An exact Gibbs Sampler for the Markov Modulated Poisson Process

Paul Fearnhead[1] and Chris Sherlock[1,2]

1. Department of Mathematics and Statistics, Lancaster University, Lancaster, LA1 4YF, UK

2. Correspondence should be addressed to Chris Sherlock.
(e-mail: c.sherlock@lancs.ac.uk).

Summary: A Markov modulated Poisson process (MMPP) is a Poisson process whose intensity varies according to a Markov process. We present a novel technique for simulating from the exact distribution of a continuous time Markov chain over an interval given the start and end states and the infinitesimal generator, and use this to create a Gibbs sampler which samples from the exact distribution of the hidden Markov chain in an MMPP. We apply the Gibbs sampler to modelling the occurrence of a rare DNA motif (the Chi site) and to inferring regions of the genome with evidence of high or low intensities for occurrences of this site.

Keywords: *Forward-Backward Algorithm, Genome Segmentation, Gibbs Sampler*

1 Introduction

A Markov Modulated Poisson Process (MMPP) is a Poisson process whose intensity depends on the current state of an independently evolving continuous time Markov chain. Points from the MMPP are often referred to as *the observed data* and the underlying Markov chain as *the hidden data*.

MMPP's are used in modelling a variety of phenomena. For example, the arrivals of photons from single molecule fluorescence experiments (Burzykowski *et al.*, 2003; Kou *et al.*, 2005), where the arrival rate of photons at a receptor is modulated by the state of a molecule which (in the simplest model formulation) alternates between its ground state and an excited state. Other examples include, wet deposition of a radionuclide emitted from a point source (Davison and Ramesh, 1996); frequency of bank transactions (Scott, 1999); requests for web pages from users of the World Wide Web (Scott and Smyth, 2003); modelling overflow in telecommunications networks; and modelling packetised voice and data streams (Fischer and Meier-Hellstern, 1992). Later in this paper we use MMPPs to model the occurence of a rare DNA motif in bacterial genomes.

We focus on inference of both the parameters and hidden state of MMPPs. The type of data available varies from application to application. In some applications the exact timings of all observed events are known and in others data are accumulated over fixed intervals. In the latter situation the observed data often appear as either a count of the number of events in each interval or a binary indication for each interval as to whether there were no events or at least one event.

MMPP parameters can be fitted to data by matching certain theoretical moments to those observed (see Fischer and Meier-Hellstern (1992) and references therein). However, it is possible to calculate the likelihood of arrival data for an MMPP (for example Asmussen (2000); see also Section 4). Ryden (1996)

summarises several likelihood approaches.

Here we consider Bayesian analysis and focus on exploring the posterior distribution via Markov chain Monte Carlo. Metropolis-Hastings algorithms (e.g. Gilks *et al.*, 1996) provide a standard mechanism for Bayesian inference about parameters when the likelihood is computable. This approach is employed for example by Kou *et al.* (2005). Alternatively an approximate Gibbs sampler has been developed in Scott (1999) and Scott and Smyth (2003). This Gibbs sampler is only applicable to event-time data, and restricts the possible transitions of the underlying Markov chain. The approximation is based on requiring that certain transitions of the underlying chain can only occur at event-times. However for the examples considered in Scott (1999) and Scott and Smyth (2003) this approximate Gibbs sampler is very efficient.

Here we present an exact Gibbs sampler which, conditional on the data, samples alternately from the *true* conditional distribution of the hidden chain given the parameters and then the conditional distribution of the parameters given the hidden chain. This Gibbs sampler can analyse data in any of the three forms (exact timings, and interval counts or binary indicators) outlined earlier in this section, and can allow for a general transition matrix for the state of the hidden the Markov chain. It also avoids the simplifying approximation used by Scott (1999). An advantage of the Gibbs sampler over a Metropolis-Hastings scheme is that the Gibbs sampler does not need to be tuned. The Gibbs sampler also allows directly for inference about the hidden states through approximate samples from their posterior distribution. This feature is important for the application in genomics that we consider.

The main novelty in the Gibbs sampler is a direct simulation algorithm for the conditional distribution of the complete continuous time path of the hidden state. This is an extension of the forward-backward algorithm of Baum *et al.* (1970) to

continuous time, and an extension of the ideas of Fearnhead and Meligkotsidou (2004). It can be applied to general continuous time Markov processes and provides an alternative to the rejection sampling algorithms of Blackwell (2003) and Bladt and Sorensen (2005). We describe the forward-backward algorithm in Section 2 and present its extension to continuous time Markov process in Section 3. We then focus on MMPPs, reviewing the derivation of the likelihood in Section 4, and present the Gibbs sampler in 5. We then the Gibbs sampler to analyse data of the occurence of a DNA motif, known as the Chi sites, in *E. coli* in Section 6 and the paper concludes with a discussion.

2 The forward-backward algorithm

The forward-backward algorithm (Baum *et al.*, 1970) applies to any discretely observed Hidden Markov Model (HMM) and allows sampling of the state of the hidden chain at the observation times given the states at the start and end of the observation window. The algorithm is easily extended to the case where there is a prior distribution on the initial state and no knowledge of the end state of the chain.

We first describe a general HMM. Let an unobserved (discrete or continuous time) Markov chain evolve over a state space of cardinality d. We observe a second process over a window $[0, t_{obs}]$ at specific times $t'_1, \ldots, t'_n$. Suppose that the value of the observed process at time t'_k is d_k, and define $\mathbf{d} := (d_1, \ldots, d_n)^t$. For notational convenience define $t'_0 = 0$, $t'_{n+1} = t_{obs}$ and $\mathbf{t}' = (t'_0, \ldots, t'_{n+1})$. Also write s_k for the state of the unobserved Markov chain at time t'_k. The likelihood of the observed process depends on the state of the hidden process via a likelihood vector $\mathbf{l}^{(k)}$ with $k = 1, \ldots, n$ where $l_i^{(k)} := P(d_k | S_k = i)$. From this define a likelihood matrix $\mathbf{L}^{(k)} = \text{diag}(\mathbf{l}^{(k)})$.

Let $\mathbf{T}^{(k)}$ be the k^{th} transition matrix of the Markov chain (i.e. $T_{ij}^{(k)}$ is the probability that the unobserved process is in state j just before t'_k given that it is in state i at t'_{k-1}.)

We define probability matrices

$$\begin{aligned} A^{(n+1)}_{s,s_{n+1}} &= P(s_{n+1}|s_n = s) \\ A^{(k)}_{s,s_{n+1}} &= P(d_k, \ldots, d_n, s_{n+1}|s_{k-1} = s) \quad (0 < k \leq n) \end{aligned}$$

And note that

$$P(d_k, \ldots, d_n, s_{n+1}|s_{k-1}) = \sum_{s_k=1}^{d} P(s_k|s_{k-1})P(d_k|s_k)P(d_{k+1}, \ldots, d_n, s_{n+1}|s_k)$$

Therefore the matrices may be calculated via a backwards recursion

$$\begin{aligned} \mathbf{A}^{(n+1)} &= \mathbf{T}^{(n+1)} \\ \mathbf{A}^{(k)} &= \mathbf{T}^{(k)}\mathbf{L}^{(k)}\mathbf{A}^{(k+1)} \quad (0 < k \leq n) \end{aligned}$$

These matrices accumulate information about the chain through the data. The final accumulation step creates $\mathbf{A}^{(0)}$, where $A^{(0)}_{s_0,s_{n+1}} = P(\mathbf{d}, s_{n+1}|s_0)$ is proportional to the likelihood of the observed data given the start and end states.

Using the Markov property we therefore have

$$\begin{aligned} P(S_k = s|\ \mathbf{d}, s_{k-1}, s_{n+1}) &= P(S_k = s|\ d_k, \ldots, d_n, s_{k-1}, s_{n+1}) \\ &= \frac{T^{(k)}_{s_{k-1},s}\ l^{(k)}_s A^{(k+1)}_{s,s_{n+1}}}{A^{(k)}_{s_{k-1},s_{n+1}}} \end{aligned} \tag{1}$$

Using (1) we may proceed forwards through the observation times $t'_1, \ldots t'_n$, simulating the state at each observation point in turn. This algorithm is often presented in the equivalent formulation of a forwards accumulation of information and a backwards simulation step through the observation times.

If the start and end states of the chain are unknown, but a prior distribution $\boldsymbol{\mu}$ on the state of the hidden process is provided, then with a slight adjustment

to the algorithm we may simulate the states at the start and end times of the chain as well as at the observation times.

The start state is simulated from

$$P(S_0 = s|\mathbf{d}) = \frac{\mu_s \ [\mathbf{A}^{(1)}\mathbf{1}]_s}{\boldsymbol{\mu}^t \ \mathbf{A}^{(1)}\mathbf{1}} \tag{2}$$

where $\mathbf{1}$ is the d-dimensional vector of ones.

The state s_k $(k > 0)$ is then simulated from

$$P(S_k = s|\mathbf{d}, s_{k-1}) = \frac{T^{(k)}_{s_{k-1},s} \ l^{(k)}_s \ [\mathbf{A}^{(k+1)}\mathbf{1}]_s}{[\mathbf{A}^{(k)}\mathbf{1}]_{s_{k-1}}} \tag{3}$$

The observation times in a Markov Modulated Poisson Process correspond to actual events from the observed Poisson process. Therefore not only do the observations contain information about the state of the hidden chain, but so do the intervals between observations, since these contain no events. In Section 4 we derive likelihoods through accumulation steps modified to take this into account. In a similar way we can use the forward-backward algorithm to simulate the state of the hidden chain at observation times for the first stage of our Gibbs sampler. The second stage of the Gibbs sampler simulates a realisation from the exact distribution of the full underlying Markov chain conditional on the data. This is more challenging and relies on a technique for simulating a realisation from a continuous time Markov chain over an interval given the start and end states.

3 Simulating a continuous time Markov chain over an interval given the start and end states

Let continuous time Markov chain W_t have generator matrix $\mathbf{G}$, and let it start the interval $[0, t]$ in state s_0 and finish in state s_t. We describe a method for simulating from the exact conditional distribution of the chain given the start and end states.

The behaviour of W_t on entering state i until leaving that state can be thought of in terms of a Poisson process of rate $\rho_i := -g_{ii}$ and a set of transition probabilities

$$\begin{aligned} p_{ij} &= g_{ij}/\rho_i \quad (i \neq j) \\ &= 0 \qquad\quad (i = j) \end{aligned}$$

The Poisson process is started as soon as the chain enters state i; at the first event from the process the chain changes to a state j determined at random using the transition probabilities for state i. A new Poisson process is then initiated with intensity corresponding to the new state.

An alternative formulation is based on the idea of uniformisation (e.g. Ross, 1996). We apply a single dominating Poisson process U_t to determine when transitions may occur; crucially the intensity ρ of the Poisson process is independent of the chain state. We call the events in this dominating Poisson process "U-events". Probabilities for the state transitions at these U-events are defined in terms of a transition matrix $\mathbf{M}$.

The intensity of the dominating process must necessarily be at least as large as the largest (in modulus) diagonal element of $\mathbf{G}$. With $\rho = \max \rho_i$ the transition matrix for the discrete time sequence of states at U-events is

$$\mathbf{M} := \frac{1}{\rho}\mathbf{G} + \mathbf{I}.$$

For any state i with $\rho_i < \rho$, $\mathbf{M}$ specifies a non-zero probability of no change in the state, so that the rate of events that change state is ρ_i. Considering an interval of length t straightforward expansion of the transition matrix for the interval gives

$$e^{\mathbf{G}t} = e^{-\rho \mathbf{I} t} e^{\rho \mathbf{M} t} = \sum_{r=0}^{\infty} e^{-\rho t} \frac{(\rho t)^r}{r!} \mathbf{M}^r \tag{4}$$

The $(i,j)^{th}$ element of the left hand side is $P(W_t = j | W_0 = i)$. If we define $N_U(t)$ as the number of U-events over the interval of length t then the $(i,j)^{th}$

element on right hand side can be interpreted as

$$\sum_{r=0}^{\infty} P(N_U(t) = r)\ P(W_t = j | W_0 = i, N_U(t) = r)$$

Thus conditional on start and end states s_0 and s_t, the distribution of the number of dominating U-events is given by

$$P(N_U(t) = r) = \frac{e^{-\rho t} \frac{(\rho t)^r}{r!} [\mathbf{M}^r]_{s_0, s_t}}{[e^{\mathbf{G}t}]_{s_0, s_t}} \tag{5}$$

We have used a single dominating Poisson process with fixed intensity independent of the chain state. Therefore conditional on the number of dominating events, the positions of these events and the state changes that occur at the events are independent of each other and may be simulated separately. Furthermore, since U_t is a simple Poisson process the U events are distributed uniformly over the interval $[0, t]$.

Suppose that r dominating U events are simulated at times $t_1^*, \ldots, t_r^*$, and let these correspond to (possible) changes of state of W_t to $s_1^*, \ldots, s_r^*$. For convenience we define $t_0^* := 0$ and $s_0^* := s_0$.

Now

$$P(W_t = s_t | W_0 = s_0) = [\mathbf{M}^r]_{s_0, s_t}$$

The start and end state for each interval are assumed known, and so we employ the forward-backward algorithm of Section 2 with $\mathbf{L}^{(k)} = \mathbf{I}$ and $\mathbf{T}^{(k)} = \mathbf{M}$ to simulate the state change at each U event

$$P(W_{t_j^*} = s | W_{t_{j-1}^*} = s_{j-1}^*, W_t = s_t) = \frac{[\mathbf{M}]_{s_{j-1}^*, s} [\mathbf{M}^{r-j}]_{s, s_t}}{[\mathbf{M}^{r-j+1}]_{s_{j-1}^*, s_t}} \ (j = 1, \ldots, r) \tag{6}$$

Our algorithm then becomes

(i) Simulate the number of dominating events using (5).

(ii) Simulate the position of each dominating event from a uniform distribution over the interval [0,t].

(iii) Simulate the state changes at the dominating events using (6).

4 Likelihood for MMPP's

We now focus exclusively on MMPP's. Let a (hidden) continuous-time Markov chain X_t on state space $\{1, \ldots, d\}$ have generator matrix $\mathbf{Q}$ and stationary distribution $\boldsymbol{\nu}$.

An MMPP is a Poisson process Y_t whose intensity is λ_i when $X_t = i$, but in all other ways is evolving independently of X_t. We write $\boldsymbol{\lambda} := (\lambda_1, \ldots, \lambda_d)^t$ and $\boldsymbol{\Lambda} := \text{diag}(\boldsymbol{\lambda})$.

We are interested in Bayesian inference about $\boldsymbol{\lambda}, \mathbf{Q}$, and X_t. Here we review derivations of likelihood for the three different data types mentioned in the introduction. Likelihoods are required for inference about $\boldsymbol{\lambda}$ and $\mathbf{Q}$ using Metropolis-Hastings schemes (see Section 5). The accumulation steps and extended state spaces used here are essential also for our Gibbs sampler which allows inference for $\boldsymbol{\lambda}, \mathbf{Q}$, and X_t and is detailed in section 5.2.

The Y process is (fully or partially) observed over an interval $[0, t_{obs}]$ with t_{obs} known and fixed in advance. We employ the symbol $\mathbf{1}$ for the matrix or (horizontal or vertical) vector all of whose elements are one, and similarly $\mathbf{0}$ is a matrix or vector all of whose elements are zero.

4.1 Derivation of the likelihood function

We are interested in inference for three commonly encountered data formats

D1 Exact times are recorded for each of the n observed events (see Kou *et al.* (2005), and Scott and Smyth (2003) for example uses of this data format).

D2 A fixed series of $n+1$ contiguous accumulation intervals of length t_i is used,

and associated with the i^{th} interval is a binary indicator b_i which is zero if there are no Y-events over the interval and one otherwise (see for example Davison and Ramesh, 1996).

D3 A fixed series of $n+1$ contiguous accumulation intervals of length t_i is used, and associated with the i^{th} interval is a count c_i of the number of Y-events over the interval (see for example Burzykowski *et al.*, 2003).

In each case it is possible to derive the likelihood function. We summarise the three derivations; for more details see Asmussen (2000), Davison and Ramesh (1996), and Burzykowski *et al.* (2003) respectively.

4.1.1 Likelihood for event-time data

We first consider the data format D1. We write $N_Y(t)$ for the number of Y-events in the interval $[0, t]$, so that $N_Y(0) = 0$ and $N_Y(t_{obs}) = n$, the total number of events. For notational convenience we set $t'_0 = 0$, $t'_{n+1} = t_{obs}$ and let $t'_1, \ldots, t'_n$ be the event times for the n events. Define $t_k = t'_k - t'_{k-1}, k = 1, \ldots, n+1$; these are respectively the time from the start of the observation period to the first event, the inter-event times, and the time from the last event until the end of the observation period. We define $\mathbf{t} := (t_1, \ldots, t_{n+1})^t$.

We first derive a form for

$$P_{ij}^{(0)}(t) := P(\text{there are no } Y \text{ events in } (0, t) \text{ and } X_t = j \mid X_0 = i)$$

We define a meta-Markov process W_t on an extended state space $\{1, \ldots, d, 1^*\}$, and let W_t combine X_t and Y_t as follows: W_t matches X_t exactly up until just before the first Y event. At the first such event W moves to the absorbing state 1^*. So if the first Y-event occurs at time t'

$$\begin{aligned} \text{for } t < t',\ W_t &= X_t \\ \text{for } t \geq t',\ W_t &= 1^* \end{aligned}$$

The generator matrix for W_t is

$$\mathbf{G_w} = \begin{bmatrix} \mathbf{Q} - \mathbf{\Lambda} & \boldsymbol{\lambda} \\ \mathbf{0} & 0 \end{bmatrix} \tag{7}$$

So the transition matrix at time t is

$$e^{\mathbf{G_w} t} = \begin{bmatrix} e^{(\mathbf{Q}-\mathbf{\Lambda})t} & (\mathbf{Q} - \mathbf{\Lambda})^{-1}(e^{(\mathbf{Q}-\mathbf{\Lambda})t} - \mathbf{I})\boldsymbol{\lambda} \\ \mathbf{0} & 1 \end{bmatrix} \tag{8}$$

From the definition of W we see that

$$P_{ij}^{(0)}(t) = [e^{(\mathbf{Q}-\mathbf{\Lambda})t}]_{ij} \tag{9}$$

So the likelihood of the observed data, and that the chain ends in state j given that it starts in state i is the $(i,j)^{th}$ element of

$$e^{(\mathbf{Q}-\mathbf{\Lambda})t_1}\mathbf{\Lambda}e^{(\mathbf{Q}-\mathbf{\Lambda})t_2}\mathbf{\Lambda}\ldots\mathbf{\Lambda}e^{(\mathbf{Q}-\mathbf{\Lambda})t_{n+1}}$$

This is the $\mathbf{A}^{(0)}$ matrix of the forward-backward algorithm as described in Section 2. Assuming that the chain starts in its stationary distribution, the likelihood of the observed data is therefore

$$L(\mathbf{Q}, \mathbf{\Lambda}, \mathbf{t}) = \boldsymbol{\nu}^t e^{(\mathbf{Q}-\mathbf{\Lambda})t_1}\mathbf{\Lambda}\ldots e^{(\mathbf{Q}-\mathbf{\Lambda})t_n}\mathbf{\Lambda}e^{(\mathbf{Q}-\mathbf{\Lambda})t_{n+1}}\mathbf{1} \tag{10}$$

4.1.2 Likelihood for accumulation interval formats

We now consider data formats D2 and D3 and for simplicity assume all the interval lengths to be equal ($t_i = t^*$, $i = 1, \ldots, n+1$). Extension to the more general case is straightforward.

Define

$$P_{ij}^{(s)} = P(\text{there are } s \; Y\text{-events over } (0, t^*) \text{ and } X_{t^*} = j | X_0 = i)]$$

and

$$\overline{P}_{ij} = P(\text{there is at least one } Y\text{-event over } (0, t^*) \text{ and } X_{t^*} = j | X_0 = i)]$$

With b_i as the binary indicator for at least one event in the i^{th} interval, the likelihood for D2 is therefore

$$\boldsymbol{\nu}^t \left(\prod_{i=1}^{n+1} \mathbf{P}^{(0)^{1-b_i}} \, \overline{\mathbf{P}}^{\, b_i} \right) \mathbf{1}$$

and with count c_i of the number of events for each interval the likelihood for D3 is

$$\boldsymbol{\nu}^t \left(\prod_{i=1}^{n+1} \mathbf{P}^{(c_i)} \right) \mathbf{1}$$

$\mathbf{P}^{(0)}$ is given by (9) and so it remains to calculate the matrices $\mathbf{P}^{(c)}$ $(c > 0)$, and $\overline{\mathbf{P}}$.

Since the probability of finishing interval $(0, t^*)$ in state j given starting state i is the $(i, j)^{th}$ element of $e^{\mathbf{Q}t^*}$, we see that

$$\overline{\mathbf{P}} = e^{\mathbf{Q}t^*} - e^{(\mathbf{Q}-\boldsymbol{\Lambda})t^*}$$

For format D3 define $c_{max} = \max \; c_i$ and create a new meta-process V_t on state space $S = (1^{(0)}, \ldots, d^{(0)}, 1^{(1)}, \ldots d^{(1)}, \ldots, 1^{(c_{max})}, \ldots, d^{(c_{max})}, 1^*)$. If the number of Y-events observed up until time t in the accumulation interval containing t is $N_Y^*(t)$, then for $N_Y^*(t) \leq c_{max}$ $V_t = X_t^{(N_Y^*(t))}$ and otherwise $V_t = 1^*$. For example if at time t, the hidden process is in state 3 and there have been 7 events so far in the accumulation interval containing t, then the meta-process V_t is in state $3^{(7)}$

The generator matrix for V_t is

$$\mathbf{G_v} = \begin{bmatrix} \mathbf{Q}-\mathbf{\Lambda} & \mathbf{\Lambda} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}-\mathbf{\Lambda} & \mathbf{\Lambda} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}-\mathbf{\Lambda} & \dots & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{Q}-\mathbf{\Lambda} & \boldsymbol{\lambda} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & 0 \end{bmatrix} \tag{11}$$

and the block of square matrices comprising of the top d rows of $e^{\mathbf{G_v}t^*}$ give the $(d \times d)$ conditional transition matrices $\mathbf{P}^{(r)}$.

5 Bayesian approach

We are interested in Bayesian analysis of MMPP's. We first briefly discuss the choice of priors, before describing our new Gibbs sampling algorithm. For background on existing MCMC schemes for MMPPs see the introduction and references therein.

5.1 Choice of prior

For computational simplicity we use conjugate priors for the parameters. If we let $\rho_i = -q_{i,i}$, be the rate that the hidden Markov chain leaves state i, and $p_{i,j} = q_{i,j}/\rho_i$ be the probability of a transition to state $j(\neq i)$ when we leave state i, then we assume independent gamma priors for the λ_is and the ρ_is and Dirichlet priors for each vector of probabilities $(p_{i,1}, \dots, p_{i,i-1}, p_{i,i+1}, \dots, p_{i,d})$.

Care must be taken with the parameters of these prior distributions. In particular, improper priors for the parameters can lead to improper posteriors (e.g. Sherlock, 2005). Also we would hope that each $q_{ij} < \lambda_i$ so that most visits to a given state will contain observed events, making it easier to identify the

separate states, as well as to infer λ_i

5.2 Gibbs sampler

We first introduce some notation. We write the state of the chain at event-times (or for D2 and D3 the end of each time-interval) and at the start and end of the observation period as $S_i = X_{t'_i}$. The distribution of the new parameter vector depends on the underlying chain through the starting state (via ν_{s_0}) and three further sufficient statistics, which we now define.

We write $\tilde{t}_i$ for the total time spent in state i by the hidden chain, r_{ij} for the number of times the chain transitions from state i to state j ($r_{ii} = 0\ \forall i$), and n_i for the number of Y-events that occur while the chain is in state i. We correspondingly define $\tilde{\mathbf{t}} = (\tilde{t}_1, \ldots, \tilde{t}_d)^t$, $\mathbf{n} = (n_1, \ldots, n_d)^t$, and $\mathbf{R}$ as the matrix with elements r_{ij} . Our Gibbs sampler acts on augmented state-space $\{\boldsymbol{\lambda}, \mathbf{Q}, X_t\}$, and each iteration has 3 distinct stages:

1. Given the parameter values $(\boldsymbol{\lambda}, \mathbf{Q})$ use the second form of the the forward-backward algorithm (Equations 2 and 3 of Section 2) to simulate the state of the hidden chain X_t at the start and end of the observation interval ($t'_0 = 0$ and $t_{obs} = t'_{n+1}$) and at a set of time points $t'_1, \ldots, t'_n$. For data format D1 $t'_1, \ldots, t'_n$ correspond to event times; for formats D2 and D3 $t'_1, \ldots, t'_{n+1}$ are the end-points of accumulation intervals.

2. Given the parameter values and the finite set of states produced in stage 1, apply the technique of Section 3 to each interval in turn to simulate the full underlying hidden chain X_t from it's exact conditional distribution.

3. Simulate a new set of parameter values.

We now describe how each of the stages may be implemented for each of the three data formats.

Data format D1

For *stage 1* we apply the forward-backward algorithm of section 2 modified to take account of the fact that observation times $t'_1, \ldots, t'_n$ correspond exactly to events of the observed process and that therefore there are no Y-events between observation times. For the k^{th} interval, which has width $t_k = t'_k - t'_{k-1}$, the transition matrix is $\mathbf{T}^{(k)} = e^{(\mathbf{Q}-\mathbf{\Lambda})t_k}$, and the likelihood vector for the k^{th} observation point is $\mathbf{l}^{(k)} = \boldsymbol{\lambda}$.

This process is exactly equivalent to straightforward application of the second form of the forward-backward algorithm to the meta-process W_t of section 4.1.1 on the extended state space $\{1, \ldots, d, 1*\}$, but replacing the d-dimensional vector $\mathbf{1}$ with the $d+1$-dimensional vector $(1, \ldots, 1, 0)^t$. For the k^{th} interval, the transition matrix is now $\mathbf{T}^{(k)} = e^{\mathbf{G_w} t_k}$, where $\mathbf{G_w}$ is defined in (7) and $e^{\mathbf{G_w} t}$ is given explicitly in (8). The likelihood vector is $\mathbf{l}^{(k)} = (\boldsymbol{\lambda}, 0)^t$.

Stage 2 applies the technique of Section 3 directly to extended state space $\{1, \ldots, d, 1^*\}$ with generator matrix $\mathbf{G_w}$.

Figure 1 shows the first two stages for data format **D1**.

Stage 3 is especially simple using our conjugate priors. The likelihood for the full data (observed data and path of hidden chain) is:

$$L(x_t, \mathbf{t}|\mathbf{Q}, \boldsymbol{\lambda}) \propto \nu_{s_0} \times \prod_{i=1}^{d} \prod_{j \neq i} \left(q_{ij}^{r_{ij}} e^{-q_{ij}\tilde{t}_i} \right) \times \prod_{i=1}^{d} \lambda_{s_i}^{n_i} e^{-\lambda_i \tilde{t}_i} \tag{12}$$

Thus conditional on observing the path of the underlying hidden Markov chain, the densities of the λ_is are gamma. The joint conditional densities for the ρ_is and the p_{ij}s is proportional to the product of independent gamma and Dirichlet distributions and the stationary probability of the hidden chain starting in state s_0. This latter distribution can be simulated from using rejection sampling; proposing values from the respective gamma and Dirichlet distributions and accepting then with the resulting stationary probability of s_0.

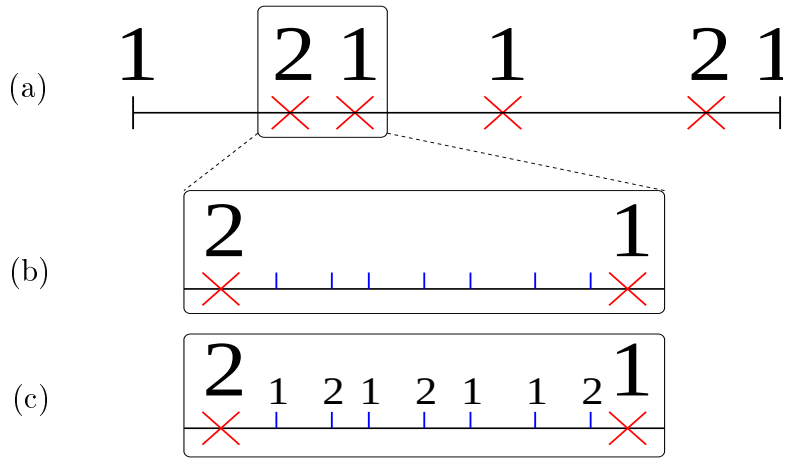

Figure 1: the Gibbs sampler (a) first simulates the chain state at observation times and the start and end time; for each interval it then simulates (b) the number of dominating events and their positions, and finally (c) the state changes that may or may not occur at these dominating events. The figure applies to a two-state chain with $\lambda_2 + q_{21} > \lambda_1 + q_{12}$.

Data format D2

For *stage 1* we apply the second form of the forward-backward algorithm with likelihood vector $\mathbf{l}^{(k)} = \mathbf{1}$ and transition matrix dependent on the binary indicator (b_k) for the interval

$$\mathbf{T}^{(k)} = \mathbf{P}^{(0)\,1-b_k}\ \overline{\mathbf{P}}^{\,b_k}$$

For *stage 2* we first consider the meta-process $\overline{W}_t$ on state space $\{1, \ldots, d, 1^*, \ldots, d^*\}$ with $\overline{W}_t = X_t$ when $Y_t = 0$ and $\overline{W}_t = X_t^*$ otherwise.

This has generator matrix

$$\mathbf{G}_{\overline{\mathbf{W}}} = \begin{bmatrix} \mathbf{Q} - \mathbf{\Lambda} & \mathbf{\Lambda} \\ \mathbf{0}^* & \mathbf{Q} \end{bmatrix}$$

For a given interval suppose that we have simulated X_t starting in state s_0 and ending state s_1. On the extended state space this corresponds to starting in state s_0 and finishing in state s_1 if there have been no events over the interval, otherwise finishing in s_1^* . We simulate the underlying chain from the algorithm of section 3. This also supplies the time of the first event in the interval, and the state at the time of this event, which we use for simulating the new parameters in stage 3.

In *stage 3*, for accumulation interval i define t_{ij}^* as the amount of time that the hidden chain spends in state j between the start of the interval and either the time of the first event (if there is a first event) or the end of the interval. Further let $t_j^* = \sum_{i=1}^{n+1} t_{ij}^*$ be the known time that the hidden chain is in state j, and n_j^* the number of intervals for which the first event occurs while the hidden Markov chain is in state j. Then the full-data likelihood is

$$L(x_t, \mathbf{t}|\mathbf{Q}, \boldsymbol{\lambda}) \propto \nu_{s_0} \times \prod_{i=1}^{d} \prod_{j \neq i} \left(q_{ij}^{r_{ij}} e^{-q_{ij}\tilde{t}_i} \right) \times \prod_{j=1}^{g} \lambda_j^{n_j^*} \mathrm{e}^{-\lambda_j t_j^*} \tag{13}$$

We then proceed as with data format D1.

Data format D3

For this data format we consider the meta-process V_t on extended state space $\{1^{(0)}, \ldots, d^{(0)}, 1^{(1)}, \ldots d^{(1)}, \ldots, 1^{(c_{max})}, \ldots, d^{(c_{max})}, 1^*\}$ as defined in section 4.1.2.

For the application of the forward-backward algorithm in *stage 1*, the transition matrices are $\mathbf{T}^{(k)} = \mathbf{P}^{(c_k)}$ and the likelihood vectors are $\mathbf{l}^{(k)} = \mathbf{1}$. For *stage 2*, in simulating from the exact distribution of the underlying chain for an interval where the start state is s_0, the end state is s_1 and there are c_k events observed we use the generator matrix $\mathbf{G_v}$ as defined in (11) with start state s_0 but end state $s_1^{(c_k)}$.

The algorithm also simulates from the exact distribution of the times at which each of the c_k events occurs over the interval, therefore we may perform *stage 3* exactly as for data format D1.

6 Analysis of Chi site data for *E.coli*

6.1 Background and the *E.coli* data

In recent years there has been an explosion in the amount of data describing both the genomes of different organisms, and the biological processes that effect the evolution of these genomes. There is much current interest in understanding the function of different features of the genome and what affects the biological processes such as mutation and recombination. One approach to learning about these is via genome segmentation (e.g. Li *et al.*, 2002): partitioning a genome into regions that are homogeneous in terms of some characteristic (e.g GC content), and then looking for correlations between this characteristic and either another characteristic, or a biological process of interest.

Here we consider segmentation of a bacterial genome based on the rate of occurence of a particular DNA motif - called the Chi site. The Chi site is

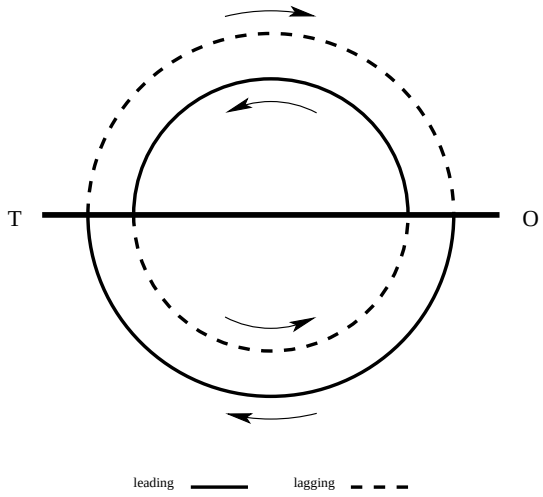

Figure 2: schematic of the leading and lagging strands on the inner and outer rings of the *E.coli* genome split by the replication origin (O) and terminus (T), together with the direction relevant for Chi site identification.

a motif of 8 basepairs: GCTGGTGG. The Chi site is of interest because it stimulates DNA repair by homologous recombination (Gruss and Michel, 2001), so the occurence of Chi sites has been conjectured to be related to recombination hotspots.

Our data is for *E.coli* DNA and consists of the position (in bases) of Chi sites along the genome. Figure 2 shows a schematic of the circular double stranded DNA genome of *E.coli*, with the two strands represented by the inner and outer rings. There is a 1-1 mapping of bases between the outer and inner strands (C $\leftrightarrow$ G and A $\leftrightarrow$ T) so that each uniquely determines the other. The figure also indicates a directionality associated with different halves of each strand as split by the replication origin (O) and terminus (T). The molecular mechanisms of DNA replication differ between the two half-strands and they are termed *leading* and *lagging*, as indicated in the figure.

The 1-1 mapping between base pairs together with the reversing of direc-

tionality between inner and outer strands implies that searching the Chi site in the outer strand is equivalent to searching for CCACCAGC in the inner strand. This sequence is different enough from the sequence of the Chi site in the inner strand, that occurrences of the Chi site in inner and outer strands are effectively independent. Occurence of Chi sites in leading and lagging halves are also independent since these are separate parts of the genome. Thus our data consists of four independent sets of positions of Chi sites - along leading and lagging halves of both inner and outer strands. Figure 3 shows the cumulative number of events along the genome for each of these data sets.

The replication and repair mechanisms for leading strands are different to those for lagging strands so in general we might expect them to have different compositional properties (densities of nucleotides and oligonucleotides). A bias in the frequency of Chi sites favouring leading strands has been noted in several genomes, including *E.coli* (e.g. Karoui *et al.*, 1999) and is evident from the figure. A more open question is whether there is variation within the leading and/or lagging strands, rather than just between the leading and lagging strands.

Our aim is to first determine whether Chi sites appear to occur uniformly at random within each of the leading and lagging strands, or whether there is evidence of the intensity of the occurence of Chi sites varying across either strand. Secondly, if there is variation then we would like to infer the regions with strong evidence for either a high or low intensity of Chi sites.

The *E.coli* genome (defined as single strand length) is 4 639 675 bases long so each of the individual halves are 2319.838 kilobases (kb) long. Henceforth we use units of kb.

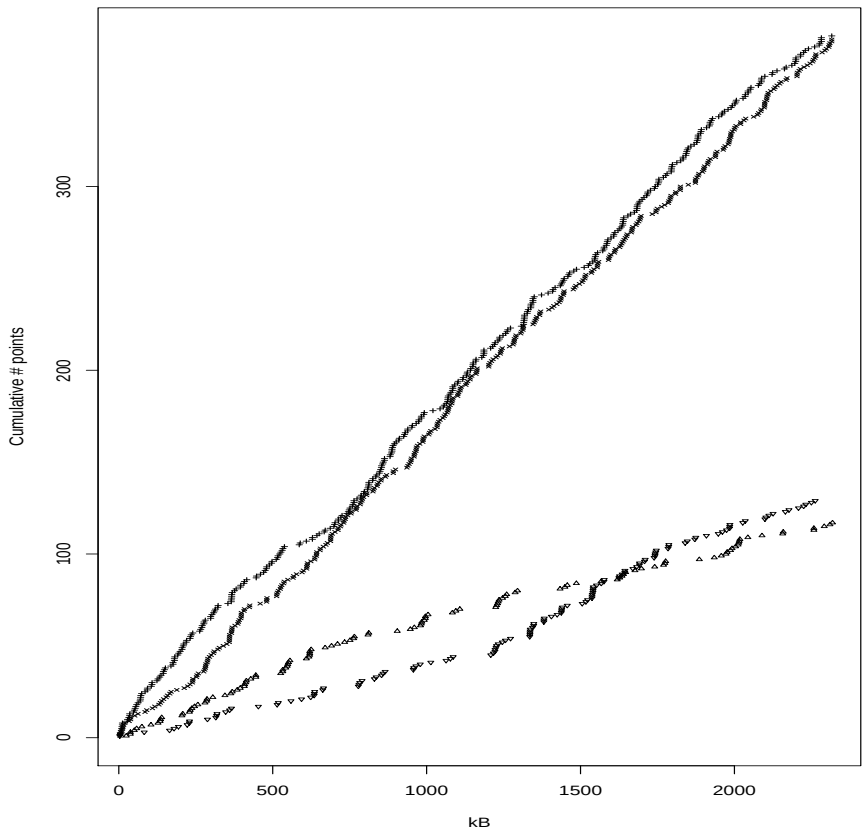

Figure 3: cumulative number of occurences of the Chi site along the genome for leading (+) and lagging ($\triangle$) halves of the outer strand and leading ($\times$) and lagging (∇) halves of the inner strand.

6.2 Model and prior

We analyse the positions of occurences of the Chi site along first leading then lagging strands using our Gibbs sampler. These positions are discrete bases and our Gibbs sampler applies to continuous data, however each of the four strands is over 2319kb long and contains less than 400 occurences of the 8-base Chi site, so it is reasonable to model this discrete process as continuous. Furthermore, a straightforward approach to discrete modelling would involve applying the forward-backward algorithm across the entire genome, which would be computationally prohibitive.

One of our aims is to perform model choice, and the choice of model will depend on the priors for each model; in particular we cannot use uninformative priors (e.g. Bernardo and Smith, 1995, Chapter 6). For the results we present here we take exponential priors (that is gamma densities with shape parameters equal to 1) for the λ_is and the ρ_is (gamma densities with shape parameter of less than 1 will lead to posteriors with an infinite density at 0); and uniform priors for the vectors of transition probabilities.

We first analyse the inner leading and lagging strands and use the results from these to inform priors for analyses of the outer leading and lagging strands, which we use to perform model choice. We also tested robustness of our results to variation in the priors.

We analyse the inner strands using exponential priors, the means of which are chosen empirically from the data for each strand. The mean for all λ parameters is set to n/t_{obs}, where n and t_{obs} are respectively the number of Chi sites and the total length in kb of the strand. The mean for all q parameters needs to be somewhere between $1/t_{obs}$ and n/t_{obs} for an analysis to be feasible so we set it to $\sqrt{n}/t_{obs}$. These latter choices are rather arbitrary, but the resulting posteriors are only used to inform the (weak) priors for the analyses of the outer strands.

Since the priors for the inner strand are exchangeable and the likelihood of an MMPP is invariant under permutation of the states, so too is the joint posterior. We therefore order the results from the analysis of the inner strand such that $\lambda_1 \leq \lambda_2$ and use the posterior means as means for the exponential priors for the analysis of the outer strands. Since the runs for the outer strands have non-exchangeable priors, we may not order the output and must treat it exactly as it appears.

For each strand we analyse the 1-d case analytically and the 2-d and 3-d cases using 100000 iterations of our Gibbs sampler. Gibbs sampler code was written in C and, when run on an AMD Athlon 1458MHz CPU, took approximately 11 minutes to perform 100000 iterations on the outer lagging strand. This strand contains 117 Chi-sites.

Matrix exponentials were calculated by truncating (4). The truncation was set so that the error in each element of the matrix exponential was less than a pre-determined tolerance (this was efficient as errors decay faster than geometrically, and accurate as it involves summing only positive values). The sum can be evaluated efficiently for all intervals lengths by calculating and storing the required powers of $\mathbf{M}$ once for each iteration. The powers of $\mathbf{M}$ are also then used when simulating the underlying hidden chain.

6.3 Results

Figure 4 shows trace plots for the first 20 000 iterations and ACF's over the first 10 000 iterations for the 2-d run on the lagging strand of the outer ring. The trace plot for λ_1 shows one of only 6 mode-switch-and-return's (all brief), indicating that the different priors fix quite firmly the ordering of the states. These brief switches do however exert a strong (and spurious for our purposes) influence on the ACF's, and so we show ACF's for a period in which there is no

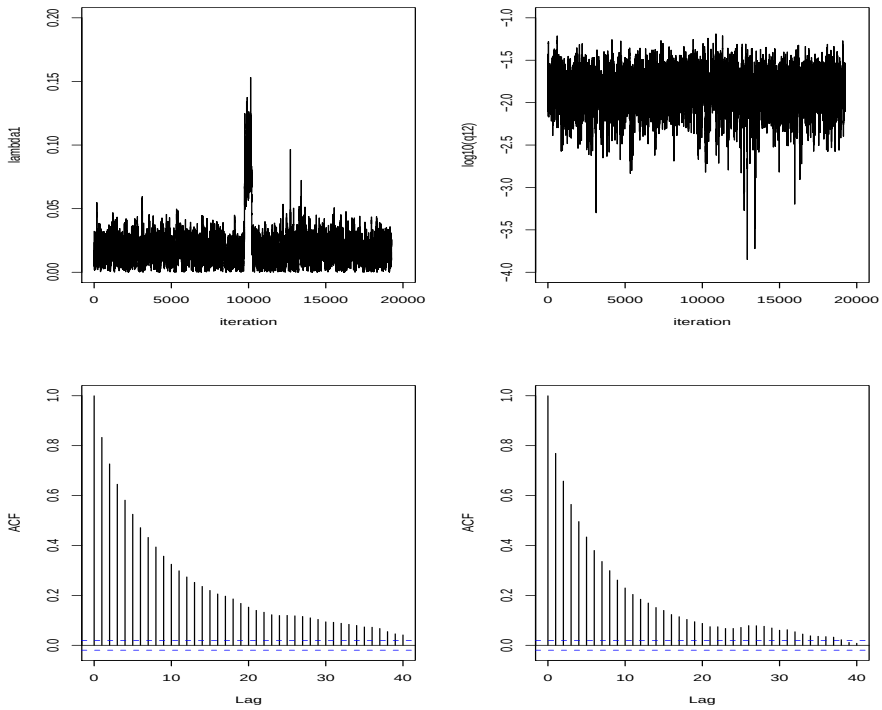

Figure 4: trace plots for the first 20 000 iterations and and ACF's for the first 10 000 iterations of the Gibbs sampler for the lagging strand of the outer ring with non-exchangeable priors derived from the run for the lagging strand of the inner ring.

Dataset	1-D	2-D	3-D
lagging (outer)	<0.01	0.83	0.17
leading (outer)	0.30	0.44	0.26

Table 1: Posterior model probabilities for leading and lagging halves of the outer strand.

mode-switching; the mixing appears to be satisfactory.

Posterior model probabilities for the leading and lagging strands were calculated using the method of Chib (1995) and are given Table 1. They indicate a clear choice of a two-dimensional model over a one-dimensional model for the lagging strand. There is also substantial evidence for a two-dimensional model in preference to a three-dimensional model. From the model probabilities alone there is nothing to choose between one, two, and three dimensional models for leading strands.

For the two-dimensional model for lagging strands the posterior mean parameter values correspond to intensities of 20.8 and 92.1 Chi sites per megabase (Mb), and an intensity of 16.0 transfers per Mb from the lower state to the higher state and 21.1 transfers per Mb from the higher state to the lower state. The one-dimensional model for leading strands has posterior mean intensity of 164.7 Chi sites per Mb.

Posterior model probabilities may be sensitive to the exact prior used, and since the data contains less information about the q parameters than the λ parameters, the q priors may be particularly influential. We performed further analyses of the outer and inner rings with exchangeable exponential priors for λ and with exchangeable exponential, (approximately) normal, and truncated exponential priors for q. There was little change in the posterior means for ordered (λ_1, λ_2), but a great deal of variability in (q_{12}, q_{21}) as expected. However

the posterior model probabilities always indicated at least a two-state model for lagging strands and little to choose between one and two state models for leading strands.

A possible biological explanation for our results is given by how replication differs on leading and lagging strands. Leading DNA strands are replicated continuously whereas lagging strands are replicated in fragments. It may be the fragmentary nature of replication that is causing the hetrogeneity in rate of occurrence of Chi sites.

We can use the output of the Gibbs sampler to perform segmentation of the lagging strands based on the intensity of the occurence of Chi sites. Figure 5 plots the mean (over 1000 chains sampled every 100 iterations) intensity against position along the genome. This gives a 'smoothed signal' of Chi site intensity which could be used to evaluate correlations with (say) recombination rates across the genome. An alternative segmentation might be based on the posterior probabilities that a given point along the genome is in each of the possible states - for this segementation, at each point the chain is simply set to the state with the highest posterior probability.

7 Discussion

We have presented a novel approach to simulating directly from the conditional distribution of a continuous time Markov process and shown how this can be used to implement a Gibbs sampler for analysing MMPPs. The Gibbs sampler can analyse data where the event-times are directly observed, and also data where the number of events or even only the presence/absence of events is known for a sequence of time intervals.

The Gibbs sampler has a number of advantages over standard Metropolis-

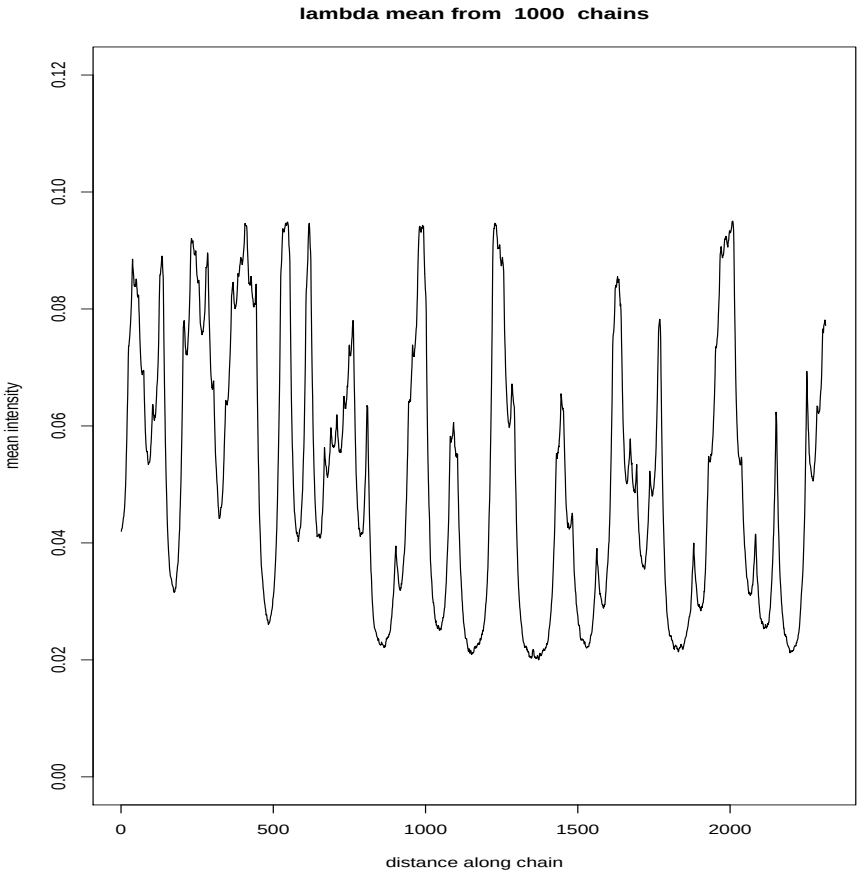

Figure 5: mean λ value from 1000 chains at each point in the lagging strand

Hastings samplers. Firstly, the Gibbs sampler requires no tuning; tuning for Metropolis-Hastings algorithms can be time consuming - especially for long datasets where the algorithm takes longer to run and for algorithms involving blocking of parameters. Further such tuning is valid for the area of the posterior being explored whilst the tuning takes place (hopefully the mode); there is no guarantee that it will be appropriate for as yet unseen tail areas that the algorithm should eventually explore.

Secondly, a by-product of the Gibbs sampler is that we can investigate the posterior distribution of the underlying chain. This allowed us to identify regions of high intensity of Chi site occurences on the lagging strand of *E.coli* DNA.

There has been previous work on developing a Gibbs sampler for MMPP's. Scott (1999) and Scott and Smyth (2003) present an approximate Gibbs sampler that can be applied to certain MMPP's, assuming the event times are directly observed. Their approximation is to assume that certain state changes coincide precisely with observed events. In many situations this approximation will be negligible; Scott (1999) models times at which a bank account is accessed, where a criminal may or may not have obtained the bank details; it is argued that it is sensible to *define* the arrival of a criminal as the time at which he/she first accesses the account. Further Scott and Smyth (2003) argue that forcing state changes to start and end at event-times 'eliminates the possibility of pathological bursts containing no events'. However their Gibbs sampler also places restrictions on the allowable state changes: all transitions to states with lower intensities than the current state are permitted, but out of all the (ordered) states with higher intensity than the current state, transitions are only permitted to the state immediately adjacent to the current one. Also the approximation of restricting state changes to event times will become less accurate as the rates of the generator for the hidden chain increase towards the same order of magnitude

as the intensities of the observed process. Our Gibbs sampler avoids these issues and there is little extra cost in implementing it.

Blackwell (2003) and Bladt and Sorensen (2005) use rejection sampling to sample from the exact distribution of a discretely observed continuous-time Markov process. A chain is simulated forward from a given observed state, and if the simulated state at the next observation time does not match the corresponding observed state then the chain is rejected and the process repeated until a match is achieved. A similar technique could replace stage 2 of our Gibbs sampler, where we simulate from the hidden chain and the observed event process and accept the hidden chain if the chain finishes in the correct state and there are no observed events. This is efficent only when the number of rejected chains is small. It is straightforward to calculate the expected number of simulations until acceptance for an interval of known length given the start and end states. We calculated this for the simulated states at event times at every iteration of our Gibbs sampler for every one of the 1164 intervals in a data set simulated over an observation window of 100 seconds with intensities $\lambda_1 = 10, \lambda_2 = 13$ and rates for the hidden Markov chain $q_{12} = q_{21} = 1$. On average for about 700 of the intervals 3 or fewer chain simulations were expected to be required. However the distribution of the expected number of simulations had a very heavy right hand tail, with about 200 intervals requiring at least 10 simulations and about 20 requiring more than 100 simulations, so that the mean expected number of simulations per interval was around 20. This number is likely to increase as the number of hidden states increases. In practice stage 2 of our Gibbs sampler takes a very small proportion of the CPU time and this would be likely to remain small if rejection sampling were to be used instead, unless the number of rejections was large.

We considered the application of MMPPs to modelling the occurence of a

specific DNA motif in *E.coli*. We found evidence for heterogeneity in the occurrence of this DNA motif, the Chi site, in the lagging strand; which may have a biological explanation in terms of the replication process on this strand. The output of our Gibbs sampler also enables us to segment the lagging strand into regions of high and low intensity of these Chi sites. Ideally we would like to use this segmentation to test for correlation of high Chi site intensity with regions of high recombination rates, but unfortunately data is not currently available on the variation in recombination rate in *E.coli*.

A computer program, written in C, which implements the Gibbs sampler for event-time data is available from:
http://www.maths.lancs.ac.uk/∼sherlocc/MMPP/index.html

Acknowledgements: We dedicate this paper to the memory of Nick Smith who helped with the application of our method to the analysis of E.Coli DNA. The first author acknowledges support from EPSRC grants GR/R91724/01 and GR/T19698/01.

References

Asmussen, S. (2000). Matrix-analytic models and their analysis. *Scandinavian Journal of Statistics* **27**, 193–226.

Baum, L. E., Petrie, T., Soules, G. and Weiss, N. (1970). A maximisation technique occuring in the statistical analysis of probababilstic functions of Markov chains. *The Annals of Mathematical Statistics* **41**, 164–171.

Bernardo, J. M. and Smith, A. F. M. (1995). *Bayesian Theory*. Wiley, Chichester, UK.

Blackwell, P. G. (2003). Bayesian inference for Markov processes with diffusion and discrete components. *Biometrika* **90**, 613–627.

Bladt, M. and Sorensen, M. (2005). Statistical inference for discretely observed Markov jump processes. *Journal of the Royal Statistical Socienty, Series B* **67**, 395–410.

Burzykowski, T., Szubiakowski, J. and Ryden, T. (2003). Analysis of photon count data from single-molecule fluorescence experiments. *Chemical Physics* **288**, 291–307.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association* **90**, 1313–1321.

Davison, A. C. and Ramesh, N. I. (1996). Some models for discretised series of events. *Journal of the American Statistical Association* **91**, 601–609.

Fearnhead, P. and Meligkotsidou, L. (2004). Exact filtering for partially-observed continuous-time Markov models. *Journal of the Royal Statistical Society, series B* **66**, 771–789.

Fischer, W. and Meier-Hellstern, K. (1992). The Markov-modulated Poisson process (MMPP) cookbook. *Performance evaluation* **18**, 149–171.

Gilks, W. R., Richardson, S. and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in practice*. Chapman and Hall, London, UK.

Gruss, A. and Michel, B. (2001). The replication-recombination connection: insights from genomics. *Current Opinion in Microbiology* **4**, 595–601.

Karoui, M. E., Biaudet, V., Schbath, S. and Gruss, A. (1999). Characteristics of chi distribution on different bacterial genomes. *Res. Microbiol.* **150**, 579–587.

Kou, S. C., Xie, X. S. and Liu, J. S. (2005). Bayesian analysis of single-molecule experimental data. *Appl. Statist.* **54**, 1–28.

Li, W., Bernaola-Galvan, P., Haghighi, F. and Grosse, I. (2002). Applications of recursive segmentation to the analysis of DNA sequences. *Computers and Chemistry* **26**, 491–510.

Ross, S. (1996). *Stochastic Processes, 2nd Ed.*. John Wiley and Sons, Inc., New York.

Ryden, T. (1996). An EM algorithm for estimation in Markov-modulated Poisson processes. *Computational Statistics* **21**, 431–447.

Scott, S. L. (1999). Bayesian analysis of a two-state Markov modulated Poisson process. *Journal of Computational and Graphical Statistics* **8**, 662–670.

Scott, S. L. and Smyth, P. (2003). The Markov modulated Poisson process and Markov Poisson cascade with applications to web traffic modelling. *Bayesian Statistics* **7**, 1–10.

Sherlock, C. (2005). In discussion of 'Bayesian analysis of single-molecule experimental data'. *Journal of the Royal Statistical Society, Series C* **54**, 500.