Bayesian Nonparametric Hidden Markov Models with application to the analysis of copy-number-variation in mammalian genomes

C. Yau[*], O. Papaspiliopoulos[†], G. O. Roberts[‡] and C. Holmes[*]

Abstract

We consider the development of Bayesian Nonparametric methods for product partition models such as Hidden Markov Models and change point models. Our approach uses a Mixture of Dirichlet Process (MDP) model for the unknown sampling distribution (likelihood) for the observations arising in each state and a computationally efficient data augmentation scheme to aid inference. The method uses novel MCMC methodology which combines recent retrospective sampling methods with the use of slice sampler variables. The methodology is computationally efficient, both in terms of MCMC mixing properties, and robustness to the length of the time series being investigated. Moreover, the method is easy to implement requiring little or no user-interaction. We apply our methodology to the analysis of genomic copy number variation.

Keywords : Retrospective sampling, block Gibbs sampler, local/global clustering, partition models, partial exchangeability

1 Introduction

Hidden Markov Models and other conditional Product Partition Models such as change point models or spatial tessellation processes form an important class of statistical regression methods dating back to Baum (1966); Barry & Hartigan (1992). Here we consider Bayesian nonparametric extensions where the sampling density (likelihood) within a state or partition is given by a Mixture of Dirichlet Process (Antoniak, 1974; Escobar, 1988).

Conventional constructions of the MDP make inference extremely challenging computationally due to the joint dependence structure induced on the observations. We develop a data augmenta-

[*] Department of Statistics, University of Oxford, yau@stats.ox.ac.uk,cholmes@stats.ox.ac.uk

[†] Department of Economics, UPF, omiros.papaspiliopoulos@upf.edu

[‡] Department of Statistics, Warwick University, Gareth.O.Roberts@warwick.ac.uk

tion scheme based on the retrospective simulation work (Papaspiliopoulos & Roberts, 2008) which alleviates this problem and facilitates computationally efficient inference, by inducing partial exchangeability of observations within states. This allows for example for the forward-backward sampling and marginal likelihood sampling of state transition paths in an HMM.

Our work here is motivated by the problem of analysing of genomic copy number variation in mammalian genomes (Colella et al., 2007). This is a challenging and important scientific problem in genetics, typified by series of observations of length $\mathcal{O}(10^5)$. In developing our methodology therefore, we have paid close attention to ensure that methods scale well with the size of the data. Moreover, our approach gives good MCMC mixing properties and needs little or no algorithm tuning.

Research on Bayesian semi-parametric modelling using Diricihlet mixtures is now widespread throught the statical literature (Müller et al., 1996; Gelfand & Kottas, 2003; Müller et al., 2005; Quintana & Iglesias, 2003; Burr & Doss, 2005; Teh et al., 2006; Griffin & Steel, 2007, 2004; Rodriguez et al., 2008; B.Dunson, 2005; Dunson et al., 2007) Inference for Dirichlet mixture models has been made feasible since the seminal development of Gibbs sampling techniques in Escobar (1988). This work constructed a *marginal* algorithm where the DP itself is analytically integrated out (see also Liu, 1996; Green & Richardson, 2001; Jain & Neal, 2004). The marginal method is more complicated to implement for non-conjugate models (though see MacEachern & Müller, 1998; Neal, 2000).

The alternative (and in principle more flexible) methodology is the *conditional method*, which does not require analytical integration of the DP. This approach was suggested in Ishwaran & Zarepour (2000); Ishwaran & James (2001, 2003) where finite-dimensional truncations are employed to circumvent the impossibe task of storing the entire Dirichlet process state (which would require infinite storage capacity). In addition to its flexibility, a major advantage of the conditional approach is that in principle it allows inference for the latent random measure P. The requirement to use finite truncations of the DP was removed in recent work (Papaspiliopoulos & Roberts, 2008). In this paper we shall essentially generalise the approach of this paper to our HMM-MDP context. Furthermore, we shall introduce a further innovation using the slice sampler construction of Walker (2007).

The paper in structured as follows. The motivating genetic problem is introduced in detail in Subsection 1.1. The HMM-MDP model is defined in Section 2 while the corresponding computational methodology is described in Section 3. The different models and methods are tested and compared in Section 4 on various simulated data sets. The genomic copy number variation analysis is presented in Section 5, and brief conclusions are given in Section 6.

1.1 Motivating Application

The development of the Bayesian nonparametric HMM reported here was motivated by on-going work by two of the authors in the analysis of genomic copy number variation (CNV) (see Colella

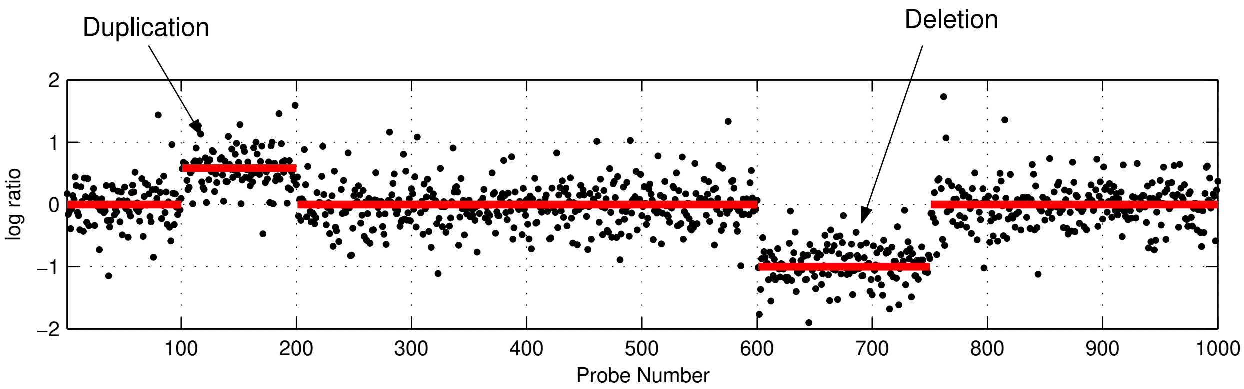

Figure 1: Example Array CGH dataset. This data sets shows a copy number gain (duplication) and a copy number loss (deletion) which are characterised by relative upward and downward shifts in the log intensity ratio respectively.

et al. (2007)). Copy number variants are regions of the genome that can occur at variable copy number in the population. In diploid organisms, such as humans, somatic cells normally contain two copies of each gene, one inherited from each parent. However, abnormalities during the process of DNA replication and synthesis can lead to the loss or gain of DNA fragments, leading to variable gene copy numbers that may initiate or promote disease conditions. For example, the loss or gain of a number of tumor suppressor genes and oncogenes are known to promote the initiation and growth of cancers.

This has been enabled by microarray technology that has enabled copy number variation across the genome to be routinely profiled using array comparative genomic hybridisation (aCGH) methods. These technologies allow DNA copy number to be measure at millions of genomic locations simultaneously allowing copy number variants to be mapped with high resolution. Copy number variation discovery, as a statistical problem, essentially amounts to detecting segmental changes in the mean levels of the DNA hybridisation intensity along the genome (see Figure 1). However, these measurements are extremely sensitive to variations in DNA quality, DNA quantity and instrumental noise and this has lead to the development of a number of statistical methods for data analysis.

One popular approach for tackling this problem utilises Hidden Markov Models where the hidden states correspond to the unobserved copy number states at each probe location, and the observed data are the hybridisation intensity measurements from the microarrays (see Shah et al. (2006); Marioni et al. (2006); Colella et al. (2007); Stjernqvist et al. (2007); Andersson et al. (2008)). Typically the distributions of the observations are assumed to be Gaussian or, in order to add robustness, a mixture of two Gaussians or a Gaussian and uniform distribution, where the second mixture component acts to capture outliers such as in Shah et al. (2006) and Colella et al. (2007). However, many data sets contain non-Gaussian noise distributions on the measurements, as pointed out in Hu et al. (2007), particularly if the experimental conditions are not ideal. As a consequence, existing methods can be extremely sensitive to outliers, skewness or heavy tails in the actual noise process that might lead to large numbers of false copy number variants being detected. As genomic technologies evolve from being pure research tools to diagnostic devices, more robust techniques are required. Bayesian nonparametrics offers an attractive solution to these problems and lead us to investigate the models we describe here.

2 HMM-MDP model formulation

The observed data will be a realization of a stochastic process $\{y_t\}_{t=1}^T$. The marginal distribution and the dependence structure in the process are specified hierarchically and semi-parametrically. Let $f(y|m,z)$ be a density with parameters m and z; $\{s_t\}_{t=1}^T$ be a Markov chain with discrete state-space $\mathcal{S}=\{1,\ldots,n\}$, transition matrix $\Pi=[\pi_{i,j}]_{i,j\in\mathcal{S}}$ and initial distribution π_0; H_θ be a distribution indexed by some parameters θ, and $\alpha>0$. Then, the model is specified hierarchically as follows:

$$\begin{aligned}
y_t \mid s_t, k_t, \boldsymbol{m}, \boldsymbol{z} &\sim f(y_t|m_{s_t}, z_{k_t})\,,\; t=1,\ldots,T\\
P(s_t=i \mid s_{t-1}=j) &= \pi_{i,j}\,, i,j\in\mathcal{S}\\
p(k_t, u_t \mid \boldsymbol{w}) &= \sum_{j:w_j>u_t}\delta_j(\cdot)=\sum_{j=1}^{\infty}1[u_t<w_j]\delta_j(\cdot)\\
z_j \mid \theta &\sim H_\theta\,, j\geq 1\\
w_1=v_1,\; w_j &= v_j\prod_{i=1}^{j-1}(1-v_i), j\geq 2\\
v_j &\sim Be(1,\alpha)\,, j\geq 1\,,
\end{aligned} \tag{1}$$

where $\boldsymbol{m}=\{m_j, j\in\mathcal{S}\}$, $\boldsymbol{s}=(s_1,\ldots,s_T)$, $\boldsymbol{y}=(y_1,\ldots,y_T)$, $\boldsymbol{u}=(u_1,\ldots,u_T)$, $\boldsymbol{k}=(k_1,\ldots,k_T)$, $\boldsymbol{w}=(w_1,w_2,\ldots)$, $\boldsymbol{v}=(v_1,v_2,\ldots)$, $\boldsymbol{z}=(z_1,z_2,\ldots)$ and $\delta_x(\cdot)$ denotes the Dirac delta measure centred at x.

The model has two characterising features, structural changes in time and flexible sampling distribution at each regime. The structural changes are induced by the hidden Markov model (HMM), $\{m_{s_t}\}_{t=1}^T$. The conditional distribution of y given the HMM state is specified as a mixture model in which $f(y\mid m,z)$ is mixed with respect to a random discrete probability measure $P(\mathrm{d}z)$. The last four lines in the hierarchy identify P with the Dirichlet process prior (DPP) with base measure H_θ and the concentration parameter α. Such mixture models are known as mixtures of Dirichlet process (MDP).

We have chosen a particular representation for the Dirichlet process prior (DPP) in terms of the allocation variables $\boldsymbol{k}$, the stick-breaking weights $\boldsymbol{v}$, the mixture parameters $\boldsymbol{z}$ and the auxiliary variables $\boldsymbol{u}$. Note that $\boldsymbol{w}$ is a transformation of $\boldsymbol{v}$, hence we will $\boldsymbol{w}$ and $\boldsymbol{v}$ interchangeably depending on the context. The representation of the DPP in terms of only $\boldsymbol{k}$, $\boldsymbol{v}$ and $\boldsymbol{z}$ (that is where $\boldsymbol{u}$ is marginalised out) is well known and has been used in hierarchical modelling among others by Ishwaran & James (2001); Papaspiliopoulos & Roberts (2008). According to this specification,

$$p(k_t \mid \boldsymbol{w}) = \sum_{j=1}^{\infty} w_j \delta_j(\cdot)\,. \tag{2}$$

Following a recent approach by Walker (2007) we augment the parameter space with further auxiliary (slice) variables $\boldsymbol{u}$ and specify a joint distribution of (k_t, u_t) in (1). Note that conditionally on $\boldsymbol{w}$ the pairs (k_t, u_t) are independent over t. The marginal for k_t implied from this joint distribution is clearly (2). Expression (1) follows from a standard representation of an arbitrary random variable k with density p as a marginal of a pair (k, u) uniformly distributed under the curve p. When p is unimodal the representation coincides with Khinchine's theorem (see Section 6.2 of Devroye, 1986). The reason why we prefer the augmented representation in terms of $(\boldsymbol{k}, \boldsymbol{v}, \boldsymbol{z}, \boldsymbol{u})$ to the marginal representation in terms of $(\boldsymbol{k}, \boldsymbol{v}, \boldsymbol{z})$ will be fully appreciated in Section 3.

Due to its structure the model will be called an HMM-MDP model. From a different viewpoint, we deal with a model with two levels of clustering for $\boldsymbol{y}$, a temporally persisting (local) clustering induced by the HMM and represented by the labels of $\boldsymbol{s}$, and a global clustering induced by the Dirichlet process and represented by the labels of $\boldsymbol{k}$. A specific instance of the model is obtained when $y_t \in \Re$, f is the Gaussian density with mean $m + \mu$ and variance σ^2, $z = (\mu, \sigma^2) \in \Re \times \Re_+$, and H_θ is a $N(0, \gamma) \times IG(a, b)$ product measure with hyperparameters $\theta = (\gamma, a, b)$. Then, according to this model, the mean $E(y_t \mid \boldsymbol{s}, \boldsymbol{m}) = m_t$ is a slowly varying random function driven by the HMM and the distribution of the residuals $y_t - m_t$ is a Gaussian MDP.

Section 5 gives an interpretation of $\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{m}$, n and Π in the context of the ROMA experiment for the study of copy number variation in the genome. In that context there exists reliable prior knowledge which allows us to treat n, Π and $\boldsymbol{m}$ as known. Hence, in the sequel we will consider these parameters as fixed and concentrate on inference for the remaining components of the hierarchical model using fixed hyperparameter values.

The model and the computational methodology we introduce, extend straightforwardly to the more general class of stick-breaking priors for P, obtained by generalising the beta distribution on the final stage of the hierarchy.

3 Simulation methodology

Our primary computational target is the exploration of the posterior distribution of $(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}, \boldsymbol{k}, \alpha)$ by Markov chain Monte Carlo. Note that $\boldsymbol{w}$ is simply a function of $\boldsymbol{v}$, hence it can be recovered from the algorithmic output. We want the computational methodology for HMM-MDP to meet three principal requirements. The model we introduce in Section 2 is targeted to uncover structural changes in long time series (T can be of $\mathcal{O}(10^5)$). Hence, the first requirement is that the algorithmic time scales well with T. Second, the algorithm should not get trapped around minor modes which correspond to confounding of local with global clustering. Informally, we would like to make moves in the high probability region of HMM configurations and then use the residuals to fit the MDP component. And third, we would like the algorithm to require as little human intervention as possible (hence avoid having to tune algorithmic parameters). Such simulation methods would allow the routine analysis of massive data sets from Array CGH and SNP genotyping platforms where it is now routine to perform microarray experiments that can generate millions of observations per sample with populations involving many thousands of individuals.

This section develops an appropriate methodology and shows that it achieves these three goals. Further empirical evidence is provided in Section 4.2. The methodology we develop has two important by-products which have interest outside the scope of this paper. The first is a theoretical result (Proposition 1 and its proof in Appendix 1) about the conditional independence structure of $\{y_t\}_{t=1}^T$, and the second is a novel algorithm for MDP posterior simulation. The rest of the Section is structured as follows. Section 3.1 outlines the main algorithm, part of which is a novel scheme for MDP posterior simulation. Section 3.2 discusses a variety of possible alternative schemes and argues why they would lead to failings in some of the three requirements we have specified.

3.1 Block Gibbs sampling for HMM-MDP

We will sample from the joint posterior distribution of $(\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}, \boldsymbol{k}, \alpha)$ by block Gibbs sampling according to the following conditional distributions:

1. $[\boldsymbol{s} \mid \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}]$
2. $[\boldsymbol{k} \mid \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}]$
3. $[\boldsymbol{v}, \boldsymbol{u} \mid \boldsymbol{k}, \alpha]$
4. $[\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{m}]$
5. $[\alpha \mid \boldsymbol{k}]$.

For convenience we will refer to this as the HMM-MDP algorithm. Steps 1 and 2 correspond to a joint update of $\boldsymbol{s}$ and $\boldsymbol{k}$, by first drawing $\boldsymbol{s}$ from $[\boldsymbol{s} \mid \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}]$ and subsequently $\boldsymbol{k}$ from $[\boldsymbol{k} \mid \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}]$. Hence, we integrate out the global allocation variables $\boldsymbol{k}$ in the update of the local allocation variables $\boldsymbol{s}$. As a result the algorithm does not get trapped in secondary modes which correspond to mis-classification of consecutive data to Dirichlet mixture components. Additionally, Step 1 can be seen as an update of the HMM component, whereas Steps 2-5 constitute an update of the MDP component. Thus, we consider each type of update separately.

HMM update

We can simulate exactly from $[\boldsymbol{s} \mid \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}]$ using a standard forward filtering/backward sampling algorithm (see for example Cappe et al. (2005)). This is facilitated by the following key result which is proved in Appendix 1.

Proposition 1. *The conditional distribution* $[\boldsymbol{s} \mid \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z}]$ *is the posterior distribution of a hidden Markov chain* $s_t, 1 \leq t \leq T$, *with state space* $\mathcal{S}$, *transition matrix* Π, *initial distribution* π_0, *and conditional independent observations* y_t *with conditional density,*

$$p_t(y_t \mid s_t, u_t, \boldsymbol{w}) = \sum_{j:w_j > u_t} f(y_t \mid m_{s_t}, z_j).$$

The number of terms involved in the likelihood evaluations is finite almost surely, since there will be a finite number of mixture components with weights $w_j > u^{*(T)} := \inf_{1 \leq t \leq T} u_t$. In particular, Walker (2007) observes that $j > j^{*(T)}$, is a sufficient condition which ensures that $w_j < u_t$, where $j^{*(T)} := \max_{1 \leq t \leq T}\{j_t^*\}$, and j_t^* is the smallest l such that $\sum_{j=1}^{l} w_j > 1 - u_t$. To see this, note that $\sum_{k \geq j} w_k < u$ implies that $w_k < u$ for all $k \geq j$. Hence, the number of terms used in the likelihood evaluations is bounded above by $j^{*(T)}$. Additionally, note that we only need partial information about the random measure $(\boldsymbol{z}, \boldsymbol{v})$ to carry out this step: the values of $(v_j, z_j), j \leq j^{*(T)}$ are sufficient to carry out the forward/backward algorithm.

However, $j^{*(T)}$ will typically grow with T. Under the prior distribution, $u^{*(T)} \downarrow 0$ almost surely as $T \to \infty$. Standard properties of the DPP imply that $j^{*(T)} = \mathcal{O}(\log T)$ (see for example Muliere & Tardella, 1998). This relates to the fact that the number of new components generated by the Dirichlet process grows logarithmically with the size of the data (Antoniak, 1974). On the other hand, it is well known that the computational cost of the forward filtering/backward sampling, when the computational cost of evaluating the likelihood is fixed, is $\mathcal{O}(T)$ (and quadratic in the size of the state space). Hence, we expect an overall computational cost $\mathcal{O}(T \log T)$ for the *exact* simulation of the hidden Markov chain in this non-parametric setup.

MDP update

Conditionally on a realisation of $\boldsymbol{s}$, we have an MDP model. Therefore, the algorithm comprised of Steps 2-5 can be seen more generally as a block Gibbs sampler for posterior simulation in an MDP model. According to the terminology of Section 1 we deal with a conditional method for MDP posterior simulation since the random measure $(\boldsymbol{z}, \boldsymbol{v})$ is imputed and explicitly updated.

The algorithm we propose is a synthesis of the retrospective Markov chain Monte Carlo algorithm of Papaspiliopoulos & Roberts (2008) and the slice Gibbs sampler of Walker (2007). The synthesis yields an algorithm which has advantages over both. Additionally, it is particularly appropriate in the context of the HMM-MDP model, as we show in Section 3.2. Simulation experiments with these algorithms are provided in Section 4.1.

We first review briefly the algorithms of Papaspiliopoulos & Roberts (2008) and Walker (2007). The retrospective algorithm works with the parametrisation of the MDP model in terms of $(\boldsymbol{k}, \boldsymbol{v}, \boldsymbol{z})$ (see the discussion in Section 2). Then, it proceeds by Gibbs sampling of $\boldsymbol{k}$, $\boldsymbol{v}$ and $\boldsymbol{z}$ according to their full conditional distributions. Simulation from the conditional distributions of $\boldsymbol{v}$ and $\boldsymbol{z}$ is particularly easy. Specifically, $\boldsymbol{v}$ consists of conditionally independent elements with

$$v_j \mid \boldsymbol{k}, \alpha \sim \text{Be}\left(m_j + 1, T - \sum_{l=1}^{j} m_l + \alpha\right) \text{ for all } j = 1, 2, \ldots\,, \tag{3}$$

where $m_j = \#\{t : k_t = j\}$. Similarly, $\boldsymbol{z}$ consists of conditionally independent elements with

$$z_j \mid \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{m}, \boldsymbol{k} \sim \begin{cases} \prod_{t:k_t=j} f(y_t \mid m_{s_t}, z_j)\pi(z_j \mid \theta) \text{ for all } j : m_j > 0\,, \\ \\ H_\theta, \text{ otherwise} \end{cases} \tag{4}$$

In this expression $\pi(z \mid \theta)$ denotes the Lebesgue density of H_θ. On the other hand, simulation from the conditional distribution of $\boldsymbol{k}$ is more involved. It follows directly from (2) that conditionally on the rest $\boldsymbol{k}$ consist of conditionally independent elements with

$$p(k_t \mid \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{v}, \boldsymbol{z}) \propto \sum_{j=1}^{\infty} w_j f(y_t \mid m_{s_t}, z_j)\delta_j(\cdot) \tag{5}$$

which has an intractable normalising constant, $\sum_{j=1}^{\infty} w_j f(y_t \mid m_{s_t}, z_j)$. Therefore, direct simulation from this distribution is difficult. Papaspiliopoulos & Roberts (2008) devise a Metropolis-Hastings scheme which resembles an independence sampler and it accepts with probability 1 most of the proposed moves.

The slice Gibbs sampler of Walker (2007) parametrises in terms of $(\boldsymbol{k}, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z})$. Hence, the posterior distribution sampled by the retrospective algorithm is a marginal of the distribution sampled by the slice Gibbs sampler, and the retrospective Gibbs sampler is a collapsed version of the slice Gibbs sampler (modulo the Metropolis-Hastings step in the update of $\boldsymbol{k}$). The slice Gibbs sampler proceeds by Gibbs sampling of $\boldsymbol{k}$,$\boldsymbol{u}$,$\boldsymbol{v}$, and $\boldsymbol{z}$ according to their full conditional distributions. The augmentation of $\boldsymbol{u}$ greatly simplifies the structure of (5), which now becomes

$$p(k_t \mid \boldsymbol{y}, \boldsymbol{s}, \boldsymbol{v}, \boldsymbol{z}) \propto \sum_{j:w_j > u_t} f(y_t \mid m_{s_t}, z_j)\delta_j(\cdot) \tag{6}$$

Note that the distribution has now finite support and the normalising constant can be computed. Hence this distribution can be simulated by the inverse CDF method by computing a number of terms no more than $j^{*(t)}$ for each t. Also $\boldsymbol{u}$ also consists of conditional independent elements with $u_t \sim Uni(0, w_{k_t})$. On the other hand, the conditioning on $\boldsymbol{u}$ creates global dependence on the v_js, which their distribution is given by (3) under the constraint $w_j > u_t\,, \forall t = 1, \dots, T$. The easiest way to simulate from this constrained distribution is by single site Gibbs sampling of the v_js. This single-site Gibbs sampling tends to be slowly mixing and deteriorating with T.

Our method updates $\boldsymbol{u}$ and $\boldsymbol{v}$ in a single block, by first updating $\boldsymbol{v}$ from its marginal (with respect to $\boldsymbol{u}$) according to (3) and consequently $\boldsymbol{u}$ conditionally on $\boldsymbol{v}$ as described above. This scheme is feasible due to the nested structure of the parametrisations of the retrospective and the slice Gibbs algorithms. The update of $\boldsymbol{k}$ is done as in the slice Gibbs sampler, and the update of $\boldsymbol{z}$ as described earlier. When a gamma prior is used for α, its conditional distribution given $\boldsymbol{k}$ and marginal with respect to the rest is a mixture of gamma distributions and can be simulated as described in Escobar & West (1995). The algorithm can easily incorporate the label-switching moves discussed in Section 3.4 of Papaspiliopoulos & Roberts (2008) (where the problem of multi-modality for conditional methods for MDP posterior simulation is discussed in detail). FORTRAN 77 and MATLAB code are available on request by the authors.

3.2 Comparison with alternative schemes

There are other Gibbs sampling schemes which can be used to fit the HMM-MDP to the observed data. They are based on alternative parametrisations of the DPP. In this section we argue in favour of the approach followed in the previous section and show that other schemes lead to difficulties in Step 1 of the HMM-MDP algorithm, i.e. the step which updates the $\boldsymbol{s}$ by integrating out $\boldsymbol{k}$. Section 4.2 complements our arguments by demonstrations on simulated data.

Section 1 described two main categories of Gibbs sampling algorithms: the marginal and the conditional. So far we have considered conditional algorithms, which impute and update the random measure $(\boldsymbol{z}, \boldsymbol{v})$ (using retrospective sampling). We first argue why we prefer conditional methods in this context. The marginal methods integrate out analytically the random weights $\boldsymbol{w}$ from the model and update the rest of the variables. A result of this marginalisation is that the allocation variables $\boldsymbol{k}$ are not apriori independent, but they have an exchangeable dependence structure. A consequence of this prior dependence is that in this scheme, it becomes infeasible to integrate out the global allocation variables $\boldsymbol{k}$, during the update of the local allocation variables $\boldsymbol{s}$. Therefore, a marginal augmentation scheme in the HMM-MDP context is likely to get trapped to minor modes in the posterior distribution which correspond to mis-classification of global and local clusters. Indeed, this is illustrated in the simulation study of Section 4.2.

A competing conditional algorithm is obtained by integrating out $\boldsymbol{u}$ from the model and working with the parametrisation of the DPP in terms of $(\boldsymbol{k}, \boldsymbol{v}, \boldsymbol{z})$ as in Papaspiliopoulos & Roberts (2008). A problem with this approach arises again in the implementation of Step 1 of the HMM-MDP algorithm. Working as in Appendix 1, it is easy to see that a version of Proposition 1 still holds, however the conditional density corresponding to each observation y_t is now

$$p_t(y_t \mid s_t, \boldsymbol{w}) = \sum_{j=1}^{\infty} w_j f(y_t \mid m_{s_t}, z_j).$$

Therefore, the likelihoods associated with each HMM state are not directly computable due to the infinite summation. Nevertheless, direct simulation at Step 1 is still feasible even though we deal with an HMM with intractable likelihood functions.

For example, if $f(y \mid m, z)$ is bounded in z,

$$B(m, y) := \sup_z f(y \mid m, z) \leq \infty$$

then the following upper bound is available for the likelihood for any integer M:

$$\tilde{p}_t(y_t \mid s_t, \boldsymbol{w}) = \sum_{j=1}^{M} w_j f(y_t \mid m_{s_t}, z_j) + B(m_{s_t}, y_t)\left(1 - \sum_{j=1}^{M} w_j\right).$$

In this setting, increasing M improves the approximation, and $\tilde{p}_t \downarrow p_t$ as $M \to \infty$. These upper bounds can be used to simulate $\boldsymbol{s}$ from $[\boldsymbol{s} \mid \boldsymbol{y}, \boldsymbol{v}, \boldsymbol{z}]$ by rejection sampling. The proposals are generated using a forward filtering/backward sampling algorithm using $\tilde{p}_t$ as the likelihood for each time t, and are accepted with probability

$$A(M, T, \boldsymbol{z}, \boldsymbol{w}) := \prod_{t=1}^{T} \frac{p_t(y_t \mid s_t, \boldsymbol{w})}{\tilde{p}_t(y_t \mid s_t, \boldsymbol{w})}.$$

Note that for any fixed M the acceptance probability will typically go to 0 exponentially quickly as $T \to \infty$. It can be shown that increasing M with T at any rate is sufficient to ensure that $A(M, T, \boldsymbol{z}, \boldsymbol{w})$ is bounded away from 0 almost surely (with respect to the prior measure on $(\boldsymbol{w}, \boldsymbol{z})$). However, to ensure that $1/A(M, T, \boldsymbol{z}, \boldsymbol{w})$ has a finite first moment (with respect to the prior measure on $(\boldsymbol{w}, \boldsymbol{z})$), i.e. the expected number of trials until first acceptance is finite, M needs to increase as $\mathcal{O}(T)$. This determines the cost for each likelihood calculation, and since to carry out the forward/backward algorithm we need an $\mathcal{O}(T)$ such calculations, we have an overall $\mathcal{O}(T^2)$ cost for the algorithm. This is clearly undesirable. Although we have only given a heuristic argument, a formal proof is feasible.

4 Simulation experiments

In this section, we compare rival MCMC schemes as described in Section 3. We begin with a detailed comparison of methodology for the MDP update in Subsection 4.1, followed by a comparison of the entire methods on simulated data sets.

4.1 Comparison of MDP posterior sampling schemes

We first carry out a comparison of different schemes for performing the "MDP update". We consider this part of the simulation algorithm separately since it can be used in various contexts which involve posterior simulation of stick-breaking processes. We have considered three main algorithm to carry out this step: the retrospective MCMC of Papaspiliopoulos & Roberts (2008) with label-switching moves (R), the slice sampler of Walker (2007) (SL) and the block Gibbs algorithm (BGS) introduced in this paper. We also consider the block Gibbs sampler with added label-switching moves (BGS/L).

For simplicity, and without compromising the comparison, we take s_t to be constant in time. We design the simulation study according to Papaspiliopoulos & Roberts (2008), where the retrospective MCMC is compared with various other (marginal) algorithms. We test the algorithms on the 'bimod 100' ('bimod 1000') dataset of which consists of 100 (1000) draws y_t from the bimodal mixture, $0.5N(-1, 0.5^2) + 0.5N(1, 0.5^2)$. We fit the non-conjugate Gaussian MDP model discussed in Section 2. We take $\alpha = 1$ and use the data to set values for the hyperparameters θ (see Section 4 of Papaspiliopoulos & Roberts (2008)).

Figure 2 summarizes the comparison between the competing approaches. We show autocorrelation plots for three different functions in the parameter space: the number of clusters, the deviance of the fit (see Papaspiliopoulos & Roberts (2008) for its calculation) and z_{k_3}. Simulation experiments suggest that the computational times per iteration (in "stationarity") of "R" and "SL" are similar, and about 50-60% higher than those of "BGS" and "BGS/L". Additionally, the computational times for all algorithms grow linearly with T, the size of the data[1]. The simulation experiment suggests that the retrospective MCMC is mixing faster than the other algorithms, and that the block Gibbs sampler (with or without label-switching moves) is more efficient than the slice Gibbs sampler.

4.2 Testing the methodology on simulated datasets

4.2.1 Data

We simulated three datasets based on the *lepto 1000* and *bimod 1000* data sets used by Green & Richardson (2001) and (Papaspiliopoulos & Roberts, 2008) and a trimodal data set (which we shall call *trimod 1000*) used by Walker (2007). The data was generated according to the following

[1] Note that naive implementations of "R" can lead to $\mathcal{O}(T^2)$ costs.

scheme:

$$\begin{aligned} y_t &\sim N(m_{0,s_t} + \mu_{k_t}, 1/\lambda_{k_t}), \\ p(s_t = i | s_{t-1} = j) &= \pi_{i,j}, \\ p(s_1 = s) &= \pi_0(s), \\ k_t &\sim \sum_{j=1}^{K} w_j \delta_j(\cdot) \end{aligned}$$

where $x_t \in \{0, 1\}$, the prior state distribution $\pi_0 = (1/2, 1/2)$ and the transition matrix Π is of the form,

$$\Pi = \begin{pmatrix} 1-\rho & \rho \\ \rho & 1-\rho \end{pmatrix}$$

where the transition probability $\rho = 0.05$ and $T = 1000$ for the simulations. Additional simulation parameters are detailed in Table 1.

Simulation Parameter	*lepto 1000*	*bimod 1000*	*trimod 1000*
K	2	2	3
m_0	$(0, 0.3)$	$(0, 1)$	$(0, 1)$
w	$(0.67, 0.33)$	$(0.5, 0.5)$	$(1/3, 1/3, 1/3)$
μ	$(0, 0.3)$	$(-1, 1)$	$(-4, 0, 8)$
λ	$(1, 1/0.25^2)$	$(1/0.5^2, 1/0.5^2)$	$(1, 1, 1)$

Table 1: Simulation Parameters.

4.2.2 Prior Specification

We used Normal priors for the mixture centers $\mu_k \sim N(0, 1)$ and Gamma distributed priors for the precisions $\lambda_k \sim Ga(1, 1)$ and fixed the concentration parameter of the Dirichlet Process $\alpha = 1$ for all simulations. The prior distribution of m was set to be a Normal distribution with mean m_0 and precision $\omega = 100$.

4.2.3 Posterior Inference

We applied five different Gibbs sampling approaches. The first is a marginal method based on Algorithm 5 from Neal (2000) that updates (s_t, k_t) from its conditional distribution $\pi(s_t, k_t|\cdot)$ according to the following scheme:

1. Draw a candidate, k_t^* from the conditional prior for k_i where the conditional prior is given by:

$$p(k_t^* = j | k_{-t}) \propto \begin{cases} \frac{n_{-t,k}}{n-1+\alpha}, & \text{if } k_t = j \text{ for some } t \\ \frac{\alpha}{n-1+\alpha}, & \text{if } k_t \neq j \text{ for all } t \end{cases}$$

where $n_{-t,k}$ is the number of data points allocated to the k-th component but not including the t-th data point.

2. Draw a candidate state, s_t^* from the conditional prior distribution $p(s_t|s_{t-1}, s_{t+1})$.

3. Accept (s_t^*, k_t^*) with probability $\alpha\{(s_t^*, k_t^*), (s_t, k_t)\}$ where

$$\alpha\{(s_t^*, k_t^*), (s_t, k_t)\} = \min\left[1, \frac{\pi_{s_{t-1},s_t^*}\pi_{s_t^*,s_{t+1}}f(y_t|m_{s_t^*}, z_{k_t^*})}{\pi_{x_{t-1},x_t}\pi_{s_t,s_{t+1}}f(y_t|m_{s_t}, z_{k_t})}\right]$$

otherwise leave (s_t, k_t) unchanged.

We also analysed the datasets using two variations of both the Slice and Block Gibbs Sampling approaches. In the first approach, we sample from the conditional distributions $\pi(s_t, k_t|\cdot)$:

1. Sample s_t from $p(s_t|\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{y})$, $t = 1, \ldots, T$.

2. Sample k_t from $p(k_t|\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{z}, \boldsymbol{y})$, $t = 1, \ldots, T$.

We denote these as the Slice Samplers and Block Gibbs Samplers *with local updates*. The second method uses forward-backward sampling to simulate $\pi(\boldsymbol{s}|\cdot)$:

1. Sample $\boldsymbol{s}$ from $p(\boldsymbol{s}|\boldsymbol{u}, \boldsymbol{z}, \boldsymbol{y})$ using the forward filtering-backward sampling method.

2. Sample k_t from $p(k_t|\boldsymbol{s}, \boldsymbol{u}, \boldsymbol{z}, \boldsymbol{y})$, $t = 1, \ldots, T$.

We denote these as the Slice Samplers and Block Gibbs Samplers *with forward-backward updates.*

For all the sampling methods, we generated 20,000 sweeps (one sweep being equivalent to an update of all T allocation and state variables) and discarded the first 10,000 as burn-in. We employed the following Gibbs updates for the mixture component parameters, for $j = 1, \ldots, k^*$,

$$\mu_j \sim N\left(\frac{\xi_j\lambda_j}{n_j\lambda_j + 1}, \frac{1}{n_j\lambda_j + 1}\right),$$
$$\lambda_j \sim Ga(1 + n_j/2, 1 + d_j/2),$$

where $k^* = \max_t\{k_t\}$, $\xi_j = \sum_{t:k_t=j}(y_t - m_{s_i})$, $n_j = \sum_{t:k_t=j} 1$ and $d_j = \sum_{t:k_t=j}(y_i - m_{s_i})^2$. The mean levels for each hidden state are updated using,

$$m_i \sim N\left(\frac{S_{\lambda,y} + \omega m_{i,0}}{S_\lambda + \omega}, \frac{1}{S_\lambda + \omega}\right)$$

where $S_\lambda = \sum_{t:s_t=i} \lambda_{k_t}$ and $S_{\lambda,y} = \sum_{t:s_t=i} \lambda_{k_t}(y_t - \mu_{k_t})$.

4.2.4 Results

Figure 3 gives autocorrelation times for the three Gibbs Samplers on the simulated datasets. In terms of updating the hidden states $\boldsymbol{s}$, the use of forward-backward sampling gives a distinct advantage over the local updates. This replicates previous findings by Scott (2002) who showed that forward-backward Gibbs sampling for Hidden Markov Models mix faster than using local updates as it is difficult to move from one configuration of $\boldsymbol{s}$ to another configuration of entirely different structure using local updates only. This result motivates the use of the conditional augmentation structure adopted here as it would otherwise be impossible to perform efficient forward-backward sampling of the hidden states $\boldsymbol{s}$.

In Figure 5 we plotted the simulation output of (v_1, v_2) for the Slice Sampler and the Block Gibbs Sampler (using forward-backward updates). The mixing of the Block Gibbs Sampler is considerably better than the Slice Sampler. The Block Gibbs Sampler appears able to explore different modes in the posterior distribution of v for each of the three datasets whereas the Slice Sampler tends to get fixated to one mode.

5 ROMA data analysis

We analysed the mouse ROMA dataset from Lakshmi et al. (2006) using the MDP-HMM and a standard HMM with Gaussian observations (G-HMM). The data set consists of approximately $84,000$ probes measurements from a DNA sample derived from a tumour generated in a mouse model of liver cancer compared to normal (non-tumour) DNA derived from the parent mouse.

Correspondence between experimental setup and model: we think of y_t representing the log-hybridisation intensity ratio obtained from measurements from the microarray experiment; t denotes the genome order (an index after which the probes are sorted by genomic position); s_t denotes the unobserved copy number state in the case subject (e.g. 0, 1, 2, 3, etc); m_j is the corresponding mean level for the jth copy number state.

5.1 Prior Specification

We assumed a three-state HMM with fixed mean levels $m = (-0.58, 0, 0.52)$ and a transition probability of $\rho = 0.01$. We used normal priors for the mixture centres $\mu_k \sim N(0, 1)$ and Gamma distributed priors for the precisions $\lambda_k \sim Ga(1, 1)$ and fixed the concentration parameter of the Dirichlet Process $\alpha = 1$ for all simulations.

5.2 Posterior Inference

We analysed the mouse ROMA dataset using the MDP-HMM and two additional HMM-based models. The first is a standard HMM model with Gaussian distributed observations (that we should denote as the G-HMM) and the second model uses a mixture of two Gaussians for the observation (which we shall denote as the Robust-HMM or R-HMM). In the R-HMM, the second mixture component has a large variance ($\lambda_2 = 10^2$) to capture outliers and is a strategy used by (Shah et al., 2006) to provide robustness against outliers. These two latter models are representative of currently available HMM-based methods for analysing aCGH datasets and can be considered to be special cases of the more general MDP-HMM. For the MDP-HMM, we used the Block Gibbs Sampler with forward-backward sampling, whilst for the G-HMM and R-HMM we employed standard forward-backward Gibbs Sampling methods for HMMs with finite Gaussian mixture observation densities.

5.3 Results

Figure 6 shows the analysis of Chromosome 5 for the mouse tumour. The G-HMM, R-HMM and MDP-HMM are both able to identify a deletion found previously in (Lakshmi et al., 2006), however, the G-HMM also identifies many other putative copy number variants. Although, mouse tumours are likely to contain many copy number alteration events, the numbers predicted by the G-HMM are far too high. The R-HMM provides much more conservative and realistic estimates of the number of putative copy number variants in the tumour, however, the MDP-HMM identifies the known deletion only whilst the R-HMM still produces many additional copy number variants whose existence cannot be confirmed. To give an indication of the required computing times for each method, 10^4 iterations of R-HMM and MDP-HMM required 40 and 60 minutes respectively using a MATLAB code, but the execution time for both can be substantially reduced by an implementation in a lower-level programming language (which handles loops more efficiently). The G-HMM required only a couple of minutes, but this time difference compared to the other methods is an artifact of the MATLAB implementation.

In Figure 7 we show a region of Chromosome 3 from the mouse tumour that contains a region consisting of a cluster of mutations known as single nucleotide polymorphisms (SNPs) (as shown in (Lakshmi et al., 2006)). These sequence mutations can disrupt the hybridisation of the genomic DNA fragements on to the microarrays causing unusually high or low (depending on whether the mutation is located on the tumour or reference sample) observed values of the hybridisation intensity ratios. This is because the probes on the microarray are designed to target specific genomic sequences and, if a mutation occurs in the target sequence, the probes will be unable to bind to the DNA. The G-HMM and R-HMM are highly sensitive to the outlier measurements caused by SNPs in this region. This leads to multiple genomic locations in this region being identified as putative copy number variants. In contrast, the MDP-HMM is robust to these effects and correctly calls no copy number alterations in the region.

The explanation for the improved performance of the MDP-HMM for this application is explained by the QQ-plots in Figure 8(d-f). Here, we drew 10,000 samples from the predictive distribution from the G-HMM, R-HMM and MDP-HMM and plotted the quantiles against the empirical quantiles of the data. We see that the G-HMM and R-HMM fails to capture the behaviour of the data in the tails and that the distribution of the data also appears to be asymmetric. Both of these features are pathological for the G-HMM and, though the R-HMM can compensate for heavy tails, it inherently assumes symmetry that is not present in the data.

6 Discussion

This paper has introduced a new methodology for Bayesian semi-parametric time-series analysis. The flexibility of the HMM structure together with the general Dirichlet error distribution suggest that the approach will have many potential applications, particularly for long time-series such as the copy number data analysed in this paper. The results in our genomic example are very promising, and we are already investigating further genetic applications of this work.

Various extensions of the methodology are possible. It is straightforward (a single line change in the code) to allow more general stick-breaking priors, as for example the two-parameter Poisson-Dirichlet process (Ishwaran & James, 2001). It is also simple to allow the joint analysis of various series simultaneously using a hierarchical model. It would be natural to extend the Bayesian analysis to account for uncertainty in the structure and dynamics of the hidden Markov model. In particular, prior structures could be imposed on both n and $\{\Pi_{ij}, 1 \leq i, j \leq n\}$. Furthermore, other product partition models such as CART or changepoint models could similarly be investigated within this MDP error structure context. Further work will explore these extensions.

References

Andersson, R., Bruder, C. E. G., Piotrowski, A., Menzel, U., Nord, H., Sandgren, J., Hvidsten, T. R., de Sthl, T. D., Dumanski, J. P. & Komorowski, J. (2008). A segmental maximum a posteriori approach to genome-wide copy number profiling. *Bioinformatics* 24 751–758. URL `http://dx.doi.org/10.1093/bioinformatics/btn003`.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to bayesian nonparametric problems. *Ann. Statist.* **2**, 1152–74.

Barry, D. & Hartigan, J. A. (1992). Product partition models for change point models. *Annals of Statistics* 20 260–279.

Baum, L. E. (1966). Statistical inference for probabilistic functions of finite state space markov chains. *Annals of Mathematical Statistics* 37 1554–1563.

B.Dunson, D. (2005). Bayesian semiparametric isotonic regression for count data. *Journal of the American Statistical Association* 100 618–627.

Burr, D. & Doss, H. (2005). A Bayesian semiparametric model for random-effects meta-analysis. *J. Am. Statist. Assoc.* **100**, 242–51.

Cappe, ., Moulines, E. & Ryden, T. (2005). *Inference in Hidden Markov Models.* Springer.

Colella, S., Yau, C., Taylor, J. M., Mirza, G., Butler, H., Clouston, P., Bassett, A. S., Seller, A., Holmes, C. C. & Ragoussis, J. (2007). Quantisnp: an objective bayes hidden-markov model to detect and accurately map copy number variation using snp genotyping data. *Nucleic Acids Res* 35 2013–2025. URL `http://dx.doi.org/10.1093/nar/gkm076`.

Devroye, L. (1986). *Non-Uniform Random Variate Generation.* Springer-Verlag.

Dunson, D. B., Pillai, N. & Park, J. H. (2007). Bayesian density regression. *Journal of the Royal Statistical Society, Series B* 69 163–183.

Escobar, M. (1988). Estimating the means of several normal populations by nonparametric estimation of the distribution of the means. *PhD Dissertation*, Department of Statistics, Yale University.

Escobar, M. D. & West, M. (1995). Bayesian density estimation and inference using mixtures. *J. Am. Statist. Assoc* 90 577–88.

Gelfand, A. & Kottas, A. (2003). Bayesian semiparametric regression for median residual life. *Scand. J. Statist.* **30**, 651–65.

Green, P. & Richardson, S. (2001). Modelling heterogeneity with and without the Dirichlet process. *Scand. J. Statist.* 28 355–75.

Griffin, J. & Steel, M. J. F. (2004). Semiparametric bayesian inference for stochastic frontier models. *J. Econometrics* 123 121–152.

Griffin, J. & Steel, M. J. F. (2007). Bayesian non-parametric modelling with the Dirichlet process regression smoother. CRiSM technical report 07-05.

Hu, J., Gao, J.-B., Cao, Y., Bottinger, E. & Zhang, W. (2007). Exploiting noise in array cgh data to improve detection of dna copy number change. *Nucleic Acids Res* 35 e35. URL `http://dx.doi.org/10.1093/nar/gkl730`.

Ishwaran, H. & James, L. (2001). Gibbs sampling methods for stick-breaking priors. *J. Am. Statist. Assoc.* **96**, 161–73.

Ishwaran, H. & James, L. F. (2003). Some further developments for stick-breaking priors: finite and infinite clustering and classification. *Sankhyā, A,* **65**, 577–92.

Ishwaran, H. & Zarepour, M. (2000). Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika* **87**, 371–90.

Jain, S. & Neal, R. M. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *J. Comp. Graph. Statist.* **13**, 158–82.

Lakshmi, B., Hall, I. M., Egan, C., Alexander, J., Leotta, A., Healy, J., Zender, L., Spector, M. S., Xue, W., Lowe, S. W., Wigler, M. & Lucito, R. (2006). Mouse genomic representational oligonucleotide microarray analysis: detection of copy number variations in normal and tumor specimens. *Proc Natl Acad Sci U S A* 103 11234–11239. URL http://dx.doi.org/10.1073/pnas.0602984103.

Liu, J. S. (1996). Nonparametric hierarchical Bayes via sequential imputations. *Ann. Statist.* **24**, 911–30.

MacEachern, S. & Müller, P. (1998). Estimating mixture of Dirichlet process models. *J. Comp. Graph. Statist.* **7**, 223–38.

Marioni, J. C., Thorne, N. P. & Tavar, S. (2006). Biohmm: a heterogeneous hidden markov model for segmenting array cgh data. *Bioinformatics* 22 1144–1146. URL http://dx.doi.org/10.1093/bioinformatics/btl089.

Muliere, P. & Tardella, L. (1998). Approximating distributions of random functionals of Ferguson-Dirichlet priors. *Can. J. Statist.* **26**, 283–97.

Müller, P., Erkanli, A. & West, M. (1996). Bayesian curve fitting using multivariate normal mixtures. *Biometrika* **83**, 67–79.

Müller, P., Rosner, G. L., De Iorio, M. & MacEachern, S. (2005). A nonparametric Bayesian model for inference in related longitudinal studies. *Appl. Statist.* **54**, 611–26.

Neal, R. (2000). Markov chain sampling: Methods for Dirichlet process mixture models. *J. Comp. Graph. Statist.* **9**, 283–97.

Papaspiliopoulos, O. & Roberts, G. O. (2008). Retrospective markov chain monte carlo for dirichlet process hierarchical models. *Biometrika* 95 169–186.

Quintana, F. & Iglesias, P. (2003). Bayesian clustering and product partition models. *J. Roy. Statist. Soc. B* **65**, 557–574.

Rodriguez, A., B.Dunson, D. & Gelfand, A. E. (2008). The nested dirichlet process. *Journal of the American Statistical Association* 103 1131–1144.

Scott, S. (2002). Bayesian Methods for Hidden Markov Models: Recursive Computing in the 21st Century. *Journal of the American Statistical Association* 97 337–351.

Shah, S. P., Xuan, X., DeLeeuw, R. J., Khojasteh, M., Lam, W. L., Ng, R. & Murphy, K. P. (2006). Integrating copy number polymorphisms into array cgh analysis using a robust hmm. *Bioinformatics* 22 e431–e439. URL `http://dx.doi.org/10.1093/bioinformatics/btl238`.

Stjernqvist, S., Rydn, T., Skld, M. & Staaf, J. (2007). Continuous-index hidden markov modelling of array cgh copy number data. *Bioinformatics* 23 1006–1014. URL `http://dx.doi.org/10.1093/bioinformatics/btm059`.

Teh, Y., Jordan, M., Beal, M. & Blei, D. (2006). Hierarchical dirichlet processes. *to appear in J. Amer. Statist. Assoc., available from http://www.cs.princeton.edu/ blei/papers/TehJordanBealBlei2006.pdf.*

Walker, S. (2007). Sampling the dirichlet mixture model with slices. *Comm. Statist. Sim. Comput.* 36 45–54.

Appendix 1: proof of Proposition 1

Proposition 1 follows directly from the following result which shows that the data $\boldsymbol{y}$ conditionally on $(\boldsymbol{s}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{u})$ are independent even when the allocation variables $\boldsymbol{k}$ are integrated out.

$$\begin{aligned} p(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{z}, \boldsymbol{v}, \boldsymbol{u}) &= \sum_{\boldsymbol{k}} p(\boldsymbol{y} \mid \boldsymbol{s}, \boldsymbol{k}, \boldsymbol{z}) p(\boldsymbol{k} \mid \boldsymbol{w}, \boldsymbol{u}) = \sum_{\boldsymbol{k}} \prod_{t=1}^{T} f(y_t \mid m_{s_t}, z_{k_t}) p(k_t \mid u_t, \boldsymbol{w}) \\ &= \prod_{t=1}^{T} \sum_{j=1}^{\infty} 1[u_t < w_j] f(y_t \mid m_{s_t}, z_j) = \prod_{t=1}^{T} \sum_{j: u_t < w_j} f(y_t \mid m_{s_t}, z_j) \end{aligned}$$

The first equality follows by standard marginalisation, where we have used the conditional independence to simplify each of the densities. The second equality follows from the conditional independence of the y_t's and the k_t's given the conditioning variables. We exploit the product structure to exchange the order of the summation and the product to obtain the third equality. The last equality is a re-expression of the previous one.

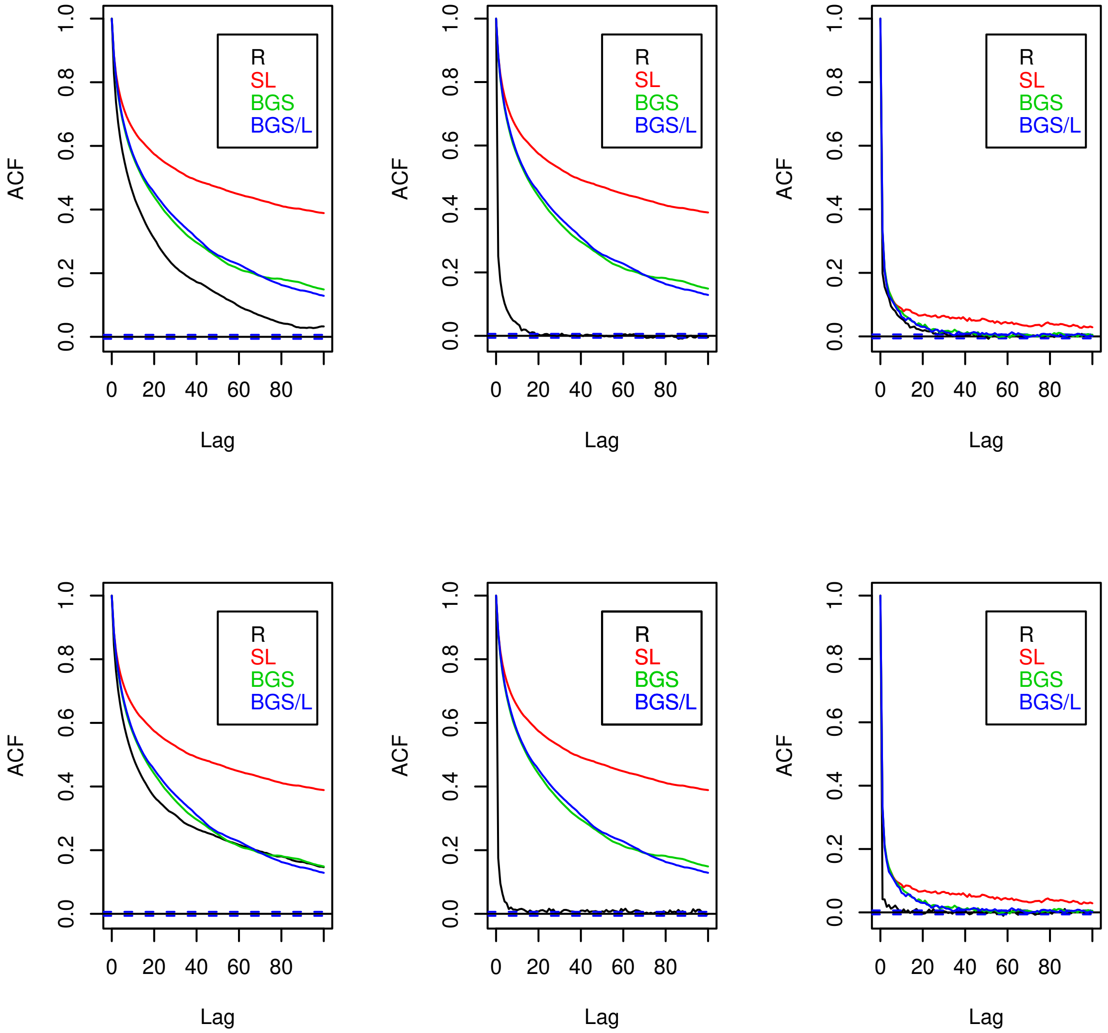

Figure 2: Simulation study of Retrospective MCMC (Retro), the slice sampler (Slice), our new Block Gibbs algorithm (Block), and the Block Gibbs algorithm with label switching moves (Block label). We show autocorrelation plots which correspond to three different functions in the parameter space: the number of clusters (left), the deviance (middle) and z_{k_3} (right). The non-conjugate Gaussian MDP model is fitted to the "bimod-100" (top) and "bimod-1000" datasets of Papaspiliopoulos & Roberts (2008), with $\alpha = 1$.

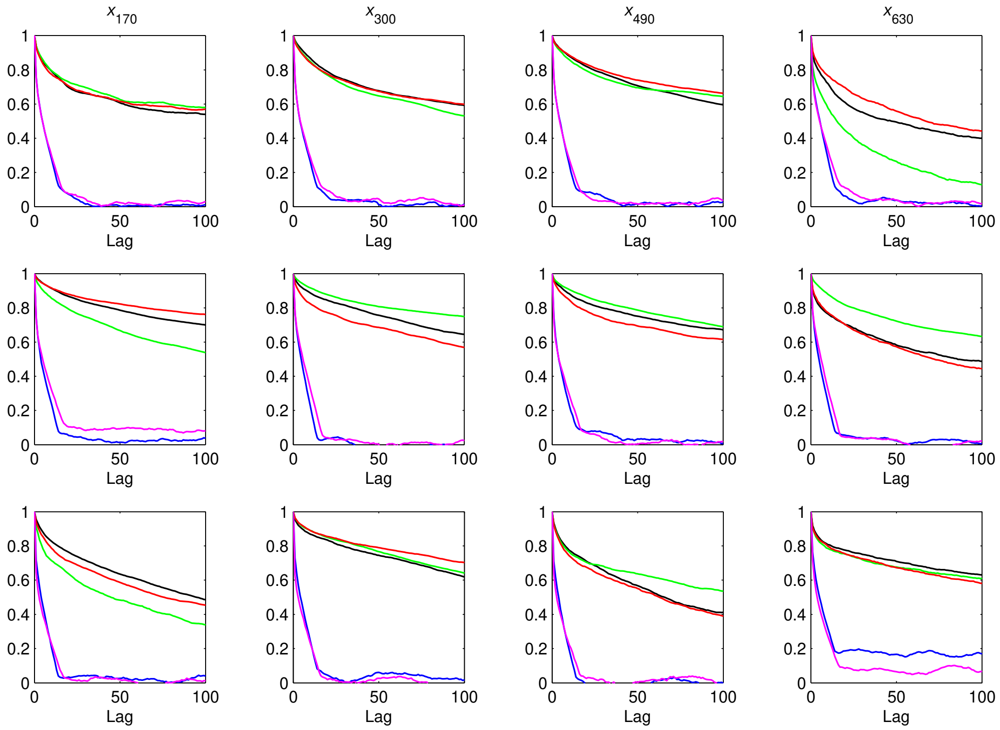

Figure 3: Autocorrelation of m_{s_t} at various time instances. (a) *lepto 1000*, (b) *bimod 1000* and (c) *trimod 1000*. The autocorrelation times are significantly larger when updating s_i one-at-a-time using local Gibbs updates compared to updating the entire sequence $\boldsymbol{s}$ using forward-backward sampling. (Black) Marginal Gibbs Sampler, (Green) Slice Sampler using local updates, (Red) Block Gibbs Sampler using local updates, (Blue) Slice Sampler using forward-backward updates and (Purple) Block Gibbs Sampler using forward-backward updates.

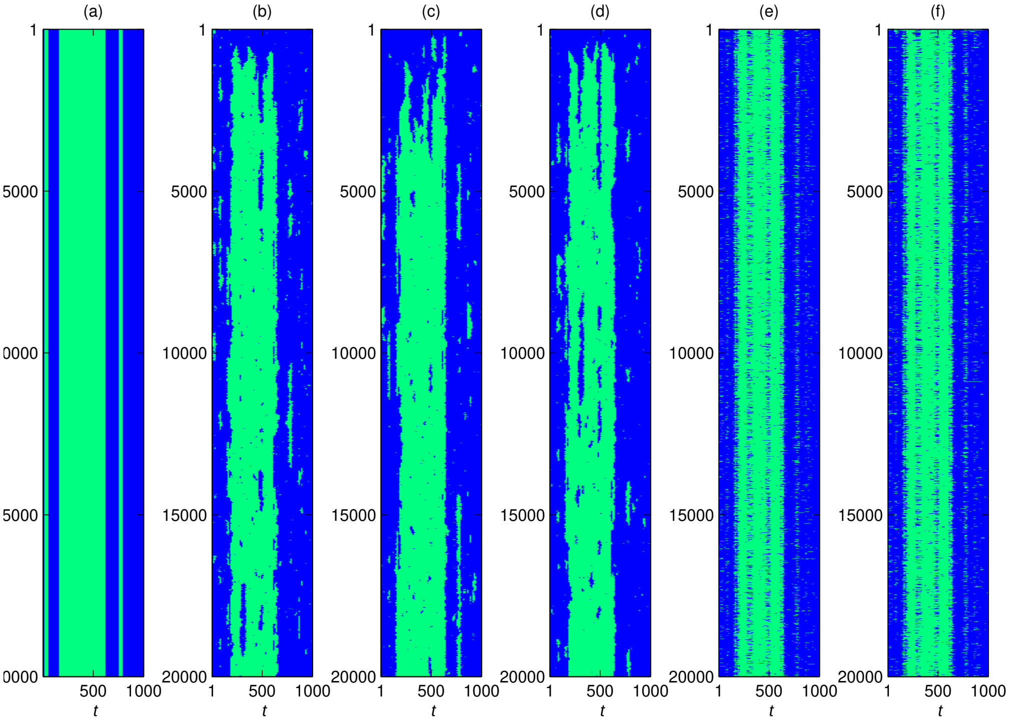

Figure 4: MCMC Samples of $\boldsymbol{s}$. (a) Ground Truth, (b) Marginal Gibbs Sampler, (c) Slice Sampler with local updates, (d) Block Gibbs Sampler with local updates, (e) Slice Sampler with forward-backward updates and (f) Block Gibbs Sampler with forward-backward updates. There is a significant amount of correlation in the samples of $\boldsymbol{s}$ from the samplers employing local Gibbs updates compared to the samplers using forward-backward sampling.

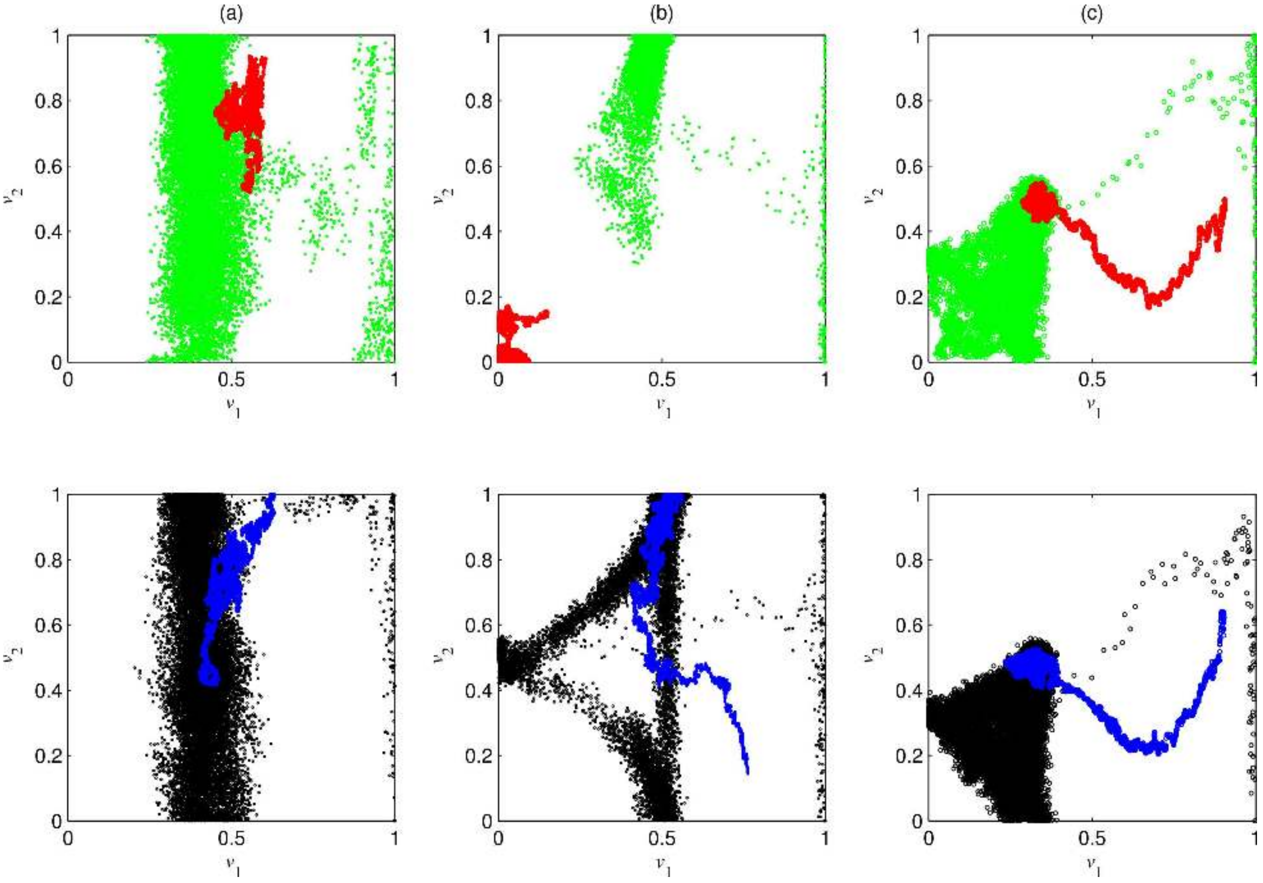

Figure 5: Gibbs Sampler output for (v_1, v_2). (a) *lepto 1000*, (b) *bimod 1000* and (c) *trimod 1000*. The combination of the Block Gibbs Sampler with forward-backward updating of the hidden states is able to explore the posterior distribution of v most efficiently. (Red) Slice sampler with local updates, (Green) Block Gibbs Sampler with local updates, (Blue) Slice sampler with forward-backward updates and (Black) Block Gibbs Sampler with forward-backward updates.

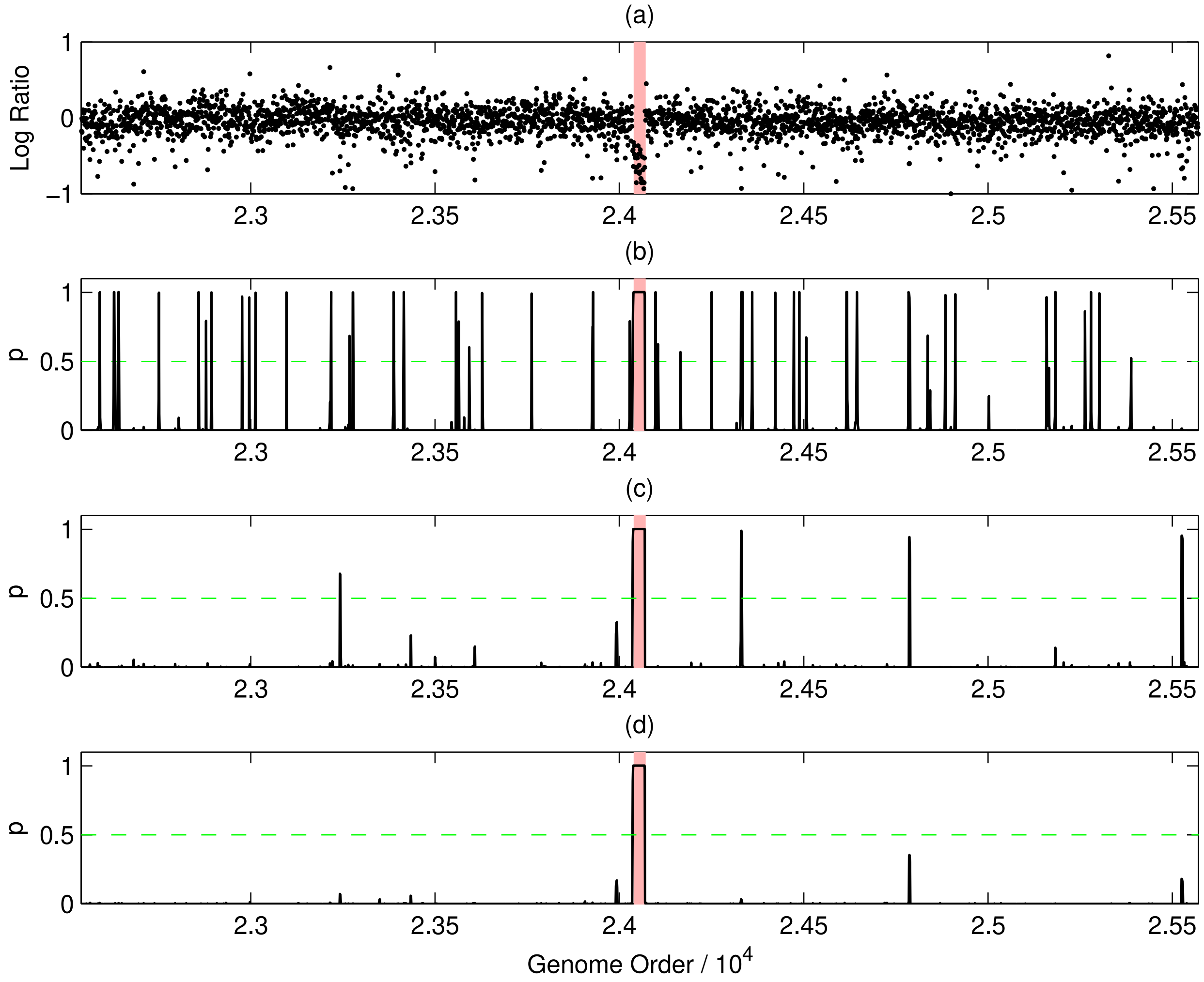

Figure 6: Mouse ROMA analysis. Chromosome 5. (a) The region indicated (red) contains a confirmed deletion (Lakshmi et al., 2006). (b) Using the G-HMM is able to identify this known copy number variant, however, it also detects many additional copy number variants on this chromosome most of which must be false positives. (c) The R-HMM reduces the number of false positives but (d) the MDP-HMM identifies only the known copy number variant and no other copy number alterations on this chromosome.

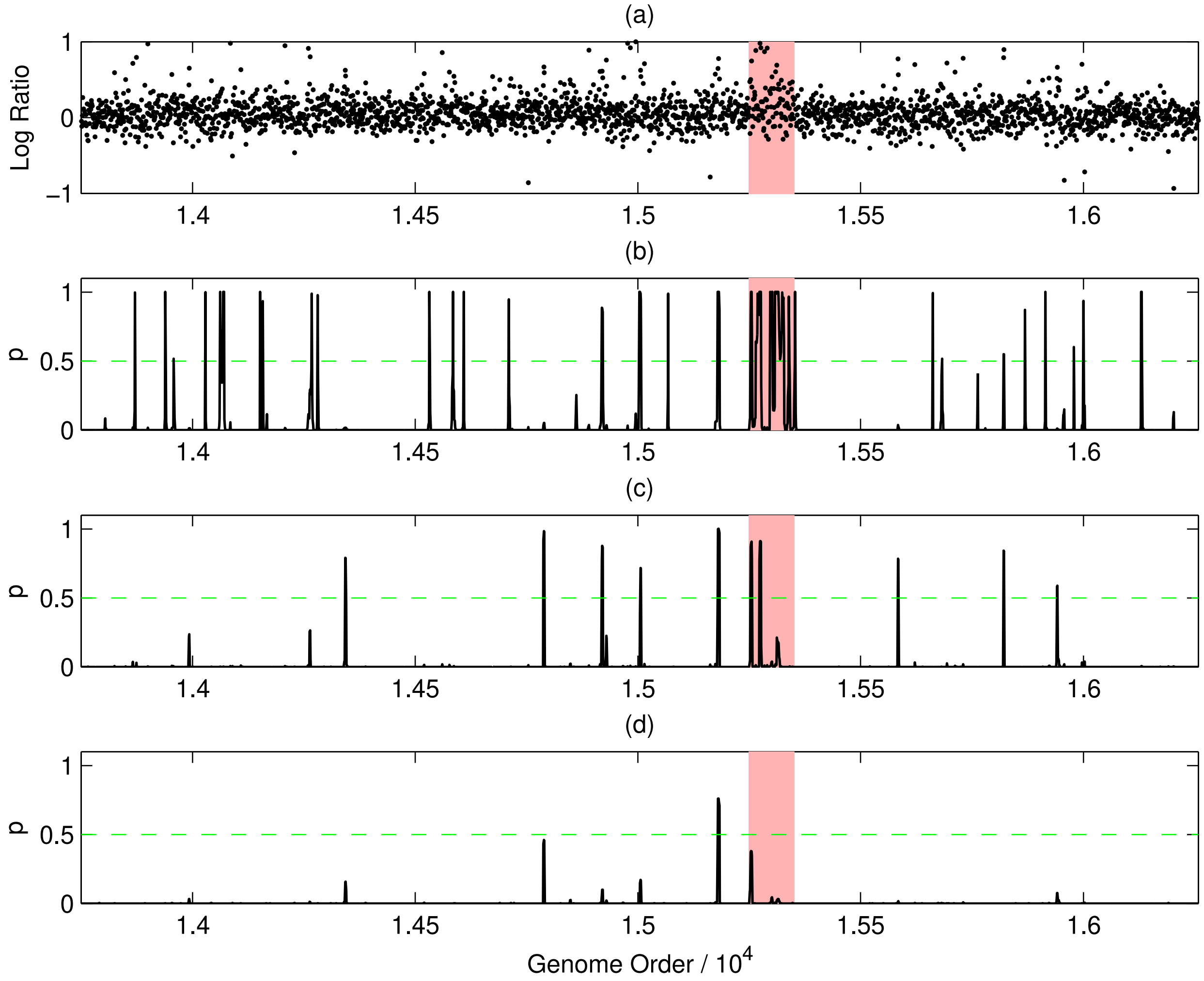

Figure 7: Mouse ROMA analysis. Chromosome 3. (a) The region indicated (red) contains no copy number alterations but contains SNPs that can disrupt the binding of probes on the microarray (Lakshmi et al., 2006). The (b) G-HMM and (c) R-HMM produce a number of false positive copy number alteration calls in this region but (d) the MDP-HMM identifies no copy number alterations with posterior probability greater than the threshold of 0.5 in the region.

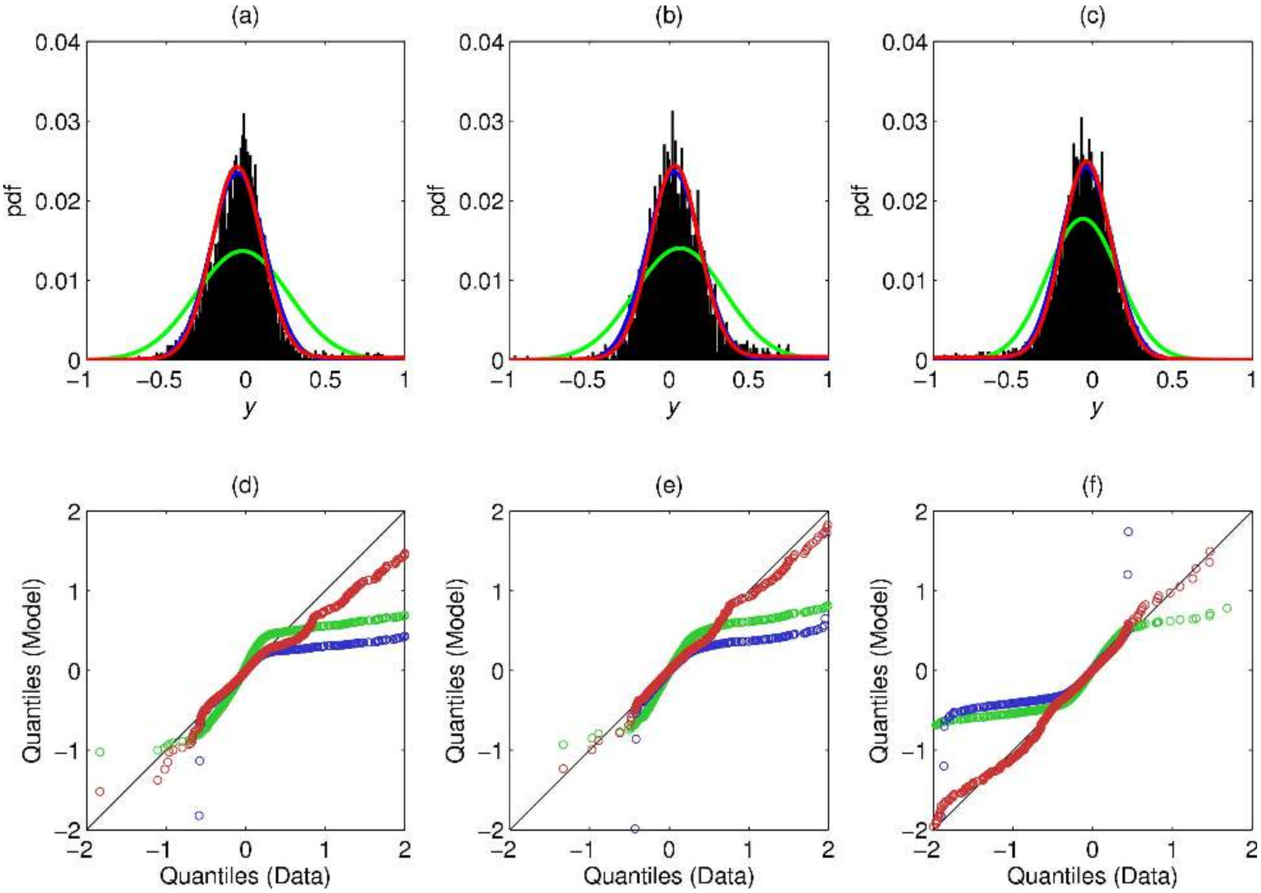

Figure 8: QQ-plots of predictive distributions versus ROMA data. (a, d) Chromosome 3, (b, e) Chromosome 5, (c, f) Chromosome 9. The empirical distribution of the ROMA data appears to be heavy-tailed and asymmetric. This asymmetry can lead to false detection of copy number variants by the G-HMM and R-HMM. The increased flexibility of the MDP-HMM allows this asymmetry to be capture and explains why the MDP-HMM is able to give far more accurate predictions for copy number alteration. (Red) MDP-HMM, (Green) R-HMM and (Blue) G-HMM.