Bayesian Sequential Inference for Stochastic Kinetic Biochemical Network Models

Andrew Golightly* and Darren J. Wilkinson

University of Newcastle upon Tyne, NE1 7RU, U.K.

4th August 2005

Abstract

As post-genomic biology becomes more predictive, the ability to infer rate parameters of genetic and biochemical networks is paramount. In this paper we explore the Bayesian estimation of stochastic kinetic rate constants governing dynamic models of intra-cellular processes. The underlying model is replaced by a diffusion approximation where a white noise term represents stochastic behaviour and the model is identified using discrete-time (and often incomplete) data that is subject to measurement error. Sequential MCMC methods are then used to sample the model parameters on-line in several data-poor contexts. The methodology is applied to the estimation of parameters in a simple prokaryotic auto-regulatory gene network.

1 Introduction

Traditionally, the time evolution of a biochemical network is described by a set of coupled differential equations derived using the law of mass action and the concentrations of each species. This widely used approach, however, assumes that the system is both continuous and deterministic. In reality, chemical reactions are intrinsically stochastic and occur as discrete events resulting from random molecular collisions (Gillespie 1977). Although relatively little work has addressed the stochasticity of biochemical networks (Arkin, Ross & McAdams 1998, McAdams & Arkin 1999), it is clear that many important intra-cellular processes such as signal transduction and gene expression can only be effectively described by stochastic processes. Stochastic effects at this level can have large significance even on high-level outcomes such as an organism’s ageing (Finch & Kirkwood 2000).

In order to perform analysis on a stochastic biochemical network model, it is essential that each network parameter is obtained (Kitano 2001). The resulting problem is known as reverse engineering (Bower & Bolouri 2000) and presents the challenge of how to estimate key rate parameters given observed time course data. Although inference for “exact” stochastic kinetic models is possible, it is computationally problematic for models of realistic size and complexity (Boys, Wilkinson & Kirkwood 2004). We therefore work with the diffusion approximation which though often inadequate for simulation, can be satisfactory for inferential purposes (Golightly & Wilkinson 2005a).

*a.golightly@ncl.ac.uk
Typically, since biochemical data arrive at discrete times, yet the model is formulated in continuous time, it is natural to work with the first order Euler discretisation. As inter-observation times are usually too large to be used as a time step in the Euler scheme, we follow Pedersen (1995) and augment the observed low frequency data by introducing \( m - 1 \) latent data points in between each pair of observations. Markov chain Monte Carlo (MCMC) methods which sample the posterior distribution of model parameters have been proposed independently by Jones (1997), Elerian, Chib & Shephard (2001) and Eraker (2001). Unfortunately, inference can be complicated if the amount of augmentation is large. It is well known that high dependence between the parameters and missing data results in slow rates of convergence of basic algorithms such as Gibbs samplers (Roberts & Stramer 2001).

In this paper, we utilise recently developed simulation-based sequential algorithms (Golightly & Wilkinson 2005b) to conduct inference for a partially and discretely observed stochastic kinetic model. Our proposed simulation filter does not break down as either the degree of augmentation or the number of observations increases and can be implemented for multiple, partially observed datasets.

The structure of this paper is organised as follows. In Section 2, methods for modeling stochastic kinetics are described; Section 2.1 describes the continuous time Markov process model, and Section 2.2 gives the diffusion approximation. Inference for non-linear, partially observed diffusion models is outlined in Section 3 before an application is presented in Section 4. Conclusions are drawn in Section 5.

## 2 Stochastic Kinetics

### 2.1 Continuous Time Markov Process Model

We typically consider a system of reactions involving \( k \) species \( Y_1, Y_2, \ldots, Y_k \) and \( r \) reactions \( R_1, R_2, \ldots, R_r \) in thermal equilibrium inside some fixed volume. The system will take the form

\[
\begin{align*}
R_1: & \quad u_{11}Y_1 + u_{12}Y_2 + \ldots + u_{1k}Y_k \longrightarrow q_{11}Y_1 + q_{12}Y_2 + \ldots + q_{1k}Y_k \\
R_2: & \quad u_{21}Y_1 + u_{22}Y_2 + \ldots + u_{2k}Y_k \longrightarrow q_{21}Y_1 + q_{22}Y_2 + \ldots + q_{2k}Y_k \\
\vdots & \quad \vdots \quad \vdots \quad \vdots \\
R_r: & \quad u_{r1}Y_1 + u_{r2}Y_2 + \ldots + u_{rk}Y_k \longrightarrow q_{r1}Y_1 + q_{r2}Y_2 + \ldots + q_{rk}Y_k
\end{align*}
\]

where, \( u_{ij} \) is the stoichiometry associated with the \( j \)th reactant of the \( i \)th reaction and \( q_{ij} \) is the stoichiometry associated with the \( j \)th product of the \( i \)th reaction. Each reaction, \( R_i \), has a stochastic rate constant, \( c_i \), and a rate law or hazard, \( h_i(Y, c_i) \), where \( Y = (Y_1, Y_2, \ldots, Y_k)' \) is the current state of the system and each hazard is determined by the order of reaction \( R_i \) under an assumption of mass action kinetics. Note that for transparency, we denote by \( Y \) both the species and the number of molecules it represents in the system. We may represent (1) somewhat more compactly as

\[
UY \longrightarrow QY,
\]

where \( U = (u_{ij}) \) and \( Q = (q_{ij}) \) are \( r \times k \) dimensional matrices (obtained from the stoichiometry of the system). As the net effect of reaction \( i \) is a change of \( a_{ij} = q_{ij} - u_{ij} \), the reaction network can be represented by the (net effect reaction) matrix \( A = Q - U \).

Stochastic models of cellular processes are reasonably well developed and are commonly based on techniques for solving the “chemical master equation”. The main element of the master equation is the function, \( P(Y_1, Y_2, \ldots, Y_k; t) \) which gives the
probability that there will be at time $t$, $Y_1, Y_2, \ldots, Y_k$ molecules of each respective species. We write this function as the sum of the probabilities of the number of ways in which the network can arrive in state $Y = (Y_1, Y_2, \ldots, Y_k)$ at time $t + \Delta t$ to give

$$P(Y; t + \Delta t) = \sum_{i=1}^{r} h_i(Y - A_i, c_i) P(Y - A_i; t) \Delta t +$$

$$+ \left\{ 1 - \sum_{i=1}^{r} h_i(Y, c_i) \Delta t \right\} P(Y; t) + o(\Delta t) \quad (2)$$

where $A_i$ denotes the $i^{th}$ row of the net effect matrix $A$. Rearrangement of (2) and taking $\Delta t \to 0$ leads to the master equation,

$$\frac{\partial}{\partial t} P(Y; t) = \sum_{i=1}^{r} \{ h_i(Y - A_i, c_i) P(Y - A_i; t) - h_i(Y, c_i) P(Y; t) \} , \quad (3)$$

further details of which have been given by van Kampen (2001) and Doraiswamy & Kulkarni (1987) among others. Although the master equation is exact, it is only tractable for a handful of cases and those exactly solvable cases have been summarised by McQuarrie (1967). Therefore, stochastic models are typically examined using a discrete event simulation algorithm known in the physical sciences as the “Gillespie algorithm” (Gillespie 1977). Although using the latter algorithm is straightforward for simulation, inference for “exact” stochastic-kinetic models is computationally problematic for models of realistic size and complexity (Boys et al. 2004). We therefore use a continuous approximation of (3) – the diffusion approximation.

### 2.2 The Diffusion Approximation

By assuming that the jumps of the Markov process governed by (3) are “small” and that the solution, $P(Y; t)$, varies slowly with $Y$, we can expand the first term in (3) by means of a second order Taylor expansion to give the Fokker-Planck equation (van Kampen 2001). Formally, for a $k$ dimensional process $Y(t)$ with components $Y_1(t), \ldots, Y_k(t)$ the nonlinear Fokker-Planck equation is given by,

$$\frac{\partial}{\partial t} P(Y; t) = -\sum_{i=1}^{k} \frac{\partial}{\partial Y_i} \{ \mu_i(Y) P(Y; t) \} + \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{\partial^2}{\partial Y_i \partial Y_j} \{ \beta_{ij}(Y) P(Y; t) \} , \quad (4)$$

where we define the infinitesimal means for $i = 1, \ldots, k$ by

$$\mu_i(Y) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \mathbb{E}[\{ Y_i(t + \Delta t) - Y_i(t) \}|Y(t) = Y] \quad (5)$$

and the infinitesimal second moments for $i, j = 1, \ldots, k$ by

$$\beta_{ij}(Y) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \text{Cov}([Y_i(t + \Delta t) - Y_i(t)], [Y_j(t + \Delta t) - Y_j(t)]|Y(t) = Y) . \quad (6)$$

Now suppose at time $t$, the state of the system is $Y(t) = (Y_1(t), \ldots, Y_k(t))'$ = $Y$ so that the hazards of $R_1, R_2, \ldots, R_r$ are $h_1(Y, c_1), h_2(Y, c_2), \ldots, h_r(Y, c_r)$. Let $N_i$ denote the number of type $i$ reactions occurring in the interval $(t, t + \Delta t]$. Then for “small” time $\Delta t, N_i \approx \text{Poisson}(h_i(Y, c_i) \Delta t)$ (due to the assumption of constant reaction hazard) and the change in the number of molecules of $Y_j$ is given by

$$Y_j(t + \Delta t) - Y_j(t) = a_{1j} N_1 + a_{2j} N_2 + \ldots + a_{rj} N_r . \quad (7)$$
For each increment \( Y_j(t + \Delta t) - Y_j(t), j = 1, \ldots, k \) given by (7), we calculate the infinitesimal means and variances through straightforward application of (5) and (6). It can be shown under certain conditions (see Kloeden & Platen (1992)) that the solution of (4) satisfies an Itô stochastic differential equation (SDE),

\[
dY(t) = \mu(Y, \Theta) \, dt + \beta(Y, \Theta) \, dW(t)
\]

where \( \mu(Y, \Theta) \) is the column vector of \( \mu_i(Y) \) (known as drift), \( \beta(Y, \Theta) \) is any matrix satisfying \( \beta'(\beta \beta') = [\beta_{ij}(Y)] = \beta(Y) \) (known as the diffusion matrix) and we let both functions depend explicitly on the parameter vector \( \Theta = (c_1, c_2, \ldots, c_r)' \). Finally, \( dW(t) = (dW_1(t), \ldots, dW_k(t))' \) is the increment of (standard, \( k \) dimensional) Brownian motion. For a reaction network with net effect matrix \( A \), we may compute

\[
\mu(Y, \Theta) = A' h(Y, \Theta), \beta(Y, \Theta) = A' \text{diag}\{h(Y, \Theta)\} A.
\]

where \( h(Y, \Theta) \) is the column vector of hazards \( h_i(Y, c_i) \). Further details of the diffusion approximation can be found in Allen (2002).

2.3 Example: Prokaryotic Auto-regulatory Gene Network

Transcriptional regulation has been studied extensively in both prokaryotic and eukaryotic organisms (see, for example McAdams & Arkin (1999), Latchman (2002) and Ng, Wilkinson & Kirkwood (2004)). In a simple model of prokaryotic auto regulation, a protein (\( I \)) coded for by a gene (\( i \)) represses its own transcription and also the transcription of another gene, (\( g \)) by binding to a regulatory region upstream of the gene. The transcription of a gene into mRNA is facilitated by an enzyme, RNA-polymerase and this process begins with the binding of this enzyme to a site on the gene called a promoter. After the initial binding, RNA-polymerase travels away from the promoter along the gene, synthesising mRNA as it moves. In our model, transcription is repressed by a repressor protein, \( I \), which can bind to sites on the DNA known as operators. We simplify the repression mechanisms with the reactions,

\[
\begin{align*}
R_1 & : \quad I + i \quad \rightarrow \quad I \cdot i \\
R_2 & : \quad I \cdot i \quad \rightarrow \quad I + i \\
R_3 & : \quad I + g \quad \rightarrow \quad I \cdot g \\
R_4 & : \quad I \cdot g \quad \rightarrow \quad I + g
\end{align*}
\]

We represent the transcription of \( i \), the binding of a ribosome to mRNA, the translation of mRNA and the folding of the resulting polypeptide chain into a folding protein, \( I \), by

\[
\begin{align*}
R_5 & : \quad i \quad \rightarrow \quad i + r_i, \quad R_6 & : \quad r_i \quad \rightarrow \quad r_i + I
\end{align*}
\]

Similarly, we represent the transcription of \( g \) and translation mechanism by

\[
\begin{align*}
R_7 & : \quad g \quad \rightarrow \quad g + r_g, \quad R_8 & : \quad r_g \quad \rightarrow \quad r_g + G
\end{align*}
\]

Finally, the model is completed by mRNA degradation,

\[
\begin{align*}
R_9 & : \quad r_i \quad \rightarrow \quad \emptyset, \quad R_{10} & : \quad r_g \quad \rightarrow \quad \emptyset
\end{align*}
\]

and protein degradation,

\[
\begin{align*}
R_{11} & : \quad I \quad \rightarrow \quad \emptyset, \quad R_{12} & : \quad G \quad \rightarrow \quad \emptyset
\end{align*}
\]
Although the model offers a simplistic view of the mechanisms involved in gene-regulation, it will provide insight into how inference might be done in more complex networks. For a detailed discussion of gene regulation see Ptashne (1992) and Latchman (2002).

We now turn our attention to calculating the diffusion approximation for the model given by (10)-(14). We order the species by setting $Y = (l, G, l\cdot i, l\cdot g, i, g, r_l, r_g)$ and use the stoichiometry of the system to obtain the net effect matrix,

$$A' = \begin{pmatrix}
-1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix} \qquad (15)$$

Now assume for reaction $i$ a stochastic rate constant of $c_i$ and consider the time evolution of the system as a Markov process with state $Y(t) = Y$ at time $t$. Reactions $R_1$ and $R_3$ are second order and their hazards can be computed (using the law of mass action) as $h_1(Y, \Theta) = c_1 l i$ and $h_3(Y, \Theta) = c_3 l g$. As the remaining reactions are first order, their hazards are straight forward to compute.

Before calculation of $\mu(Y, \Theta)$ and $\beta(Y, \Theta)$ it should be noted that the net effect matrix $A$ is not of full rank (as the number of molecules of $l\cdot i$ and $l\cdot g$ are related to the number of molecules of $i$ and $g$ respectively). Inspection of (15) reveals that adding row 3 of $A'$ to row 5 implies

$$l\cdot i + i = K_1 \qquad (16)$$

and similarly, adding row 4 to row 6 yields

$$l\cdot g + g = K_2 \qquad (17)$$

where $K_1$ and $K_2$ are conservation constants. As this rank degeneracy will cause problems for the inference method considered in Section 3, we remove rows 3 and 4 from $A'$ to obtain $A$ of full rank. We then use (16) and (17) to substitute $K_1 - i$ and $K_2 - g$ for $l\cdot i$ and $l\cdot g$ respectively to reduce our model to one involving just 6 chemical species, $Y = (l, G, i, g, r_l, r_g)'$. The full diffusion approximation can then be computed using (9), for example,

$$\mu(Y, \Theta) = \begin{pmatrix}
c_2(K_1 - i) + c_4(K_2 - g) + c_6 r_l - c_1 l i - c_3 l g - c_{11} l \\
c_8 r_g - c_{12} G \\
c_2(K_1 - i) - c_1 l i \\
c_4(K_2 - g) - c_3 l g \\
c_5 l - c_9 r_l \\
c_7 g - c_{10} r_g \\
\end{pmatrix}.$$ 

Note that our parameter vector $\Theta$ consists of all stochastic rate constants and is given by $\Theta = (c_1, c_2, \ldots, c_{12})'$. For a further discussion of how to calculate the diffusion approximation for a given reaction network, see Golightly & Wilkinson (2005a).
3 Inference for non-linear Diffusion Models

3.1 Models

We consider inference for a \(d\)-dimensional Itô Diffusion that satisfies a stochastic differential equation of the form given by (8) and assume that the conditions under which the SDE can be solved for \(Y(t)\) are satisfied (Øksendal 1995).

Often, \(Y(t)\) will consist of both observable and unobservable components. To deal with this, we define \(Y(t) = (X(t), Z(t))'\), where \(X(t)\) defines the observable part and \(Z(t)\) the unobservable part of the system. Note that \(X(t)\) and \(Z(t)\) have dimensions \(d_1\) and \(d_2\) respectively such that \(Y(t)\) has dimension \(d = d_1 + d_2\). We assume further that the process \(X(t)\) is subject to measurement error such that we actually observe

\[
V(t) = X(t) + \epsilon(t),
\]

where \(\epsilon(t) \sim N(0, \Sigma)\) and \(\Sigma = \text{diag}\{\sigma_i^2\}\). Note that for unknown \(\Sigma\), we have \(\Theta = (\theta_1, \ldots, \theta_p, \sigma_1, \ldots, \sigma_d)'\). The process \(V_t\) will be observed at a finite number of times and the objective is to conduct inference for the (unknown) parameter vector \(\Theta\) on the basis of these noisy, partial and discrete observations.

In practice it is necessary to work with the discretized version of (8), given by the Euler approximation,

\[
\Delta Y(t) = \mu(Y(t), \Theta)\Delta t + \beta^2(Y(t), \Theta)\Delta W(t)
\]

where \(\Delta W(t)\) is a \(d\) dimensional iid \(N(0, I\Delta t)\) random vector. Now suppose we have measurements \(v(\tau_i)\) at evenly spaced times \(\tau_0, \tau_1, \ldots, \tau_T\) with intervals of length \(\Delta^t = \tau_{i+1} - \tau_i\). As \(\Delta^t\) is often too large to be used as a time step in (19), we put \(\Delta t = \Delta^t/m\) for some positive integer \(m > 1\). By choosing \(m\) to be sufficiently large, we can ensure that the discretization bias is arbitrarily small, but this also introduces the problem of \(m - 1\) missing values in between every pair of observations.

We deal with these missing values by dividing the entire time interval \([\tau_0, \tau_T]\) into \(mT + 1\) equidistant points \(\tau_0 = t_0 < t_1 < \ldots < t_n = \tau_T\) such that \(V(t)\) is observed at times \(t_0, t_m, \ldots, t_n\). Altogether we have \(d(nm + 1)\) missing values which we substitute with simulations \(Y(t_i)\). We refer to the collection of simulated data as the augmented data. Eraker (2001) denotes by \(\hat{Y}\) the \(d \times (n + 1)\) matrix obtained by stacking all elements of the augmented data, that is

\[
\hat{Y} = \begin{pmatrix}
X_1(t_0) & X_1(t_1) & \cdots & X_1(t_m) & X_1(t_{m+1}) & \cdots & X_1(t_n) \\
X_2(t_0) & X_2(t_1) & \cdots & X_2(t_m) & X_2(t_{m+1}) & \cdots & X_2(t_n)
\end{pmatrix}
\]

We now let \(Y^i = (X^i, Z^i)\) denote the \(i^{th}\) column of \(\hat{Y}\). If \(\pi(\Theta)\) and \(\pi(Y^0)\) are the prior densities of \(\Theta\) and \(Y^0\) respectively, then the joint posterior density for parameters and augmented data is given by

\[
\pi(\hat{Y}, \Theta | v_{\text{obs}}) \propto \pi(\Theta)\pi(Y^0)\left[\prod_{i=0}^{n-1} \pi(Y^{i+1} | Y^i, \Theta)\right]\left[\prod_{i \in \{0, m, \ldots, n\}} \pi(v^i | X^i, \Theta)\right],
\]

where \(v^i = Y^{i+1} - Y^i\).
interested in the online estimation of the unknown parameter vector, $\Theta$. 

By arriving at times $t_m$ sampled from a tractable approximation to $(\hat{Y}^t, \pi(\hat{Y}^t))$, the joint posterior, (20) is usually high dimensional, a Gibbs sampler is a particularly missing data given the observed data and the current state of the model parameters. As between simulation of parameters conditional on augmented data, and simulation of the density obtained from the Euler discretization.

As discussed in Tanner & Wong (1987), inference may proceed by alternating between simulation of parameters conditional on augmented data, and simulation of the missing data given the observed data and the current state of the model parameters. As each new observation arrives, our proposed simulation filter samples a new $\Theta^*$, $\hat{Y}^*$ in two stages: first $\Theta^*$ is sampled from a suitable proposal and then $\hat{Y}^*$ is sampled from a tractable approximation to $(\hat{Y}|\Theta^*, v_{\text{obs}})$. By simulating the latent data to be consistent with $\Theta^*$, the dependence between them is overcome (Golightly & Wilkinson 2003b). For further discussions on the use of MCMC methods for the Bayesian analysis of diffusions, see Roberts & Stramer (2001), Elerian et al. (2001) and Eraker (2001).

### 3.2 Simulation Filter

In the context of discrete time series with unobserved state variables, Bayesian sequential filtering has been discussed extensively, e.g., Berzuini, Best, Gilks & Larizza (1997), Pitt & Shephard (1999) and Doucet, Godsill & Andrieu (2000). Filtering for both parameters and state has been discussed by Liu & West (2001) among others.

We consider data $D_j = (v^0, v^m, \ldots, v^j)$, (where $j$ is an integer multiple of $m$) arriving at times $t_0, t_m, \ldots, t_j$ such that at time $t_{j+m}$, new data $v^{j+m}$ are accompanied by $m$ missing columns, $Y^{j+1}, \ldots, Y^{j+m}$. As each observation becomes available we are interested in the online estimation of the unknown parameter vector, $\Theta$.

We assume that we have an equally weighted sample of size $S$, ${\{\Theta(s), Y^{j} (s)\}, s = 1, \ldots, S}$ (with weights $w^{j} (s) = 1/S$), from the distribution $\pi(\Theta, Y^{j}|D_j)$, which we will denote by $\pi_j(\Theta, Y^{j})$. At time $t_{j+m}$, we observe $v^{j+m}$, which we will refer to as $v^M$ (putting $M = j + m$). Assimilation of the information contained in $v^M$ consists of generating a sample, ${\{\Theta(s), Y^{3M} (s)\}, s = 1, \ldots, S}$ from the posterior $\pi_M(\Theta, Y^M)$ which can be found by formulating the posterior for parameters and augmented data, then integrating out the latent data. Using (20) we have

$$
\pi_M(\Theta, Y^M) \propto \int_{\bar{Y}_M} \pi(\Theta) \pi(Y^0) \prod_{i=0}^{M-1} \pi(Y^{i+1}|Y^i, \Theta) \prod_{i \in \{0, m, \ldots, M\}} \pi(v^i|X^i, \Theta) \quad (23)
$$

where we define $\bar{Y}_M = (Y^0, Y^1, \ldots, Y^{M-1})$ and is simply the vector of latent values
up to time $t_M$. Hence our target is

$$\pi_M(\Theta, Y^M) \propto \pi_j(\Theta, Y^j)\pi(v^M|X^M, \Theta) \prod_{i=j}^{M-1} \pi(Y^{i+1}|Y^i, \Theta)$$

(24)

with $Y^1, \ldots, Y^{M-1}$ integrated out. We sample (24) by drawing $(Y^j, Y^{j+1}, \ldots, Y^M, \Theta)$, via MCMC, then discarding all components except $(\Theta, Y^M)$.

3.3 Filtering for Parameters and State

As $\pi_j(\Theta, Y^j)$ has no analytic form, we recursively approximate $\Theta, Y^j|D_j$ by the “particles” $\{(\Theta_{(s)}, Y^j_{(s)}), s = 1, \ldots, S\}$ with each $\Theta_{(s)}, Y^j_{(s)}$ having a discrete probability mass of $w^j_{(s)} = 1/S$. We assume that as $S \to \infty$, the particles approximate the filtering density, $\pi_j(\Theta, Y^j)$ increasingly well. The class of filters which treat the discrete support generated by the particles as the true (filtering) density are known as particle filters. Various implementations of particle filters have been proposed in the literature including sampling/importance resampling (Doucet et al. 2000) and MCMC (Pitt et al. 1999). Here we focus on an MCMC approach which we refer to as the simulation filter.

In the first step of our MCMC scheme, propose $(\Theta, Y^j)$ from $\pi_j(\Theta, Y^j)$ using the kernel density estimate of $\pi_j(\cdot, \cdot)$. First select an integer, $u$, uniformly from the set \{1, \ldots, S\} and then put

$$(\Theta_u, Y^j_u) \sim N\{(\Theta_{(u)}, Y^j_{(u)})\}, \omega^2B$$

(25)

where $B$ is the Monte Carlo posterior variance and the overall scale of the kernel is a function of the smoothing parameter, $\omega^2$ usually around 0.02. For large datasets however, Liu & West (2001) suggest that the random disturbances add up to give “information loss” over time (as the kernel density function is always over-dispersed relative to the posterior sample by a factor $1 + \omega^2$). To correct this, Liu & West (2001) employ a kernel shrinkage method by setting

$$(\Theta_u, Y^j_u) \sim N\{a(\Theta_{(u)}, Y^j_{(u)}), (1-a)(\Theta, Y^j), \omega^2B\}$$

(26)

where $a = 1 - \omega^2$, $\omega^2 = 1 - ((3\delta - 1)/2\delta)^2$, $\delta$ is a discount factor usually around 0.99 and $\Theta, Y^j$ is the Monte Carlo posterior mean of $\pi_j(\Theta, Y^j)$. For the data considered in Section 4 we found that using (25) works sufficiently well. See Liu & West (2001) and also West (1993) for further discussions on kernel smoothing.

Given $X^M_s \sim \pi(\cdot|v^M, \Theta_s)$, we are then tasked with simulating $Y^j_{s+1}, \ldots, Y^M_{s+1}, Z^M_s$ conditional on $\Theta_s, Y^j_s$ and $X^M_s$. However, obtaining the conditional density of missing values between two “observations” that are $m$ steps apart, under the non-linear structure of the diffusion process, is not trivial. We deal with this problem by adopting a “modified bridge” construct proposed by Durham & Gallant (2002). That is, treating $Y^j_s$ and $X^M_s$ fixed, we draw $Y^i_{s+1}$, for $i = j, \ldots, M - 2$, from a Gaussian approximation to $\pi(Y^i_{s+1}|Y^i_s, X^M_s, \Theta_s)$ which we denote the approximate density by $\tilde{\pi}(Y^i_{s+1}|Y^i_s, X^M_s, \Theta)$ (see Golightly & Wilkinson (2005b), Durham & Gallant (2002) and Ethier et al. (2001) for a review).

The final step in our MCMC scheme is to draw $Z^M_s \sim \pi(\cdot|Y^M_{s-1}, X^M_s, \Theta_s)$, that is, from the one step ahead Euler density further conditioned on $X^M_s$. Hence, if at some iteration, $s$, of our sampler we have current value, $\Phi_{(s)} = (Y^j_s, \ldots, Y^M_s, \Theta_s)$, then at
iteration $s+1$ we accept a move to $\Phi_s = (Y^j_s, \ldots, Y^M_s, \Theta_s)$ with probability $\min\{1, \alpha\}$, where

$$\alpha = \frac{\pi(Z^M | Y^{M-1}_s, X^M_s, \Theta) \prod_{i=j}^{M-2} \pi(Y^{i+1}_s | Y^i_s, X^M_s, \Theta) \prod_{i=j}^{M-1} \pi(Y^{i+1}_s | Y^i_s, \Theta_s)}{\pi(Z^M_s | Y^{M-1}_s, X^M_s, \Theta_s) \prod_{i=j}^{M-2} \pi(Y^{i+1}_s | Y^i_s, X^M_s, \Theta_s) \prod_{i=j}^{M-1} \pi(Y^{i+1}_s | Y^i_s, \Theta_s)}, \quad (27)$$

and store $(\Theta_{s+1}, Y^M_{s+1})$, ready for the next time point. The Markov chain generated in this way has the posterior distribution of interest, $\pi(\Theta, Y^j)$, as its invariant distribution. The simulation filter then has the following algorithmic form:

1. Set $j = 0$. For $s = 1, \ldots, S$ draw $\Theta_{(s)} \sim \pi(\Theta)$, $X^0_{(s)} \sim \pi(X^0 | v^0, \Theta_{(s)})$ and $Z^0_{(s)} \sim \pi(Z^0)$.
2. Set $M = j + m$. For $s = 1, \ldots, S$,
   - Propose $(\Theta_s, Y^j_s)$ using (25).
   - Draw $X^M_s \sim \pi(\cdot | v^M_s, \Theta_s)$.
   - For $i = j, \ldots, M - 2$ simulate $Y^{i+1}_s \sim \pi(\cdot | Y^i_s, X^M_s, \Theta_s)$.
   - Draw $Z^M_s \sim \pi(\cdot | Y^{M-1}_s, X^M_s, \Theta_s)$.
   - Set $\Phi_s = (Y^j_s, \ldots, Y^M_s, \Theta_s)$ and put $\Phi_{(s+1)} = \Phi_s$ with probability $\min\{1, \alpha\}$ (where $\alpha$ is given by (27)) else put $\Phi_{(s+1)} = \Phi_{(s)}$.
   - Store $(\Theta_{(s)}, Y^M_{(s)})$.
3. Set $j = j + m$.
4. Return to step 2.

Thus step 2 performs the update for a given time point. As with any MCMC sampler, this scheme can be modified by allowing a number of iterations to be discarded as “burn-in”. A further $S$ iterations may then be performed to generate the desired sample, $\{(\Theta_{(s)}, Y^M_{(s)}), s = 1, \ldots, S\}$ from $\pi_M(\Theta, Y^M)$. Further modifications may be made by thinning the MCMC output at the expense of running the sampler for longer. This is done separately for each time point, with our final posterior sample used as the prior for the next time point.

### 4 Simulation Study: Prokaryotic Auto-regulatory Gene Network

To illustrate the methodology of Section 3.2, the simulation filter is applied to the diffusion approximation of the regulatory gene network characterised by the reactions (11)–(13).

As well as exploring the fully observed case, we report results for several data-poor contexts; for example, only measuring protein and RNA levels leads to a model with observable part $X(t) = (l(t), G(t), r_l(t), r_g(t))^T$ and unobservable part of the reduced system, $Z(t) = (i(t), g(t))^T$. Note that formulating the partially observed model in this way implies that we only know the conservation constants, $K_1$ and $K_2$ (see (16) and (17)), and not the split into $l$ and $i$ or $l$ and $g$. In practice it is reasonable to observe $K_1$ and $K_2$ as they correspond to the number of copies of each gene on the genome. In Section 4.2 we assume that both $K_1$ and $K_2$ are known but we do not observe $l$ and $i(t)$ and $i(t)$ or $l$ and $g(t)$ and $g(t)$ at any time $t$.

Realistically, we may have two (or more) independent experimental datasets, one consisting of measurements only on the proteins, $l(t), G(t)$ and another on RNA levels,
Here, we take advantage of the sequential nature of the simulation filter, running the algorithm for each dataset in turn and using the posterior sample obtained from the first dataset as the prior sample for the second.

### 4.1 Results: Fully Observed Model

We first implement the MCMC scheme given in Section 3.3 for the fully observed case; we assume that we observe \( Y(t) = (l(t), G(t), i(t), g(t), r_i(t), r_g(t))' \) at all times \( t \).

We consider equispaced data, \( D_1 \), consisting of 30 observations on \([0, 29]\), simulated exactly using the Gillespie algorithm (see Figure 1). Each data point is subjected to measurement error by adding a Gaussian random variable with zero mean and variance \( \sigma^2 = 3 \) (so that \( \Sigma = \sigma^2 I \) in (18)), which we assume to be unknown. True values for \((c_1, \ldots, c_{12})\) are chosen to be 0.08, 0.82, 0.09, 0.25, 0.1, 0.35, 0.3, 0.1, 0.12, 0.1 and we place Uniform \( U(-5, 1) \) priors on each \( \log(c_i) \), for \( i = 1, \ldots, 12 \) and \( \sigma \). Note that \( K_1 \) and \( K_2 \) (the number of copies of each gene) are set to be 10.

The simulation filter is run for 5 million iterations with a thin of 250, giving a final sample of size \( S = 20,000 \). Discretization is set by running the MCMC algorithm with \( m = 5, 8, 16, 20 \). Figure 2 and Table 1 summarise the posterior distributions; trace, density and autocorrelation plots can be seen in Figure 2 for a selection of parameters with \( m = 20 \). Table 1 reports posterior means and standard deviations for \( \Theta \), based on the output from the simulation filter for each choice of \( m \).

As the estimated MCMC error is related to the autocorrelations within the chains, the relative performance of the simulation filter can be assessed by studying the sample autocorrelation functions for each parameter. Figure 2 shows that autocorrelations die down very quickly despite large \( m \). We also see that the sampler produces estimates close to the true values that generated the data.

Inspection of Table 1 reveals the advantage of including latent variables in the estimation framework. For large \( m \) there is a notable decrease in discretization error; for example \( c_{10} \) (the stochastic rate constant for mRNA degradation) has a true value of 0.1 while it was estimated to be 0.031 with \( m = 5 \) and 0.084 with \( m = 20 \). Similarly the standard deviation of the measurement error, \( \sigma \) was estimated to be 1.846 with \( m = 5 \) and 1.547 with \( m = 20 \) and has a true value of 1.414. Note also that although estimates of \( c_1, c_2, c_3 \) and \( c_4 \) appear imprecise (perhaps due to the small number of observations), estimates of \( c_1/c_2 \) and \( c_3/c_4 \) (corresponding to the propensities of reactions \( R_1 \) and \( R_3 \) respectively) are fairly accurate for all choices of \( m \).

### 4.2 Results: Partially Observed Model

We now apply the MCMC algorithm to the partially observed model. We consider three equispaced datasets, \( D_2, D_3 \) and \( D_4 \), each independently simulated using the Gillespie algorithm with stochastic rate constants, \( c_1, \ldots, c_{12} \) as in Section 4.1. \( D_2 \) consists of 30 observations on \( X(t) = (l(t), G(t), r_i(t), r_g(t))' \) with each data point subject to measurement error with variance \( \sigma^2 = 3.0 \). \( D_3 \) contains 40 observations on protein levels only; \( X(t) = (l(t), G(t))' \) and the variance of the measurement error is \( \sigma^2 = 3.0 \).
Finally, \( D_4 \) contains 20 observations on RNA levels; \( X(t) = (r_{i}(t), r_{g}(t))' \) with \( \sigma^2 = 2.0 \). For each dataset, we assume that the variance of the measurement error is known and that the number of copies of each gene is known to be \( K_1 = K_2 = 10 \). As in Section 4.1, we place uniform priors on each \( \log(c_i) \) and also on \( \log(Z^0) \).

The simulation filter is run for each dataset for 4 million iterations with a thin of 200, giving a final sample of size \( S = 20,000 \). Discretization is set by running the algorithm with \( m = 20 \). Table 2 summarises the posterior distribution for each dataset. In addition we provide summaries of \( \pi(\Theta|D_3, D_4) \), obtained by using the posterior sample from \( \pi(\Theta|D_3) \) as the prior sample for data \( D_4 \). Figure 3 shows posterior densities of \( c_5, c_7, c_{11} \) and \( c_{12} \) given data \( D_3, D_4 \) and \( (D_3, D_4) \).

Intuitively, when observing just two species, estimates are generally more accurate for rate constants governing reactions involving those species. For example, \( c_9 \) and \( c_{10} \) (pertaining to RNA degradation reactions given by (13)), both with true values of \( 0.1 \), are estimated to be 0.148 and 0.201 respectively when using 40 observations on protein levels \( (D_3) \). However, when using just 20 observations on RNA levels \( (D_4) \), we see an increase in accuracy with estimates of 0.099 and 0.097 respectively. Similarly, \( c_1/c_2 \) and \( c_3/c_4 \), the propensities of repression reactions \( R_1 \) and \( R_2 \) have true values of 0.096 and 0.1. Using the protein data \( (D_3) \) we see estimates of 0.241 and 0.290 respectively, whilst using just RNA levels, estimates are very poor (0.414 and 0.507).

Running our algorithm for the RNA data with a prior, \( \pi(\Theta) \) given by the posterior sample for parameters given protein data, \( \pi(\Theta|D_3) \) yields, in general, fairly precise estimates of those parameters governing RNA reactions and also of parameters governing protein reactions. For example, \( c_1/c_2 \) and \( c_3/c_4 \) are now estimated to be 0.227 and 0.345 whilst posterior means for \( c_9 \) and \( c_{10} \) are 0.091 and 0.112 respectively. As we would expect, for those parameters governing protein-only reactions, \( \pi(\Theta|D_3, D_4) \) is dominated by the prior, \( \pi(\Theta|D_3) \), as this is in fact the posterior obtained when only using protein data, \( D_3 \) (see Figure 3).

Finally, it appears that we gain the most information by observing as many species as possible in a single experiment rather than combining datasets on a few species obtained from multiple independent experiments. This can be seen by comparing the columns in Table 2 corresponding to \( D_2 \) (30 observations on RNA and protein levels) and \( (D_3, D_4) \).

5 Discussion

We have implemented a sequential Bayesian approach to conduct rigorous inference for rate constants governing biochemical reactions. By adopting a diffusion approximation, the solution to the problem of reverse engineering rate constants from noisy time course data corresponds to the estimation of nonlinear, discretely (and perhaps partially) observed stochastic differential equations. However, the task of inferring parameters in SDEs is not trivial. The estimation framework necessarily introduces missing values and high dependence between these latent values and parameters results in poor mixing properties of MCMC schemes such as the Gibbs sampler (Roberts & Stramer 2001). The utility of the simulation filter is two-fold; firstly, by performing a joint update of the parameters and missing values at each time point, we can overcome the dependence...
between them (Golightly & Wilkinson 2005b). Also, as the method is sequential, we can use a posterior sample obtained from one dataset as the prior for the next, allowing us to handle multiple datasets from different experiments.

The methodology was applied to synthetic data generated from a prokaryotic auto regulatory gene network model. Naturally, the integration of actual measurements into the modelling framework remains of great interest and is the subject of ongoing research.

References


Figure 1: $D_1$: 30 (noisy) observations on $(l(t), g(t), i(t), g(t), r_i(t), r_g(t))$. 
Figure 2: Trace, density and autocorrelation plots for $c_6$, $c_7$, $c_{10}$, $c_{11}$, $c_{12}$ and $\sigma$ for the fully observed model using 30 observations and $m = 20$. Results are based on a final sample of size 20,000, thinned from 5,000,000 MCMC iterations.
Figure 3: Posterior density plots for $c_5$, $c_7$, $c_{11}$ and $c_{12}$ obtained from $D_3$ (dotted line), $D_4$ (dashed line) and $(D_3, D_4)$ (solid line). Results are based on a final sample of size 20,000, thinned from 4,000,000 MCMC iterations.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>True Value</th>
<th>Mean (Standard Deviation)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$m = 5$</td>
</tr>
<tr>
<td>$c_1$</td>
<td>0.08</td>
<td>0.010 (0.003)</td>
</tr>
<tr>
<td>$c_2$</td>
<td>0.82</td>
<td>0.112 (0.093)</td>
</tr>
<tr>
<td>$c_1/c_2$</td>
<td>0.096</td>
<td>0.119 (0.115)</td>
</tr>
<tr>
<td>$c_3$</td>
<td>0.09</td>
<td>0.010 (0.002)</td>
</tr>
<tr>
<td>$c_4$</td>
<td>0.9</td>
<td>0.070 (0.050)</td>
</tr>
<tr>
<td>$c_3/c_4$</td>
<td>0.1</td>
<td>0.115 (0.115)</td>
</tr>
<tr>
<td>$c_5$</td>
<td>0.25</td>
<td>0.341 (0.208)</td>
</tr>
<tr>
<td>$c_6$</td>
<td>0.1</td>
<td>0.302 (0.168)</td>
</tr>
<tr>
<td>$c_7$</td>
<td>0.35</td>
<td>0.121 (0.071)</td>
</tr>
<tr>
<td>$c_8$</td>
<td>0.3</td>
<td>0.056 (0.031)</td>
</tr>
<tr>
<td>$c_9$</td>
<td>0.1</td>
<td>0.048 (0.037)</td>
</tr>
<tr>
<td>$c_{10}$</td>
<td>0.1</td>
<td>0.031 (0.023)</td>
</tr>
<tr>
<td>$c_{11}$</td>
<td>0.12</td>
<td>0.369 (0.238)</td>
</tr>
<tr>
<td>$c_{12}$</td>
<td>0.1</td>
<td>0.023 (0.015)</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>1.414</td>
<td>1.846 (0.205)</td>
</tr>
</tbody>
</table>

Table 1: Posterior means and standard deviations for $\Theta$ estimated on 30 observations ($D_1$) from the fully observed model. Estimation results are based on a final sample of size 20,000, thinned from 5,000,000 MCMC iterations.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>True Value</th>
<th>Mean (Standard Deviation)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( D_2 )</td>
<td>( D_3 )</td>
</tr>
<tr>
<td>(c_1)</td>
<td>0.08</td>
<td>0.024 (0.019)</td>
</tr>
<tr>
<td>(c_2)</td>
<td>0.82</td>
<td>0.254 (0.231)</td>
</tr>
<tr>
<td>(c_{1}/c_2)</td>
<td>0.096</td>
<td>0.176 (0.224)</td>
</tr>
<tr>
<td>(c_3)</td>
<td>0.09</td>
<td>0.031 (0.022)</td>
</tr>
<tr>
<td>(c_4)</td>
<td>0.9</td>
<td>0.214 (0.244)</td>
</tr>
<tr>
<td>(c_{3}/c_4)</td>
<td>0.1</td>
<td>0.204 (0.319)</td>
</tr>
<tr>
<td>(c_5)</td>
<td>0.25</td>
<td>0.418 (0.296)</td>
</tr>
<tr>
<td>(c_6)</td>
<td>0.1</td>
<td>0.072 (0.066)</td>
</tr>
<tr>
<td>(c_7)</td>
<td>0.35</td>
<td>0.228 (0.159)</td>
</tr>
<tr>
<td>(c_8)</td>
<td>0.3</td>
<td>0.275 (0.100)</td>
</tr>
<tr>
<td>(c_9)</td>
<td>0.1</td>
<td>0.133 (0.081)</td>
</tr>
<tr>
<td>(c_{10})</td>
<td>0.1</td>
<td>0.046 (0.037)</td>
</tr>
<tr>
<td>(c_{11})</td>
<td>0.12</td>
<td>0.076 (0.083)</td>
</tr>
<tr>
<td>(c_{12})</td>
<td>0.1</td>
<td>0.103 (0.041)</td>
</tr>
</tbody>
</table>

Table 2: Posterior means and standard deviations for parameters estimated using datasets \(D_2, D_3\) and \(D_4\) from the partially observed model. Discretization is set at \(m = 20\) and the estimation results are based on a final sample of size 20,000, thinned from 4,000,000 MCMC iterations.