deviation 0.003. Three repeats, $r \in R = \{1, 2, 3\}$, of the data are generated in this way and the model simulation and synthetic data can be seen in Figure 2.

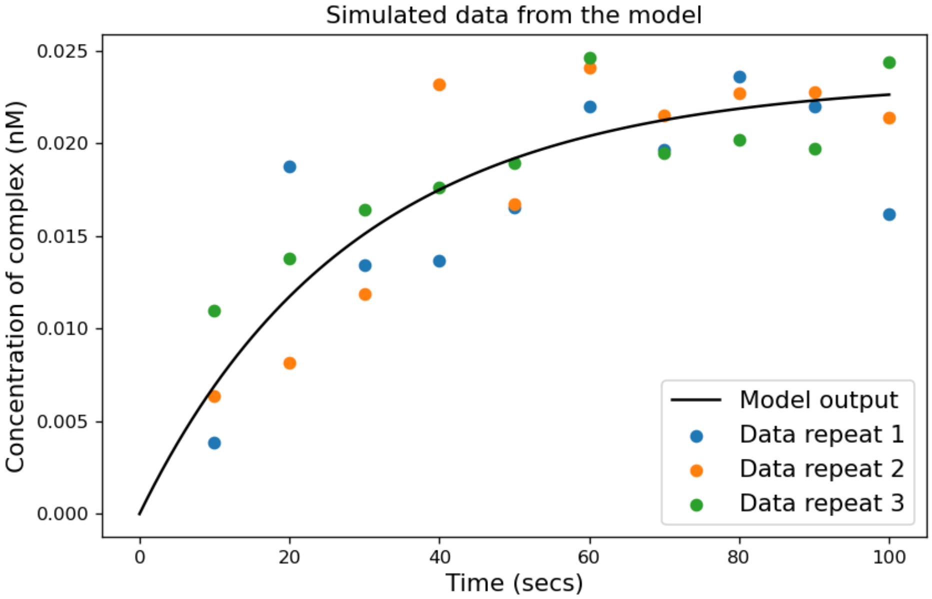

Figure 2: A model solution generated from Equation 1, along with three replicates of simulated data.

3 Bayesian inference

Bayes' theorem is introduced by [2] and in many other probability and statistics texts, as a way of relating conditional probabilities of events A and B, and is given as

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A)\mathbb{P}(B|A)}{\mathbb{P}(B)}.$$

In statistical inference, Bayes' theorem is instead formulated as

$$\pi(\boldsymbol{\theta}|\boldsymbol{D}) = \frac{\pi(\boldsymbol{\theta})\pi(\boldsymbol{D}|\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} \pi(\boldsymbol{D}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}}, \tag{2}$$

where $\boldsymbol{\theta}$ is a model parameter, or vector of model parameters, and $\boldsymbol{D}$ is the observed data. In this formulation, $\pi(\boldsymbol{\theta})$ is known as the *prior distribution*, and encodes the users prior beliefs about the parameter(s). If the user has a strong prior knowledge of a parameter value then an *informative* prior can be used, such as a normal or beta distribution, giving more density to regions of the parameter space in which the true value is thought to lie. If however, the user has very little prior knowledge of a parameter value then a *non-informative* prior can be used, such as a uniform distribution, which would only require an upper and lower bound for the parameter value. $\pi(\boldsymbol{D}|\boldsymbol{\theta})$ is the *likelihood* of observing the data $\boldsymbol{D}$, given the parameter(s) $\boldsymbol{\theta}$. Finally, $\pi(\boldsymbol{\theta}|\boldsymbol{D})$ is the *posterior distribution* that the user aims to evaluate. The integral on the denominator of Equation (2) is just a normalisation constant, and hence a simpler form of Equation (2) is the proportionality equation,

$$\pi(\boldsymbol{\theta}|\boldsymbol{D}) \propto \pi(\boldsymbol{\theta})\pi(\boldsymbol{D}|\boldsymbol{\theta}).$$

One method of obtaining the posterior distribution, is via MCMC.

3.1 MCMC

In this section, the mathematical model given by Equation 1 is fitted to the experimental data generated in Section 2 using the Metropolis-Hastings algorithm, a commonly used form of MCMC. For simplicity, we assume that the parameter k_r is known and hence we need only infer a posterior distribution for the parameter k_f. The standard deviation of the data, denoted by σ, will also be inferred in the MCMC. To run the algorithm we need a prior distribution for the model parameter k_f, a proposal distribution to indicate how to choose a new parameter based on the current parameter and we also need to know the form of the likelihood.

3.1.1 Prior distribution

Here a normal prior distribution for k_f is chosen such that $k_f = 10^r$, where $r \sim N(-2, 1)$. This prior will be denoted $\pi(k_f)$.

3.1.2 Proposal distributions

A proposal distribution must be defined for both k_f and σ, so that a new parameter set can be generated at each iteration of the algorithm. For a general parameter θ, the proposal distribution is denoted $Q(\theta'|\theta)$, where θ' is the new parameter generated. The proposal distributions chosen here are $Q(k_f'|k_f) = N(k_f, 0.1)$ and $Q(\sigma'|\sigma) = N(\sigma, 0.01)$.

3.1.3 Likelihood function

We know that the error between the model and the data is normally distributed and hence the likelihood function takes the form of a normal with mean $C(t)$ for any time t, and unknown variance σ. For given values of the parameters k_f and σ therefore, the likelihood of a data point $d(r, t)$ for repeat r and at time point t is given by

$$\pi(d(r,t)|k_f,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{d(r,t)-C(t)}{\sigma}\right)^2\right].$$

The total likelihood for all data points $\boldsymbol{D} = d(r, t)$ with $r \in R$ and $t \in T$, given parameters k_f and σ is then defined as

$$\pi(\boldsymbol{D}|k_f,\sigma) = \prod_{r\in R}\prod_{t\in T}\frac{1}{\sigma\sqrt{2\pi}} exp\left[-\frac{1}{2}\left(\frac{d(r,t)-C(t)}{\sigma}\right)^2\right].$$

In practice, the log-likelihood

$$log(\pi(\boldsymbol{D}|k_f,\sigma)) = \sum_{r\in R}\sum_{t\in T} -log(\sigma) - log(2\pi) - \frac{1}{2}\left(\frac{d(r,t)-C(t)}{\sigma}\right)^2,$$

is used to avoid numerical problems.

3.1.4 Metropolis-hastings algorithm

With the prior distribution, proposal distributions and likelihood function in hand, the Metropolis-Hastings algorithm can be run. The algorithm consists of simulating a Markov chain whose stationary distribution is the posterior distribution of interest. An initial parameter value, k_f, is sampled from its prior distribution and at each iteration of the algorithm, a new value for the parameter, k_f', is proposed using the proposal distribution. Similarly, an initial value for σ is chosen and at each iteration, a new value σ' is proposed. At each iteration the new parameters are either accepted, or rejected, based on the acceptance ratio

$$\alpha = min\left(1, \frac{\pi(\boldsymbol{D}|k_f',\sigma')\pi(k_f')}{\pi(\boldsymbol{D}|k_f,\sigma)\pi(k_f)}\right).$$

If $\alpha = 1$, then k_f' is accepted with probability 1, and if $\alpha < 1$, k_f' is accepted with probability α. If $\alpha < 1$, the acceptance decision can be implemented computationally be simulating a random uniform variable, $u \sim U(0, 1)$, and accepting k_f' if $u < \alpha$. In practice, one might use the log-likelihood instead of the likelihood, in which case one would accept k_f' with probability 1 if

$$log(\pi(\boldsymbol{D}|k_f',\sigma')) + log(\pi(k_f')) > log(\pi(\boldsymbol{D}|k_f,\sigma)) + log(\pi(k_f)),$$

and otherwise accept k_f' with probability α if for $u \sim U(0, 1)$,

$$u < exp[log(\pi(\boldsymbol{D}|k_f',\sigma')) + log(\pi(k_f')) - log(\pi(\boldsymbol{D}|k_f,\sigma)) - log(\pi(k_f))].$$

The algorithm should be run for a large number of iterations, and then the posterior distributions for the parameters are found by considering the last values in the Markov chains, representing the stationary distributions of the chains. The beginning of the Markov chain, is a “burn-in” section and the parameter values accepted at these iterations are not included in the final posterior distributions. Here, given that the model is very simple and that we are only attempting to estimate two parameters (k_f and σ), the algorithm is run until 500 parameter sets are accepted. The final 100 parameter sets in the chain are then taken as the posterior distributions for the parameters.

3.1.5 Results

In this section the results of the MCMC are shown. Figure 3 shows a trace plot of all of the values of k_f and σ that were accepted by the algorithm. Of the 500 accepted values, the last 200 in the chain were chosen as the posterior distributions, such that the burn-in period is excluded. Kernel density estimates of these posterior distributions are seen in Figure 4 as well as the prior distribution for k_f. The dashed lines represent the values of the parameters used to generate the synthetic data (see Section 2).

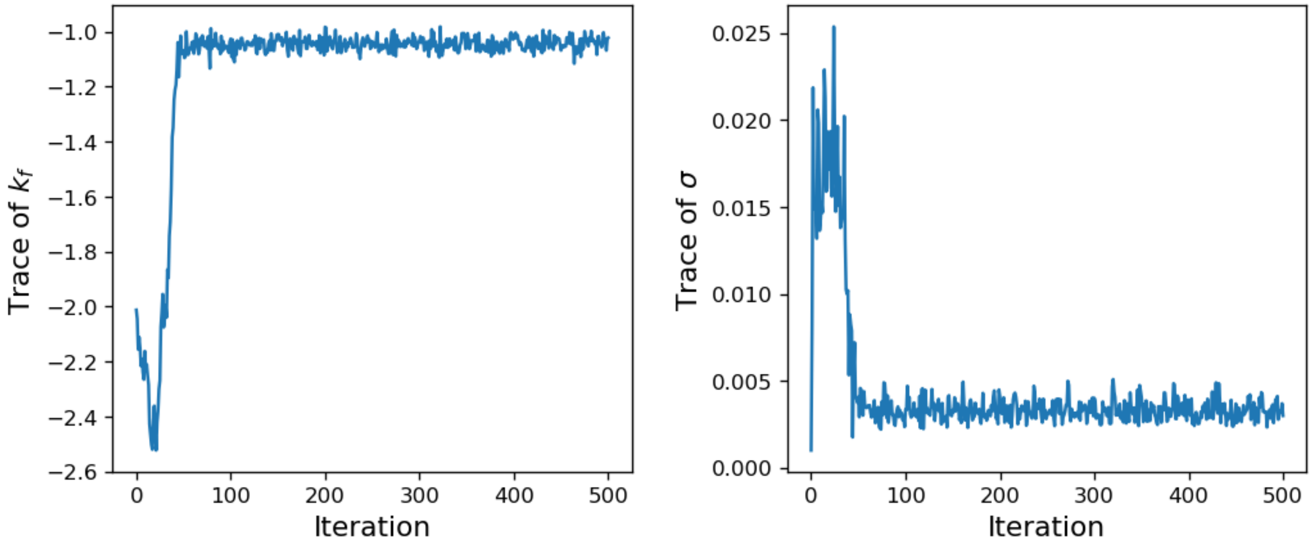

Figure 3: Trace plot showing the accepted values in the Markov chain for each parameter in the model from the MCMC.

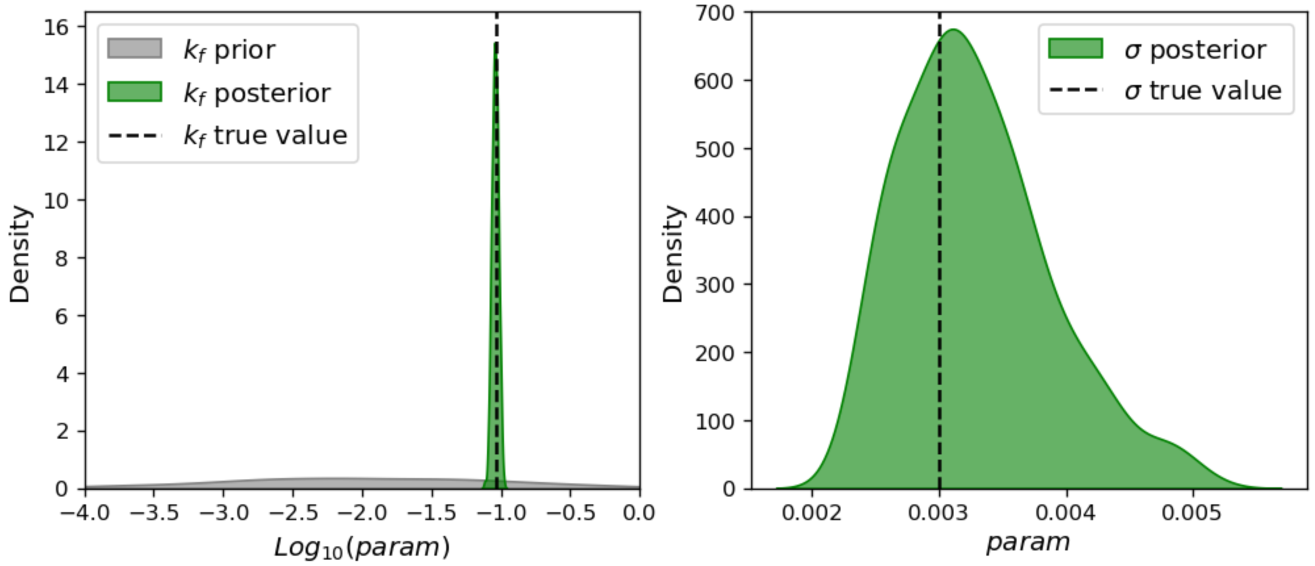

Figure 4: Prior and posterior distributions for the parameters in the model from the MCMC. Also shown are the values of the parameters used to generate the synthetic data.

Finally, the model fit to the data is seen in Figure 5 and it can be seen that the model well represents the data points.

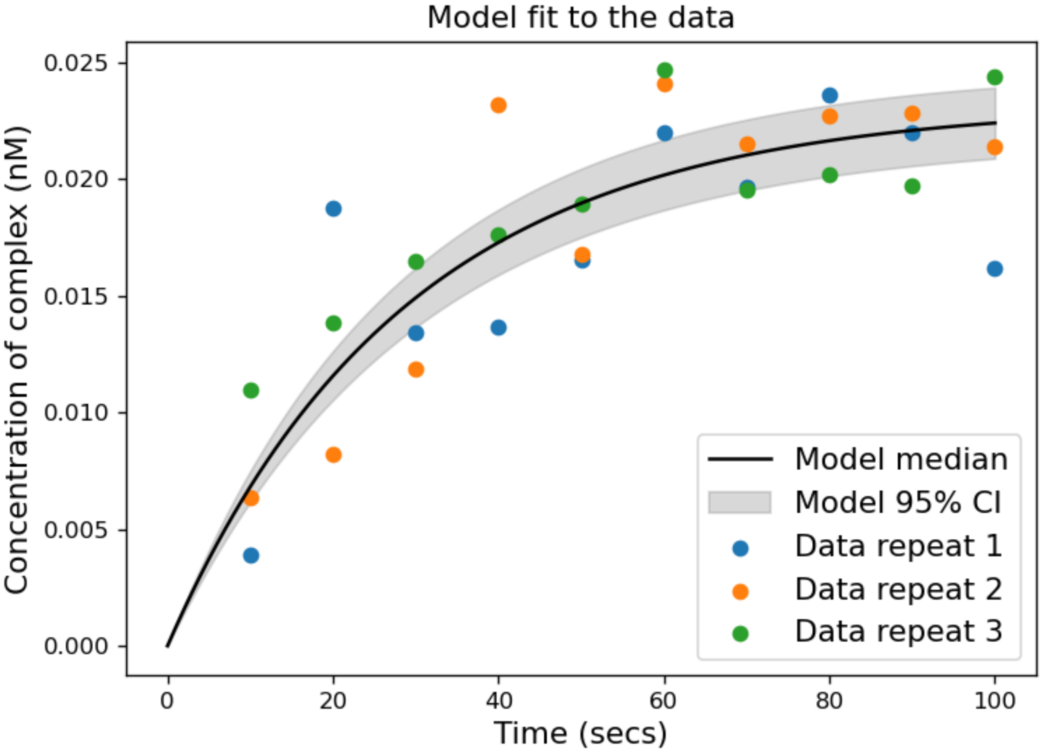

Figure 5: Median and 95% credible interval of model simulations using the parameter values of the posterior distributions. Also shown are the synthetic data points.

References

[1] Noga Kozer, Dipak Barua, Suzanne Orchard, Eduoard C Nice, Antony W Burgess, William S Hlavacek, and Andrew HA Clayton. Exploring higher-order egfr oligomerisation and phosphorylation—a combined experimental and theoretical approach. *Molecular BioSystems*, 9(7):1849–1863, 2013.

[2] Joseph K Blitzstein and Jessica Hwang. *Introduction to probability*. Crc Press Boca Raton, FL, 2015.