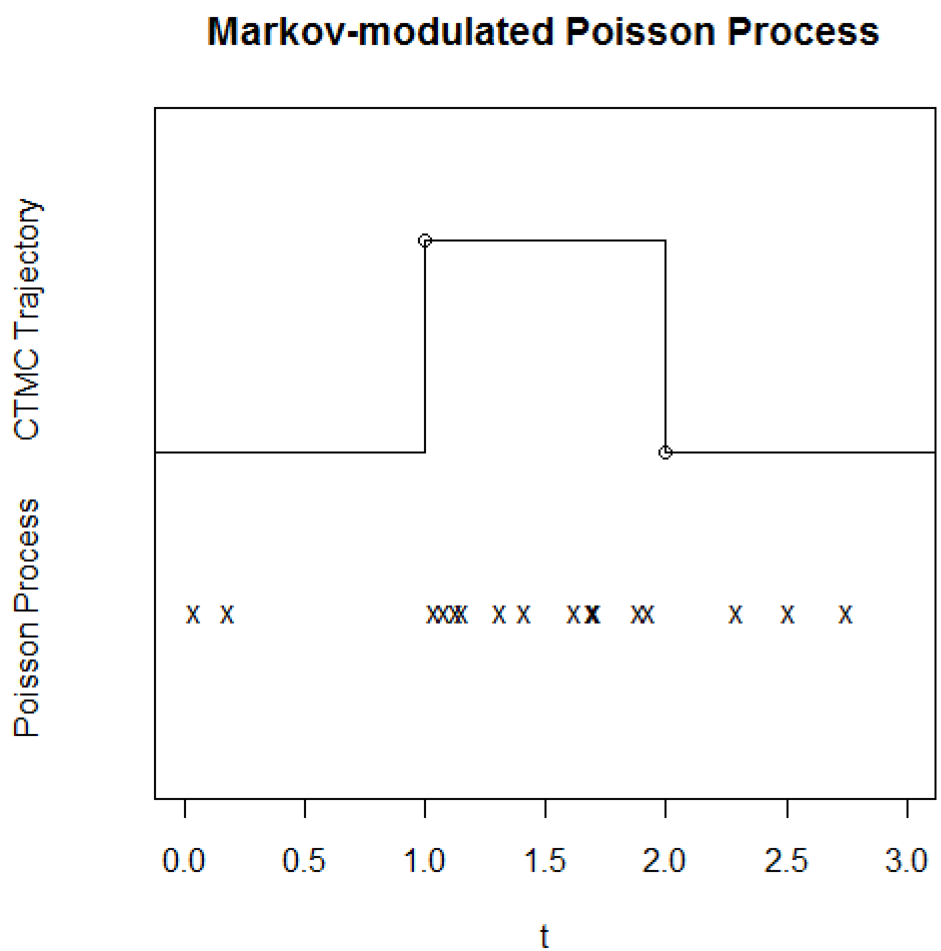
An exact Gibbs sampler for the Markov-modulated Poisson Process

Paul Fearnhead and Chris Sherlock

Markov-modulated Poisson process

- A Poisson process whose intensity is determined by a continuous-time Markov chain (CTMC)



Of what use is it?

MMPPs model a wide variety of phenomena

- Photon emission rate for a molecule switching between excited and ground states
- Frequencies of bank transactions
- Web page requests over time
- Telecom network overflow
- DNA motif propagation (featured example)

Objective

We seek to sample from the posterior distributions of:

- The trajectory of the hidden CTMC
- The infinitesimal generator for the hidden CTMC
- The Poisson intensities

Methodology

1) Forward-backward algorithm; given a set of parameters, use the forward-backward algorithm to simulate the state of the hidden chain at the event times.

2) Simulate a continuous-time Markov chain; given the above, simulate the full hidden chain trajectory.

3) Update the parameters; given the hidden chain, and predetermined priors, generate a new set of Poisson intensities and a new infinitesimal generator from their full conditional distributions.

Forward-backward algorithm

We can recurse backward through the data points, gathering information.

$$\left[\mathbf{A}^{(k)}\right]_{ij} = P(\mathbf{d}_{k:n}, s_{n+1} = j | s_{k-1} = i) = \left[\mathbf{T}^{(k)}\mathbf{L}^{(k)}\mathbf{A}^{(k+1)}\right]_{ij}$$

$$\mathbf{A}^{(n+1)} = \mathbf{T}^{(n+1)}$$

We can then recursively impute the hidden states at the observation times.

$$P(S_0 = s|\mathbf{d}) = \frac{\mu_s[\mathbf{A}^{(1)}\mathbf{1}]_s}{\mu^{\mathrm{T}}\mathbf{A}^{(1)}\mathbf{1}} \qquad P(S_k = s|\mathbf{d}, s_{k-1}) = \frac{T^{(k)}_{s_{k-1},s} l^{(k)}_s [\mathbf{A}^{(k+1)}\mathbf{1}]_s}{[\mathbf{A}^{(k)}\mathbf{1}]_{s_{k-1}}}$$

Uniformization

Let G be the generator matrix for a continuous-time Markov chain with diagonal entries $-\rho_{1,}-\rho_{2,}\ldots,-\rho_d$.

Let $\rho = \max_i \rho_i$ and $M = \frac{1}{\rho} G + I.$ If we use ρ as a homogeneous Poisson intensity, and M as a transition matrix invoked at each Poisson point, the result is a realization of the original CTMC.

This allows us to consider a homogeneous process that generates ("dominates") a continuous-time Markov chain.

Chain simulation

1) Impute the number of dominating events; from a single dominating Poisson process.

2) Impute the positions of the dominating events; given the number of events.

3) Impute the state change at each dominating event; due to domination, it is possible to remain at the same state after a dominating event.

Chain simulation

We calculate the number of dominating events:

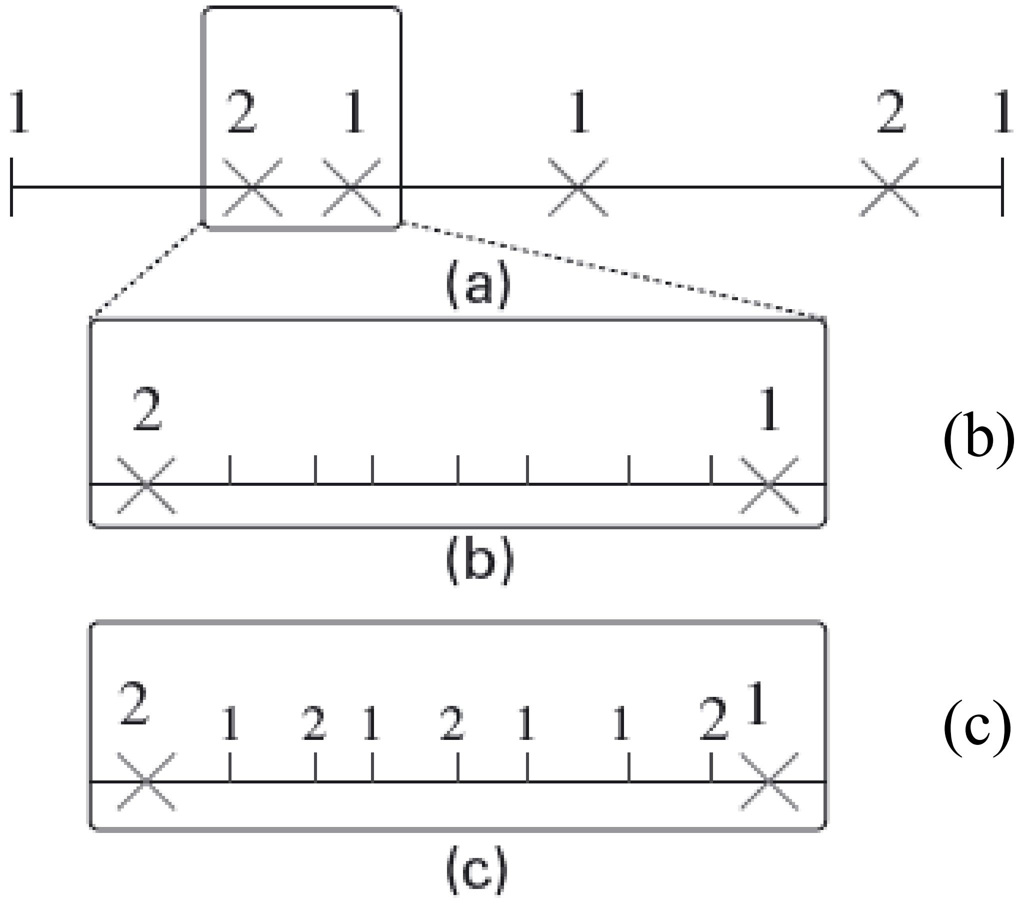
$$P(N_U(t) = r | W_t = j, W_0 = i) = \frac{\left[\exp(-\rho t)\frac{(\rho t)^r}{r!}\right][\mathbf{M}^r]_{ij}}{[\exp(\mathbf{G}t)]_{ij}}$$

Since the dominating process is homogeneous, we can distribute the events uniformly over the subinterval.

We can recurse forward through the dominating events, imputing the (potential) state change.

$$P(s_l^* = j | s_t = k, s_{l-1}^* = i) = \frac{[\mathbf{M}]_{ij}\left[\mathbf{M}^{r-l}\right]_{jk}}{[\mathbf{M}^{r-l+1}]_{ik}}$$

Methodology illustration



(a) Impute hidden states at event times

(b) Generate quantity and locations of dominating events

(c) Impute the state transitions at the dominating events

Sampling parameters

With the imputed hidden chain trajectory in hand, the complete-data likelihood is simple to calculate.

$$L(X_{\mathbf{t}}, \mathbf{t}) \propto \nu_{s_0} \prod_i \left(\prod_{j \neq i} q_{ij}^{r_{ij}} \right) \exp(-\rho_i \tilde{t}_i) \prod_i \lambda_i^{n_i} \exp(-\lambda_i \tilde{t}_i)$$

Sampling parameters

Using independent Gamma priors for the Poisson intensities and diagonal entries in the generator, and using Dirichlet priors to fill out the generator, we get a closed form for the full conditional:

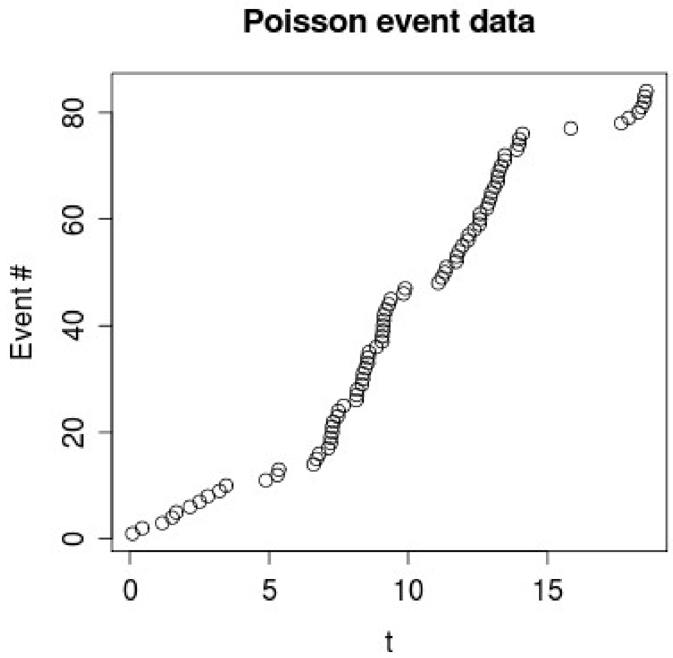
$$\pi(\mathbf{Q}, \boldsymbol{\lambda} | X_{\mathrm{t}}, \mathrm{t}) \propto \nu_{s_0} \prod_i \prod_{j \neq i} p_{ij}^{r_{ij} + \alpha_{p_{ij}} - 1} \prod_i \rho_i^{r_{i+} + \alpha_{\rho_i} - 1} \exp(-\rho_i(\tilde{t}_i + \beta_{\rho_i}))$$
$$\prod_i \lambda_i^{n_i + \alpha_{\lambda_i} - 1} \exp(-\lambda_i(\tilde{t}_i + \beta_{\lambda_i})).$$

This form is easily factored into the product of Gamma and Dirichlet distributions. This makes sampling from the full conditional very straightforward.

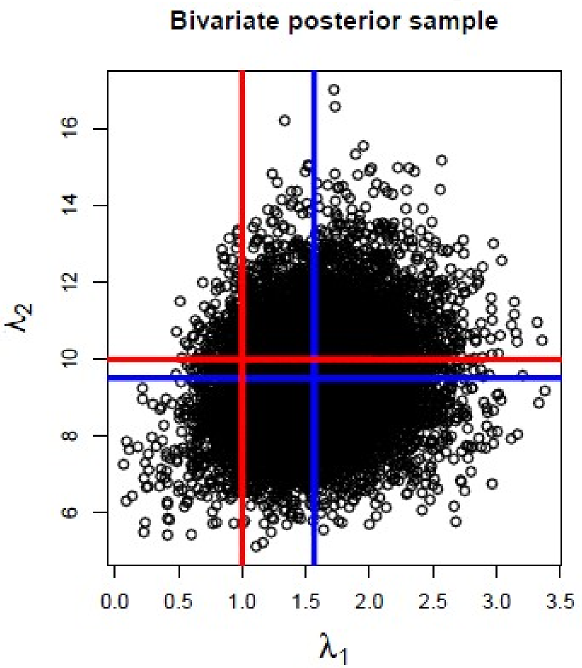
Results (simulated data)

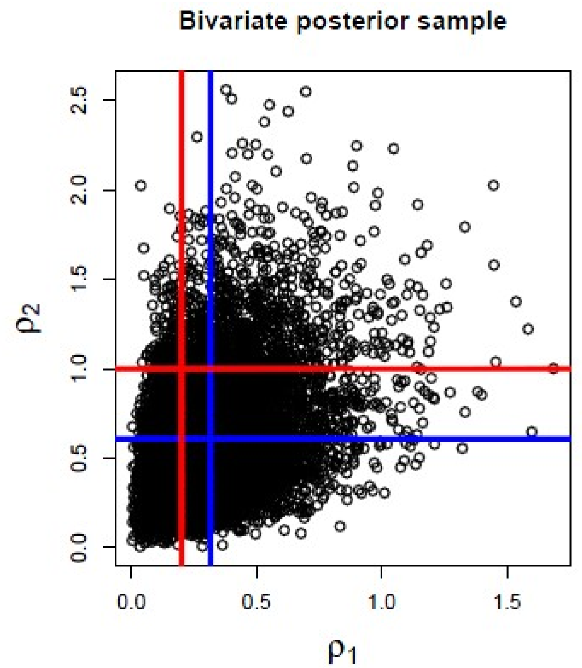
$$\lambda_1 = 1, \ \ \lambda_2 = 10, \ \ \rho_1 = 0.2, \ \ \rho_2 = 1.0$$

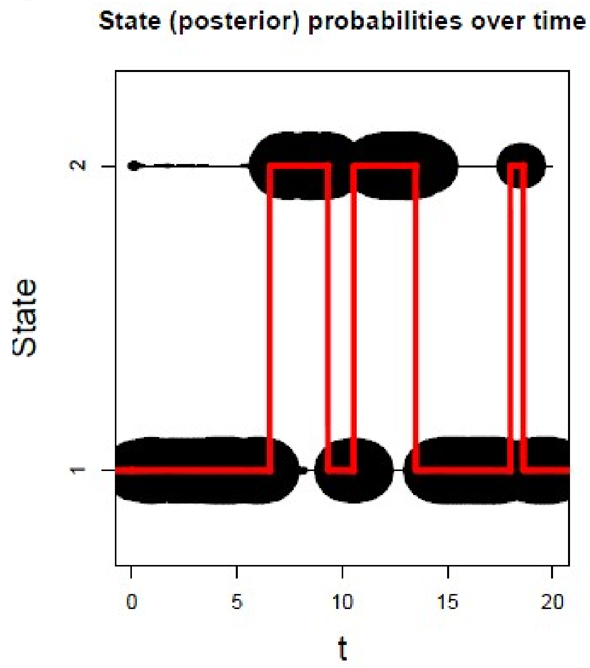
State	Time
1	0.000
2	6.559
1	9.315
2	10.543
1	13.489
2	17.986
1	18.605



Results (simulated data)







$$\hat{\lambda}_1 = 1.566$$
$$CI_{0.95} = (0.759, 2.500)$$

$$\hat{\lambda}_2 = 9.499$$
$$CI_{0.95} = (6.881, 12.586)$$

$$\hat{\rho}_1 = 0.316$$
$$CI_{0.95} = (0.061, 0.802)$$

$$\hat{\rho}_2 = 0.608$$
$$CI_{0.95} = (0.145, 1.417)$$

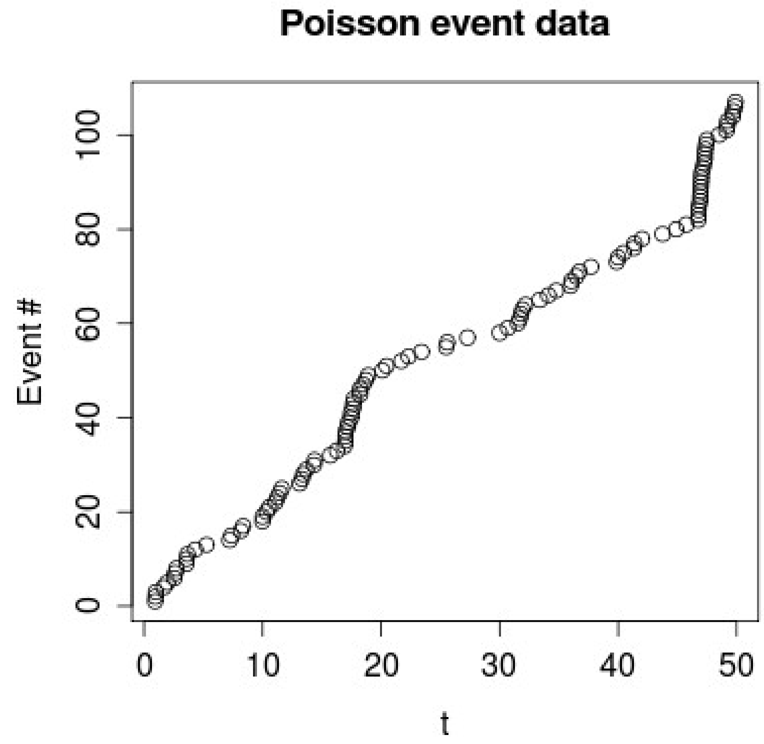
Red is the true value
Blue is posterior mean

Results (3 state simulated data)

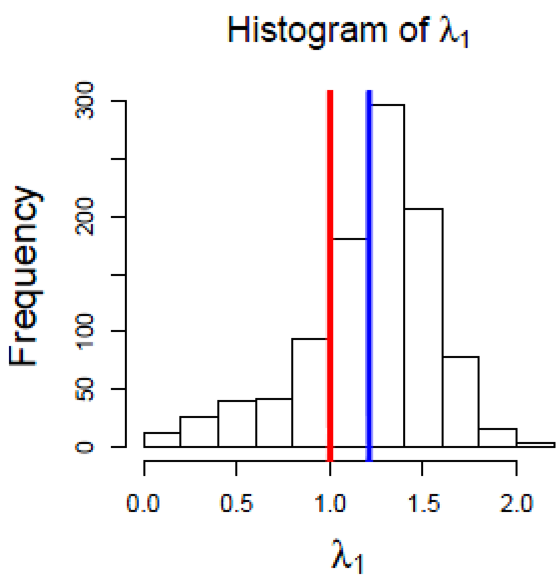
State	Time
1	0.000
3	0.933
1	1.050
3	15.695
1	15.733
2	16.883
1	18.736
2	39.185
3	39.334
1	39.348
3	46.704
1	47.536
2	49.085

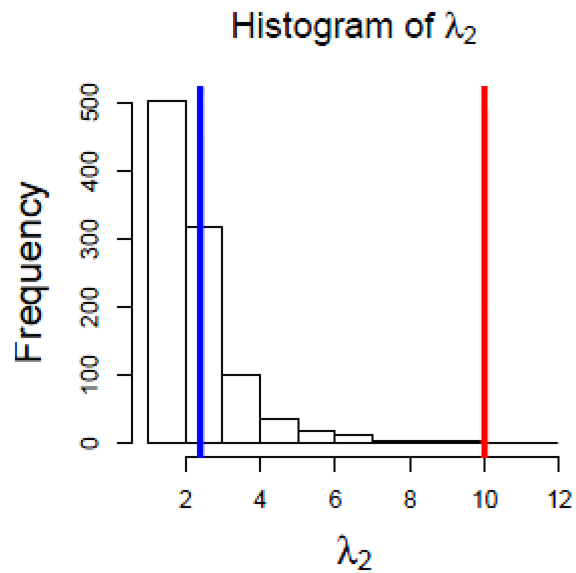
$$\lambda_1 = 1, \;\; \lambda_2 = 10, \;\; \lambda_3 = 30$$

$$\rho_1 = 0.2, \;\; \rho_2 = 1.0, \;\; \rho_3 = 5.0$$



Results (3 state simulated data)





$$\hat{\lambda}_1 = 1.210$$

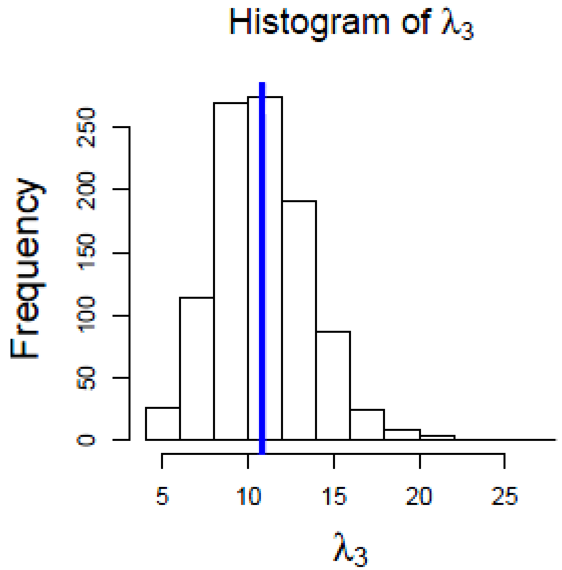
$$CI_{0.95} = (0.312, 1.772)$$

$$\hat{\lambda}_2 = 2.400$$

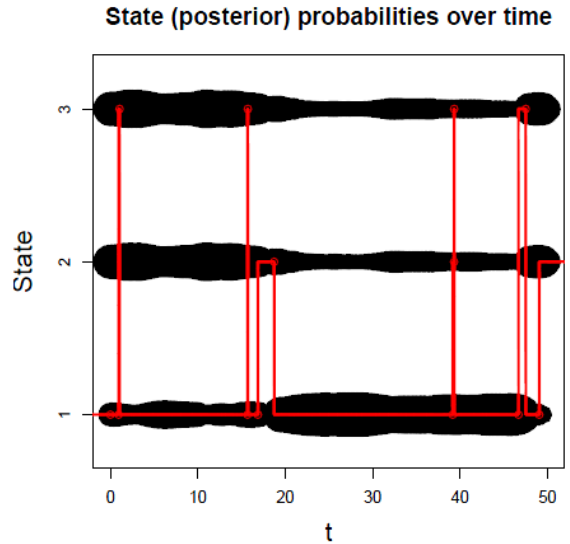
$$CI_{0.95} = (1.317, 5.952)$$

$$\hat{\lambda}_3 = 10.830$$

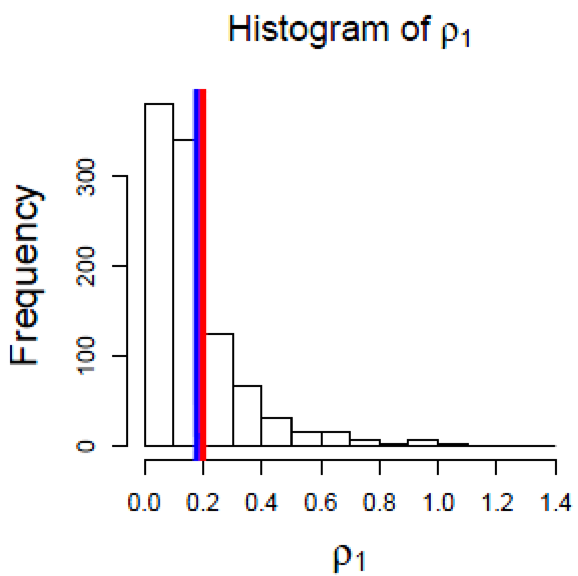
$$CI_{0.95} = (5.986, 17.095)$$

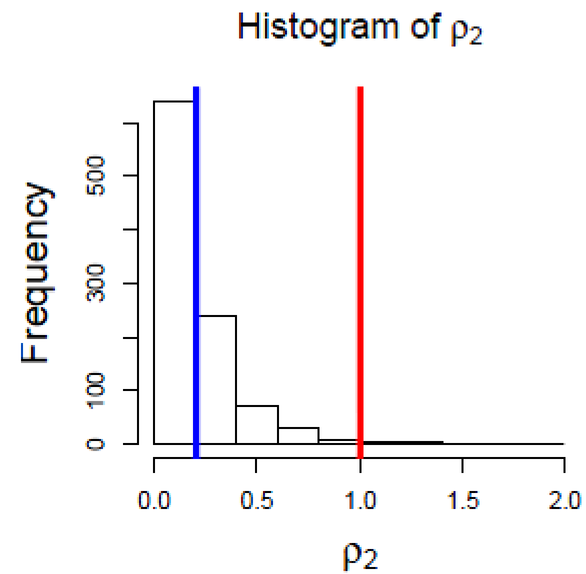


Red is the true value
Blue is posterior mean



Results (3 state simulated data)





$\hat{\rho}_1 = 0.178$

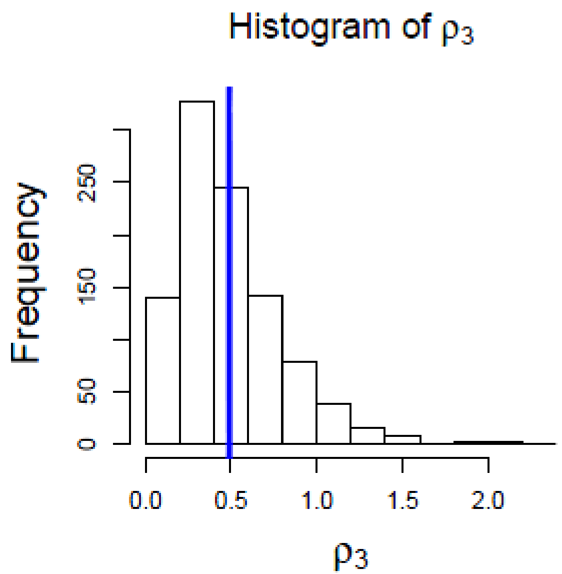
$CI_{0.95} = (0.021, 0.672)$

$\hat{\rho}_2 = 0.204$

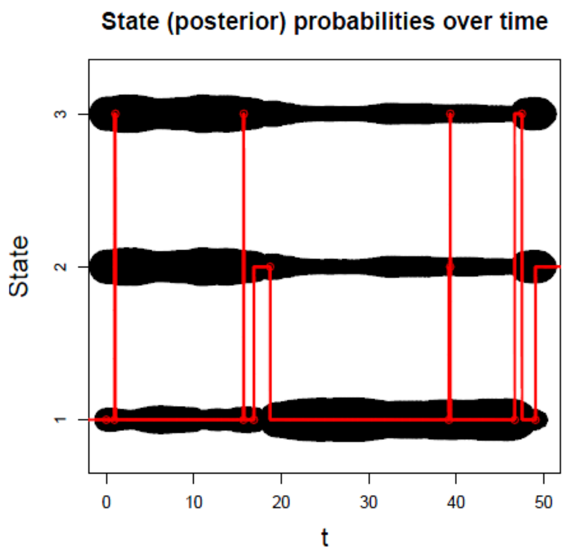
$CI_{0.95} = (0.021, 0.708)$

$\hat{\rho}_3 = 0.490$

$CI_{0.95} = (0.097, 1.238)$



Red is the true value
Blue is posterior mean



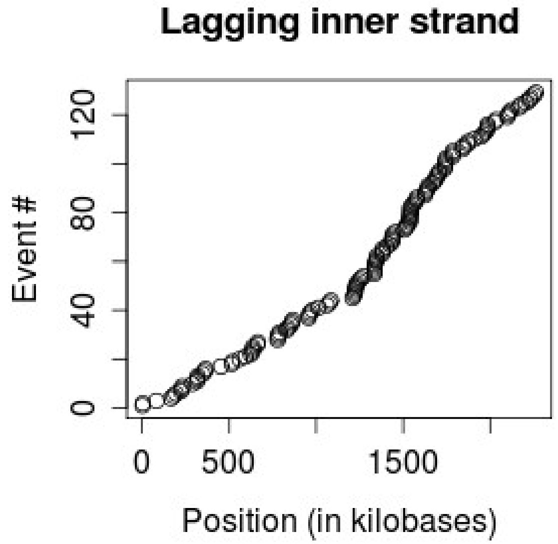
Chi site example

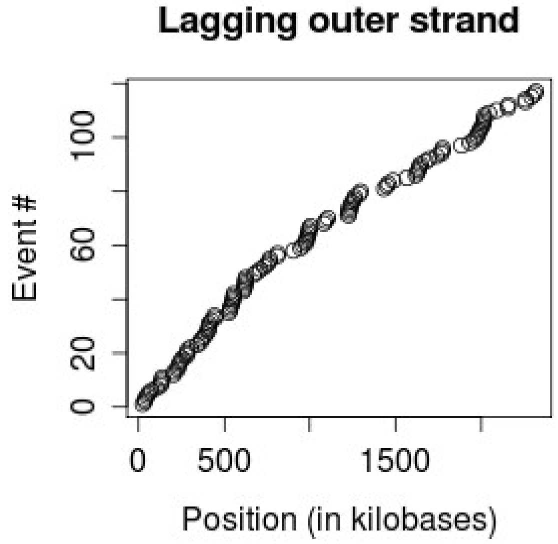
Chi site: a sequence of eight base pairs—GCTGGTGG—in the *E. Coli* genome.

We seek evidence that the Chi site loci follow an MMPP and attempt to determine the latent trajectory and parameters.

We compare sections of the genome that the authors grouped together to asses their compatibility.

Results (Chi site data)





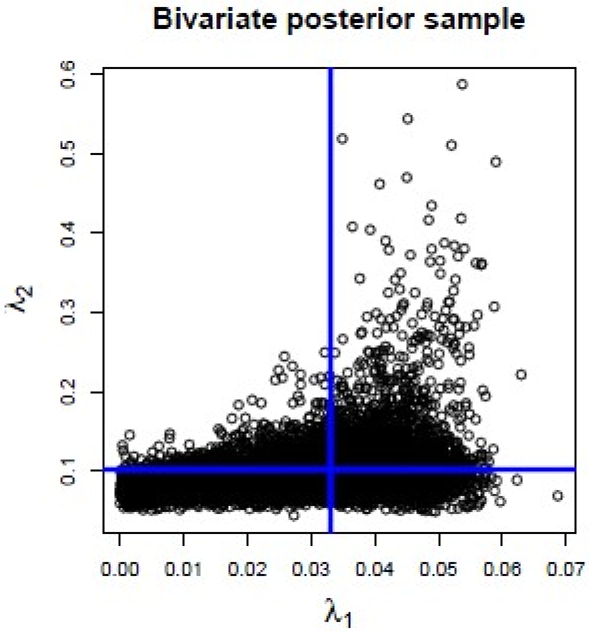
The authors compute these values using the inner strand to inform a prior for the outer strand.

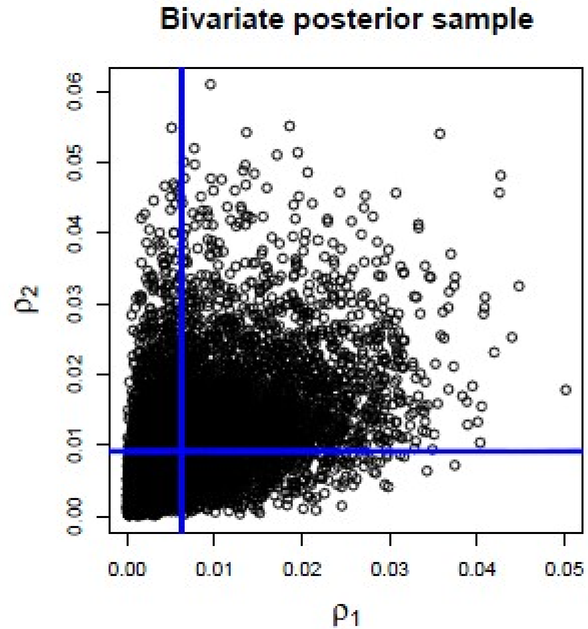
$$\lambda_1 = 0.0208, \quad \lambda_2 = 0.0921$$

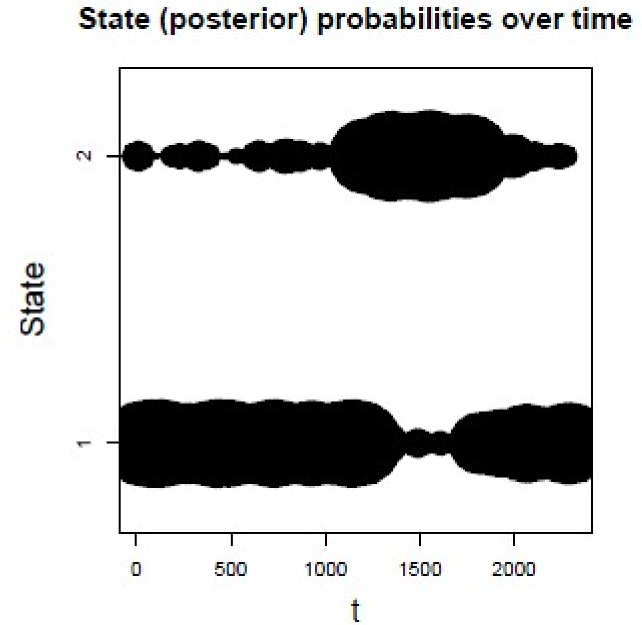
$$\rho_1 = 0.016, \quad \rho_2 = 0.021$$

Results (Chi site data)

Lagging inner strand







$$\hat{\lambda}_1 = 0.033$$

$$CI_{0.95} = (0.003, 0.051)$$

$$\hat{\lambda}_2 = 0.102$$

$$CI_{0.95} = (0.064, 0.180)$$

$$\hat{\rho}_1 = 0.006$$

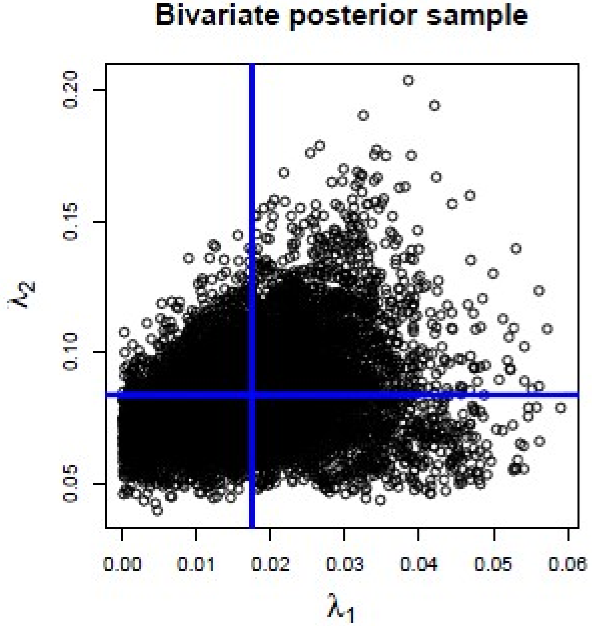
$$CI_{0.95} = (0.0002, 0.0258)$$

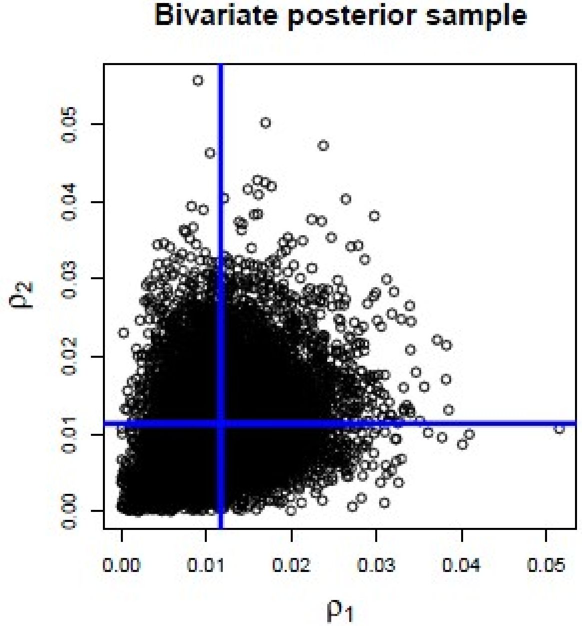
$$\hat{\rho}_2 = 0.009$$

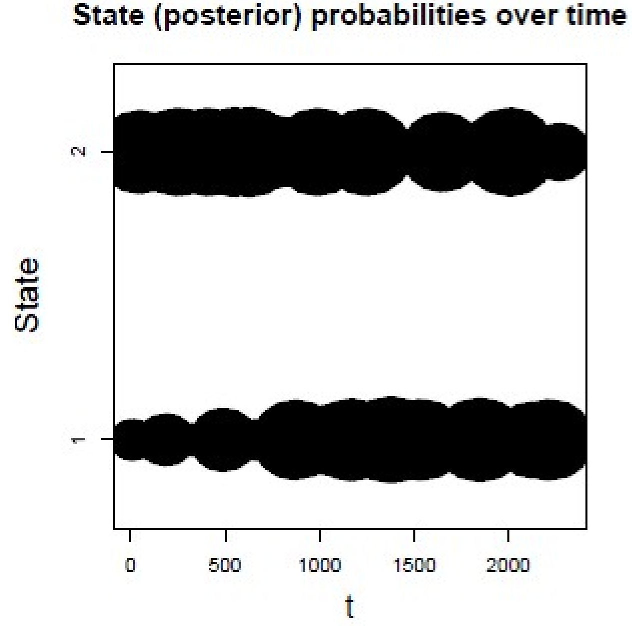
$$CI_{0.95} = (0.001, 0.031)$$

Results (Chi site data)

Lagging outer strand







$$\hat{\lambda}_1 = 0.017$$
$$CI_{0.95} = (0.002, 0.039)$$
$$\hat{\lambda}_2 = 0.083$$
$$CI_{0.95} = (0.055, 0.126)$$

$$\hat{\rho}_1 = 0.012$$
$$CI_{0.95} = (0.003, 0.024)$$
$$\hat{\rho}_2 = 0.011$$
$$CI_{0.95} = (0.002, 0.026)$$

Conclusions

	F&S	Inner	Outer
λ_1	0.021	0.033	0.017
λ_2	0.092	0.102	0.089
ρ_1	0.016	0.006	0.012
ρ_2	0.021	0.009	0.011

In many cases, my implementation produces good results, but it is quite possible to break it.

With respect to the Chi site data, my results approach those of the authors.

Differences in the lambdas are accountable to their using the inner strand to inform the prior of the outer strand, blending their values.

Differences among the rhos are accountable to the instability of convergence for rhos, as I witnessed and was noted in the original paper.

Thanks

Mike Karcher