A Generalised Markov Sampler

JONATHAN M. KEITH, DIRK P. KROESE, DARRYN BRYANT

Department of Mathematics,
University of Queensland, Brisbane 4072, Australia
j.keith1@mailbox.uq.edu.au, kroese@maths.uq.edu.au, db@maths.uq.edu.au

Abstract

A recent development of the Markov chain Monte Carlo (MCMC) technique is the emergence of MCMC samplers that allow transitions between different models. Such samplers make possible a range of computational tasks involving models, including model selection, model evaluation, model averaging and hypothesis testing. An example of this type of sampler is the reversible jump MCMC sampler, which is a generalisation of the Metropolis-Hastings algorithm. Here we present a new MCMC sampler of this type. The new sampler is a generalisation of the Gibbs sampler, but somewhat surprisingly, it also turns out to encompass as particular cases all of the well-known MCMC samplers, including those of Metropolis, Barker, and Hastings. Moreover, the new sampler generalises the reversible jump MCMC. It therefore appears to be a very general framework for Markov chain Monte Carlo sampling. This paper describes the new sampler and illustrates its use in three applications in Computational Biology, specifically determination of consensus sequences, phylogenetic inference and delineation of isochores via multiple change-point analysis.

Keywords: model determination, Markov chain Monte Carlo, Gibbs sampler, simulated annealing, string sampler, consensus sequence, phylogenetic inference, isochores, multiple change-point analysis.

1 Introduction

Markov chain Monte Carlo (MCMC) sampling (also called Markov chain Monte Carlo simulation) is a computational technique for simulating the drawing of a sample from a given probability distribution. MCMC sampling is frequently used in Bayesian inference to simulate sampling of a posterior distribution. However, MCMC sampling is not limited to Bayesian applications. In particular, it is the basis of the versatile optimisation technique known as *simulated annealing* (discussed below).

The idea of MCMC is to generate a (time-homogeneous) Markov chain $\{X_0, X_1, \ldots\}$ in a *target space* $\mathscr{X}$ in such a way that the limiting distribution of the chain has the required distribution. The random behaviour of the chain is governed by a *transition kernel* K, defined by

$$K(x, B) = \mathbb{P}(X_{n+1} \in B \,|\, X_n = x),$$

for any $x \in \mathscr{X}$ and (measurable) subset B of $\mathscr{X}$. For *countable* $\mathscr{X}$ it is easier to work with the (one-step) transition matrix P instead, which is defined by

$$P(x, y) = \mathbb{P}(X_{n+1} = y \,|\, X_n = x), \quad x, y \in \mathscr{X}.$$

For *non-denumerable* $\mathcal{X}$, the Markov chain is usually defined such that the kernel K has a *transition density*, also denoted by P, that is,

$$K(x, B) = \int_B P(x, y)\, dy,$$

so that in this case we may interpret $P(x, y)\, dy$ as the infinitesimal probability of jumping from x to $(y, y + dy)$. In our review of the major MCMC samplers below, we formulate the samplers for the *countable* case. In that case we assume that the required distribution, π say, has a probability mass function (pmf) f. In other words $\pi(\{x\}) = f(x)$, for all x. The samplers are easily generalised for the non-denumerable case by substituting pmf's with probability density functions (pdf's) and sums with integrals. We will view both pmf and pdf as a *density* with respect to some underlying measure – usually based on the counting or Lebesgue measure.

Metropolis-Hastings Sampler

The original MCMC sampler was developed by Metropolis *et al.* [15]. This sampler relies on a symmetric but otherwise arbitrary transition matrix A. The sampler consists of the following steps performed iteratively.

1. Given $X_n = x$ draw $Y \in \mathcal{X}$ in accordance with the density $A(x, \cdot)$.
2. Draw a Uniform(0,1) random variable U.
3. If $U \leq f(Y)/f(X_n)$ then set $X_{n+1} = Y$, otherwise set $X_{n+1} = X_n = x$.

The ratio $f(y)/f(x)$ is called the *acceptance ratio* and the matrix A is called the *proposal* transition matrix. The one-step transition matrix P of the Markov chain generated in this way is given by

$$P(x, y) = A(x, y)\ \min\left\{\frac{f(y)}{f(x)}, 1\right\}, \quad x \neq y\ .$$

Consequently, using also the symmetry of A, the density f satisfies the *detailed balance equation*:

$$f(x)\, P(x, y) = f(y)\, P(y, x)\ .$$

In particular, $\{f(x)\}$ gives the *stationary* distribution of the Markov process, that is, $\sum_x f(x) P(x, y) = f(y)$ for all y. This is the *limiting* distribution if the process is irreducible and aperiodic. (Metropolis' sampler was actually somewhat more complicated than this: it involved performing the above steps for each coordinate separately, and cycling through the coordinates. However, the sampler described above is the one generally meant when one refers to 'the Metropolis sampler'.)

Another MCMC sampler, rarely used but relevant to this paper, was developed by Barker [1]. It differs from the Metropolis sampler only in that it defines the acceptance ratio to be $f(y)/(f(x) + f(y))$, rather than $f(y)/f(x)$.

Hastings [7] generalised both of these samplers by defining the acceptance ratio to be:

$$\alpha(x,y) = \frac{s(x,y)}{1 + \dfrac{f(x)A(x,y)}{f(y)A(y,x)}} \tag{1}$$

where s is a symmetric, non-negative function such that $0 \leq \alpha(x,y) \leq 1$ for all $x, y \in \mathcal{X}(x \neq y)$. The transition matrix A no longer needs to be symmetric. With $s(x,y) = 1$ and symmetric A, Barker's sampler is seen to be an instance of Hastings'. With

$$s(x,y) = \min\left\{1 + \frac{f(x)A(x,y)}{f(y)A(y,x)},\ 1 + \frac{f(y)A(y,x)}{f(x)A(x,y)}\right\}$$

and symmetric A, Metropolis' sampler is seen to be an instance of Hastings'. When this function s is used with general A, the resulting sampler is known as the *Metropolis-Hastings sampler.*

Gibbs Sampler

The *Gibbs sampler* [Geman and Geman [5], Gelfand and Smith [4]] uses a somewhat different approach. It was originally developed to sample from Gibbs distributions, but is applicable whenever the state variable is a random vector. Let the dimension of the state space $\mathcal{X}$ be d. Suppose X is a random vector taking values in $\mathcal{X}$, with density f. Let $f_i(\cdot|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$ represent the conditional density of the ith coordinate of X given that the other components are $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d$. Then the sampler consists of the following steps performed iteratively:

1. Given $X_n = (x_{n.1}, \ldots, x_{n.d})$, generate $Y = (Y_1, \ldots, Y_d)$ consecutively as follows:

   given the values $Y_1 = y_1, \ldots, Y_{i-1} = y_{i-1}$ draw Y_i in accordance with the conditional density $f_i(\cdot|y_1, \ldots, y_{i-1}, x_{n.i+1}, \ldots, x_{n.d})$.

2. Let $X_{n+1} = Y$.

Gibbs sampling is advantageous if it is easier to sample from the conditional distributions than from the full distribution.

Recently, a number of MCMC samplers have been developed for the purposes of Bayesian model determination and comparison [2, 19, 6]. These samplers allow transitions between models that are parameterised differently, or that are not amenable to parameterisation, and are here referred to as *model-switching samplers.* One reason why these samplers are important is that they enable Bayesian inference in situations where there is uncertainty about not only the parameters of a model, but also about the model itself. Such samplers can be used for Bayesian model selection, model evaluation, model averaging and hypothesis testing. They can also be used in non-Bayesian applications. For example, they can be used in the context of simulated annealing to search for optima in spaces where the dimension of points is not fixed.

The *reversible-jump MCMC sampler* [6] is a model-switching generalisation of the Metropolis-Hastings sampler. The generalisation involves two elements. The first is to allow countably many move types, each with its own proposal transition matrix. Let $\sigma_x(m)$ be the probability of selecting move type m when the current element is x and let A_m be the proposal transition matrix for move type m. The second element is to define an individual acceptance ratio α_m for each move type m as

$$\alpha_m(x,y) = \min\left\{1, \frac{\sigma_y(m)\, f(y)\, A_m(y,x)}{\sigma_x(m)\, f(x)\, A_m(x,y)}\right\}. \tag{2}$$

In this paper, we develop a model-switching sampler that generalises the Gibbs sampler in a natural way. In so doing, we refute Green's claim that "the Gibbs sampler hardly even makes sense when x has a length that is not fixed, and elements which need not have a fixed interpretation across all models" [6]. Interestingly, it turns out that the new sampler encompasses the samplers of Metropolis, Barker and Hastings as particular cases. Moreover, it encompasses the reversible jump MCMC. It therefore appears to provide a very general framework for MCMC sampling.

Simulated Annealing

In our examples, we make much use of the simulated annealing technique. This technique uses MCMC sampling to find a *mode* of a density f – that is, a point where f is maximal. It involves defining a family of densities of the form $f_\gamma(x) \propto [f(x)]^{1/\gamma}$ where the parameter γ is called the *temperature* of the distribution. MCMC sampling is used to draw a single element x_k from f_{γ_k}, for successively lower temperatures $\gamma_1, \gamma_2, \ldots$. Each element x_k is used as the initial element of the next chain. As the temperature is reduced, the distributions become sharply peaked at the global maxima of f. Thus the x_k converge to a point. The x_k can converge to a local maximum, but this possibility is reduced by careful selection of successive temperatures. The sequence of temperatures, or *annealing schedule*, is therefore critical to the success of the method. In the examples described in this paper, the annealing schedule is a geometric progression starting with a specified initial temperature and multiplying by a *cooling factor* in the interval $(0,1)$ after each iteration. Simulated annealing can also be applied to non-probabilistic optimisation problems. Given an objective function $S(x)$, one defines a Boltzmann distribution via the density $f(x) \propto e^{-S(x)}$ or $f(x) \propto e^{S(x)}$, depending on whether the objective is to minimise or maximise S. Global optima of S are then obtained by searching for the mode of the Boltzmann distribution. Thus, model-switching samplers extend the scope of simulated annealing to optimisation problems in spaces composed of various 'models'.

The paper is structured as follows. In Section 2, the new sampler is described for countable spaces and the common MCMC samplers are shown to be instances of it. In Section 3, we illustrate the new sampler on some countable spaces. In Section 4, we extend the sampler for non-denumerable spaces, and show that it generalises the reversible jump MCMC. In Section 5, we present an example illustrating the use of the sampler in a non-denumerable space. Some concluding remarks are made in Section 6.

2 A Generalised MCMC Sampler

Suppose that one wishes to sample from a distribution with density f over a space $\mathcal{X}$. We refer to $\mathcal{X}$ as the *target set*. The Markov samplers mentioned above generate a Markov chain in $\mathcal{X}$. The sampler presented in this section is different in that it generates a chain in a space $\mathcal{I} \times \mathcal{X}$, where $\mathcal{I}$ is referred to as the *index set*. Its role is to provide an index for the types of transitions that can be made at each step of the chain. To motivate the introduction of this set, we show that it arises naturally within the context of the Gibbs sampler.

Let $G := \{X_1, X_2, \ldots\}$ be a Markov chain generated by a Gibbs sampler in a target set of dimension d. Each new element X_{n+1} of the chain is obtained by updating each coordinate of X_n in turn, thus generating a sequence of d elements $X_{n1}, X_{n2}, \ldots, X_{nd}$ in $\mathcal{X}$, where the last element is equal to X_{n+1}. (It is important to note that X_{ni} does *not* refer to the ith coordinate of X_n, but rather to the element of $\mathcal{X}$ obtained after updating the ith coordinate.) The stochastic process $G' = \{X_{11}, \ldots, X_{1d}, X_{21}, \ldots, X_{2d}, X_{31}, \ldots\}$ has the same limiting distribution as G, but is no longer a Markov chain. However, if we define the index set $\mathcal{I} = \{1, 2, \ldots, d\}$ then the chain $G'' = \{(1, X_{11}), (2, X_{12}), \ldots, (d, X_{1d}), (1, X_{21}), \ldots\}$ *is* a Markov chain in $\mathcal{I} \times \mathcal{X}$, and its projection onto $\mathcal{X}$ is G'. Thus the Gibbs sampler may be regarded as generating a Markov chain in the space $\mathcal{I} \times \mathcal{X}$ whose projection onto $\mathcal{X}$ has the required limiting distribution.

This perspective on the Gibbs sampler can be readily generalised in the following manner. Given a target set $\mathcal{X}$, and an index set $\mathcal{I}$, let $\mathcal{U} \subseteq \mathcal{I} \times \mathcal{X}$ be such that the projections of $\mathcal{U}$ onto $\mathcal{X}$ and $\mathcal{I}$ are surjective. For each $x \in \mathcal{X}$, let $\mathcal{Q}(x)$ be the set $\{(k, z) \in \mathcal{U} : z = x\}$. The set $\mathcal{Q}(x)$ functions as a catalogue of the *types* of transitions that one may make from the element x. For example, in the Gibbs sampler $\mathcal{Q}(x) = \mathcal{I} \times \{x\} = \{(1, x), (2, x), \ldots, (d, x)\}$ and may be interpreted as a list of the coordinates of x that may be updated. See Figure 1 for an illustration.

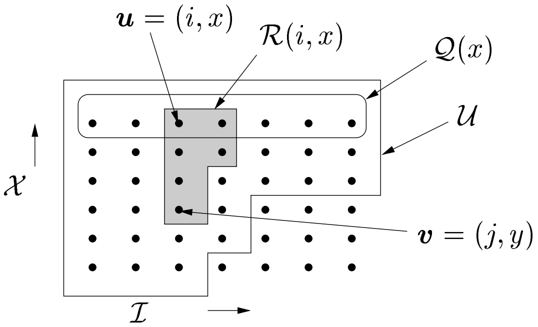

Figure 1: An illustration of the definitions

To select the type of transition to be made at a given step of the chain, we define for every $x \in \mathcal{X}$ a transition matrix Q_x on $\mathcal{Q}(x)$. Let q_x be the density of a distribution that is stationary with respect to Q_x. It will be convenient to also define a "global" transition matrix Q on $\mathcal{U}$ by

$$Q((i,x),(j,y)) = \begin{cases} Q_x((i,x),(j,y)), & \text{for } (j,y) \in \mathcal{Q}(x), \\ 0 & \text{otherwise.} \end{cases}$$

For the Gibbs sampler we have

$$Q((i,x),(j,x)) = \begin{cases} 1 & \text{if } j = i+1 \text{ or } j = 1, i = d \\ 0 & \text{otherwise.} \end{cases}$$

This transition matrix cycles through the d types of update that may be performed on x.

Having selected the type of transition to be made, a transition of that type is then selected in the following manner. For each $\boldsymbol{u} = (i,x) \in \mathcal{U}$, let $\mathcal{R}(\boldsymbol{u}) = \mathcal{R}((i,x)) \equiv \mathcal{R}(i,x)$ be the set of possible transitions (see Figure 1). We require that these sets form a partition of $\mathcal{U}$, that is:

$$\boldsymbol{v} \in \mathcal{R}(\boldsymbol{u}) \quad \Longleftrightarrow \quad \boldsymbol{u} \in \mathcal{R}(\boldsymbol{v}) \tag{3}$$

and

$$\boldsymbol{v} \in \mathcal{R}(\boldsymbol{u}),\, \boldsymbol{w} \in \mathcal{R}(\boldsymbol{v}) \quad \Longrightarrow \quad \boldsymbol{w} \in \mathcal{R}(\boldsymbol{u})\,. \tag{4}$$

Note that $\boldsymbol{u} \in \mathcal{R}(\boldsymbol{u})$. On $\mathcal{R}(\boldsymbol{u})$ we define a transition matrix $R_{\boldsymbol{u}}$ as follows. Let $\boldsymbol{u} = (i,x)$, $\boldsymbol{v} = (j,y)$ and $\boldsymbol{w} = (k,z)$ then

$$R_{\boldsymbol{u}}(\boldsymbol{u},\boldsymbol{v}) = \frac{f(y)\, q_y(\boldsymbol{v})}{\displaystyle\sum_{\boldsymbol{w} \in \mathcal{R}(\boldsymbol{u})} f(z)\, q_z(\boldsymbol{w})}\,. \tag{5}$$

Again, it will be convenient to also define a "global" transition matrix R on $\mathcal{U}$ by

$$R((i,x),(j,y)) = \begin{cases} R_{(i,x)}((i,x),(j,y)) & \text{for } (j,y) \in \mathcal{R}(i,x), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

For the Gibbs sampler, $\mathcal{R}(i,x)$ is the set of vectors (i,y) such that all coordinates of y are the same as those of x except for possibly the i-th coordinate. Moreover, q_x is the discrete uniform distribution on $\mathcal{Q}(x)$ – in other words q_x is constant. Thus for the Gibbs sampler R takes the form:

$$R((i,x),(j,y)) = \begin{cases} \dfrac{f(y)}{\displaystyle\sum_{(k,z) \in \mathcal{R}(i,x)} f(z)} & \text{for } (j,y) \in \mathcal{R}(i,x), \\ 0 & \text{otherwise.} \end{cases}$$

Returning to the general case, we now consider a Markov chain $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots\}$ on $\mathcal{U}$ with transition matrix P defined by

$$P = Q\,R\,.$$

Note that for $\boldsymbol{u} = (i,x)$ and $\boldsymbol{v} = (j,y)$,

$$P(\boldsymbol{u},\boldsymbol{v}) = \sum_{\boldsymbol{w}\in\mathcal{U}} Q(\boldsymbol{u},\boldsymbol{w})\, R(\boldsymbol{w},\boldsymbol{v}) = \sum_{\boldsymbol{w}\in\mathcal{Q}(x)\cap\mathcal{R}(\boldsymbol{v})} Q(\boldsymbol{u},\boldsymbol{w})\, R(\boldsymbol{w},\boldsymbol{v}) \ .$$

In many instances, the intersection of $\mathcal{Q}(x)$ and $\mathcal{R}(\boldsymbol{v})$ in the formula above must be either empty or contain only a single element (k,x), in which case $P((i,x),(j,y)) = Q((i,x),(k,x))\, R((k,x),(j,y))$. This is the case in the Gibbs sampler, where

$$P((i,x),(j,y)) = \begin{cases} \dfrac{f(y)}{\displaystyle\sum_{(k,z)\in\mathcal{R}(j,x)} f(z)} & \text{if } (j = i+1 \text{ or } i = d, j = 1) \\ & \text{and } (j,y) \in \mathcal{R}(j,x), \\ 0 & \text{otherwise.} \end{cases}$$

Let μ be the distribution on $\mathcal{U}$ defined by

$$\mu(\{(i,x)\}) = f(x)\, q_x(i) \ ,$$

where $q_x(i)$ is an abbreviation for $q_x((i,x))$. Let us write $\mu(i,x)$ for $\mu(\{(i,x)\})$. It is easy to check that μ is indeed a distribution on $\mathcal{U}$. Moreover, μ is stationary with respect to R. This follows from the fact that the local balance equations

$$(7) \qquad \mu(i,x)\, R((i,x),(j,y)) = \mu(j,y)\, R((j,y),(i,x))$$

hold, and that consequently

$$\sum_{(i,x)\in\mathcal{U}} \mu(i,x)R((i,x),(j,y)) = \mu(j,y) \ .$$

The distribution μ is also stationary with respect to Q, since

$$\begin{aligned} \sum_{(i,x)\in\mathcal{U}} \mu(i,x)\, Q((i,x),(j,y)) &= \sum_{(i,y)\in\mathcal{Q}(y)} f(y)\, q_y(i)\, Q((i,y),(j,y)) \\ &= f(y)\, q_y(j) = \mu(j,y) \ . \end{aligned}$$

Theorem 2.1 *The distribution μ is stationary with respect to P.*

PROOF. This follows directly from the fact that $P = QR$ and that μ is stationary for both Q and R. Specifically, in matrix notation

$$\mu P = \mu\, QR = \mu R = \mu \ .$$

□

Note that if P is irreducible and aperiodic, then μ is also the limiting distribution of the process P.

Based on the discussion above we propose the following algorithm.

Algorithm 2.1 [Generalised MCMC Sampler] Starting with an arbitrary $\boldsymbol{U}_0$, perform the following steps iteratively:

1. **[Q-step]** Given $\boldsymbol{U}_n = (i, x)$, generate $\boldsymbol{V} \in \mathcal{Q}(x)$ by drawing from the distribution with density $Q((i, x), \cdot)$.

2. **[R-step]** Given $\boldsymbol{V} = (j, y)$, generate $\boldsymbol{W} \in \mathcal{R}(j, y)$ by drawing from the distribution with density $R((j, y), \cdot)$.

3. Let $\boldsymbol{U}_{n+1} = \boldsymbol{W}$.

This algorithm generates a Markov chain $\{\boldsymbol{U}_0, \boldsymbol{U}_1, \ldots\} = \{(I_0, X_0), (I_1, X_1), \ldots\}$ such that the limiting distribution of X_n as $n \to \infty$ is f, provided that P is irreducible and aperiodic.

The above discussion makes clear that the Gibbs sampler is an instance of our generalised Markov sampler. With a slight modification, the sampler can be further generalised to include as instances Metropolis' sampler, Hastings' generalisations, and the reversible jump sampler. The modification is to redefine R as:

$$R(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} \dfrac{s(\boldsymbol{u}, \boldsymbol{v})\, f(y)\, q_y(j)}{\displaystyle\sum_{\boldsymbol{w} \in \mathcal{R}(\boldsymbol{u})} f(z)\, q_z(k)} & \text{if } \boldsymbol{v} \in \mathcal{R}(\boldsymbol{u}) \setminus \{\boldsymbol{u}\} \\ 1 - \displaystyle\sum_{\boldsymbol{w} \in \mathcal{R}(\boldsymbol{u}) \setminus \{\boldsymbol{u}\}} R(\boldsymbol{u}, \boldsymbol{w}) & \text{if } \boldsymbol{v} = \boldsymbol{u} \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $\boldsymbol{u} = (i, x)$, $\boldsymbol{v} = (j, y)$, $\boldsymbol{w} = (k, z)$ and s is a non-negative, symmetric function such that

$$\sum_{\boldsymbol{w} \in \mathcal{R}(\boldsymbol{u}) \setminus \{\boldsymbol{u}\}} R(\boldsymbol{u}, \boldsymbol{w}) \le 1.$$

Note that the local balance equations (7) still hold and hence μ is still stationary with respect to R and P.

We now show that Hastings' sampler (and by implication, the samplers of Metropolis and Barker) is an instance of our generalisation. We will see that the transition matrix Q functions as the proposal transition function, whereas R functions as the acceptance ratio. Let the index set $\mathcal{I}$ be a copy of the target set $\mathcal{X}$ and let $\mathcal{U} = \mathcal{I} \times \mathcal{X}$. Then for each $x \in \mathcal{X}$, the set $\mathcal{Q}(x) = \mathcal{I} \times \{x\}$. Now, given an arbitrary transition matrix A on $\mathcal{X}$, or more specifically the proposal transition matrix used by Hastings' sampler, define $Q_x((i, x), (j, x)) = A(x, j)$, for all $j \in \mathcal{I}$, so that

$$Q((i, x), (j, y)) = \begin{cases} A(x, j) & \text{if } y = x, \\ 0 & \text{otherwise.} \end{cases}$$

The transition $(i, x) \to (j, x)$ may be interpreted as the proposal of a new element j. Note that since $Q_x((i, x), (j, x))$ does not depend on i, $q_x := Q_x((i, x), \cdot) = A(x, \cdot)$ is the stationary density of Q_x on $\mathcal{Q}(x)$.

Now define $\mathcal{R}(i,x) = \{(i,x),(x,i)\}$ for each $(i,x) \in \mathcal{U}$. Then (8) reduces to:

$$R((i,x),(j,y)) = \begin{cases} \dfrac{s((i,x),(x,i))}{1+\dfrac{f(x)A(x,i)}{f(i)A(i,x)}} & \text{if } (j,y) = (x,i) \\ 1 - R((i,x),(x,i)) & \text{if } (j,y) = (i,x) \\ 0 & \text{otherwise} \end{cases}$$

or alternatively

$$R((i,x),(j,y)) = \begin{cases} \alpha(x,i) & \text{if } (j,y) = (x,i) \\ 1 - \alpha(x,i) & \text{if } (j,y) = (i,x) \\ 0 & \text{otherwise,} \end{cases}$$

where α is the acceptance ratio given in (1). The transition $(i,x) \to (x,i)$ may be interpreted as acceptance of the element i. Thus, in effect, Q is used to propose a new element in accordance with the transition matrix A, and R is used to accept or reject it in accordance with the acceptance ratio $\alpha(x,j)$. This is exactly the procedure used by Hasting's sampler.

3 Applications in countable spaces

In this section, the sampler is applied to two problems involving countable spaces. The first example illustrates that the new sampler extends the scope of the Gibbs sampler to spaces that do not have a fixed coordinate system. The second example illustrates that the new sampler extends the scope of the Gibbs sampler to spaces where the dimension of elements is not fixed.

3.1 Coordinate-free Gibbs sampling with an application to phylogenetic inference

The Gibbs sampler is applicable to spaces that can be parameterised in terms of some fixed system of coordinates. It involves systematically updating these coordinates one at a time (or sometimes in blocks). However, the essential idea of the Gibbs sampler, that of systematically updating parts of the previous element while holding the other parts constant, is potentially useful in many instances where the space cannot be parameterised using a fixed coordinate system, or where the coordinates cannot be varied independently. In this section, we demonstrate that our generalisation of the Gibbs sampler enables Gibbs-like sampling of such spaces. We refer to the approach as *coordinate-free Gibbs sampling.* One application of coordinate free Gibbs sampling is in solving combinatorial optimisation problems via simulated annealing.

Coordinate-free Gibbs sampling is applicable whenever a fixed, finite number d of move types can be defined for each element x in the target space $\mathcal{X}$. Let $\mathcal{M}(x)$ be the set of move types available at x. In the conventional Gibbs sampler, the d move types involve updating each of the d coordinates. Here, however, the move types need not be defined in terms of coordinates at all. Moreover, a different set of move types may be defined for each $x \in \mathcal{X}$.

In order to implement systematic updating, one must be able to determine what move type one is currently up to. The move types available before a transition must therefore be placed in correspondence with those available after. This is achieved by defining a bijection $h_{x,y}$ between $\mathcal{M}(x)$ and $\mathcal{M}(y)$ for all adjacent elements $x, y \in \mathcal{X}$. (An element y is *adjacent* to x if it can be reached in a single transition from x.) Now, given an ordering of $\mathcal{M}(x)$, one may use $h_{x,y}$ to induce an ordering of $\mathcal{M}(y)$ at an adjacent element y. Continuing this process for a sequence of transitions, one may induce an ordering of $\mathcal{M}(z)$ at an element $z \in \mathcal{X}$ *not* adjacent to x. However, in general, this ordering is not unique. An arbitrary element $z \in \mathcal{X}$ may be reached by more than one sequence of transitions from x, and the induced order of $\mathcal{M}(z)$ may depend on the sequence of transitions used. We therefore allow more than one ordering of the move types at each element. Let $\Phi(x)$ be the set of allowed orderings (that is, permutations) of $\mathcal{M}(x)$, for each $x \in \mathcal{X}$. If a canonical ordering of move types *can* be defined for each $x \in \mathcal{X}$, as in the conventional Gibbs sampler, then $\Phi(x)$ need only contain a single permutation.

The coordinate-free Gibbs sampler can now be described in the notation of Section 2. Define an index space $\mathcal{I} = \cup_{x\in\mathcal{X}} (\Phi(x) \times \{1, \ldots, d\})$ and let $\mathcal{Q}(x) = (\Phi(x) \times \{1, \ldots, d\}) \times \{x\}$ for each $x \in \mathcal{X}$. Thus an element $(\phi, m, x) \equiv ((\phi, m), x)$ of $\mathcal{Q}(x)$ contains a permutation ϕ of the move types at x and a position m in that permutation (so $\phi(m)$ is the current move type). Define $\mathcal{U} = \cup_{x\in\mathcal{X}} \mathcal{Q}(x)$. For the Q-step, define

$$Q_x((i,x),(j,x)) = \begin{cases} 1 & \text{if } i = (\phi, m), j = (\phi, m+1),\ \phi \in \Phi(x), \\ & \quad m = 1, \ldots, d-1 \\ 1/M & \text{if } i = (\phi, d), j = (\phi', 1), \phi, \phi' \in \Phi(x) \\ 0 & \text{otherwise.} \end{cases}$$

where M is the cardinality of $\Phi(x)$. Thus the Q-step cycles through $\mathcal{M}(x)$ in the order induced by ϕ until it reaches the last move type, at which point it selects a new permutation ϕ' uniformly and randomly in $\Phi(x)$ and begins again. (Selecting a new bijection at the end of each cycle ensures that Q_x is irreducible and aperiodic. This may not always be necessary to ensure that the overall process is irreducible and aperiodic.) Note that the limiting distribution of Q_x is the uniform distribution $q_x(i, x) = 1/(Md)$.

Next, we define a partition of $\mathcal{U}$ for the R-step. Let $\mathcal{R}(\phi, m, x) = \{(h_{x,y} \circ \phi, m, y) \in \mathcal{U} : x \to y \text{ is a move of type } \phi(m)\}$. Note that $h_{x,y} \circ \phi \in \Phi(y)$ is the ordering of move types at y induced by the transition $x \to y$. The R-step of the algorithm may now be performed by selecting an element of $\mathcal{R}(\phi, m, x)$, with the probability of selecting each element given by Equation (9).

$$R((i,x),(j,y)) = \begin{cases} \dfrac{f(y)}{\displaystyle\sum_{(k,z)\in\mathcal{R}(i,x)} f(z)} & \text{for } (j,y) \in \mathcal{R}(i,x), \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Here f is the density of the target distribution. Note that the term $q_x(i, x)$ has cancelled out, thus giving the conventional Gibbs formula. However, the absence of a fixed co-ordinate system differentiates this sampler from the conventional Gibbs sampler.

The coordinate-free Gibbs sampler is summarised in the following algorithm.

Algorithm 3.1 [Coordinate-free Gibbs Sampler] Starting with an arbitrary $x \in \mathcal{X}$ and $\phi \in \Phi(x)$, set $\boldsymbol{U}_0 = (\phi, 1, x)$, and perform the following steps iteratively:

1. **[Q-step]** Given $\boldsymbol{U}_n = (\phi, m, x)$, set $\boldsymbol{V} = (\phi, m+1, x)$ if $m < d$. If $m = d$, set $\boldsymbol{V} = (\phi', 1, x)$ for ϕ' selected uniformly and randomly from $\Phi(x)$.

2. **[R-step]** Given $\boldsymbol{V} = (\phi', m', x)$, generate $\boldsymbol{W} = (h_{x,y} \circ \phi', m', y)$ by drawing from the distribution with density $R((\phi', m', x), \cdot)$ defined in Equation (9).

3. Let $\boldsymbol{U}_{n+1} = \boldsymbol{W}$.

Application to phylogenetic trees

To illustrate coordinate-free Gibbs sampling, we describe a sampler for a space of phylogenetic trees. Such spaces arise in phylogenetic inference, that is, inference of evolutionary history. MCMC samplers for phylogenetic inference in a Bayesian context have been developed by Yang and Rannala [20], Mau and Newton [14], Larget and Simon [13], and Huelsenbeck [8]. The sampler presented here is applied in a non-Bayesian context. However, it can be adapted for Bayesian phylogenetic inference.

A phylogenetic tree for a set of taxa is a graph showing putative evolutionary relationships amongst those taxa. For example, the three possible phylogenetic trees for the taxa {mouse, human, pig, chicken} are shown in Figure 2(a). These trees are said to be *unrooted* because they show the divergences between taxa, but not the point of origin or *root* corresponding to the common ancestor. A rooted tree can be constructed by inserting a node into any edge of an unrooted tree, as shown in Figure 2(b). A sub-tree of an unrooted tree is obtained by deleting an edge of the tree and selecting one of the connected components, as illustrated in Figure 2(c).

Let the target space $\mathcal{X}$ be the set of all possible unrooted phylogenetic trees for n taxa. For each $x \in \mathcal{X}$, we consider move types of the form illustrated in Figure 3. These move types involve removing a sub-tree (Figure 3(b)) and re-inserting it at a new position in the tree (Figure 3(c), (d) and (e)). However, moves involving sub-trees containing $n-1$ taxa are excluded, since there are no alternative positions for such sub-trees. There are $3(n-2)$ remaining sub-trees, and thus the number of move types is $d = 3(n-2)$. The number of moves of each type is equal to the number of edges remaining after the sub-tree is removed. For each $x \in \mathcal{X}$, let $\Phi(x)$ be the set of all possible permutations of the $3(n-2)$ move types at x. The cardinality of $\Phi(x)$ is $[3(n-2)]!$.

For adjacent trees x and y, a bijection $h_{x,y}$ from the sub-trees of x to the sub-trees of y can be defined as follows. Suppose that x and y differ only by the

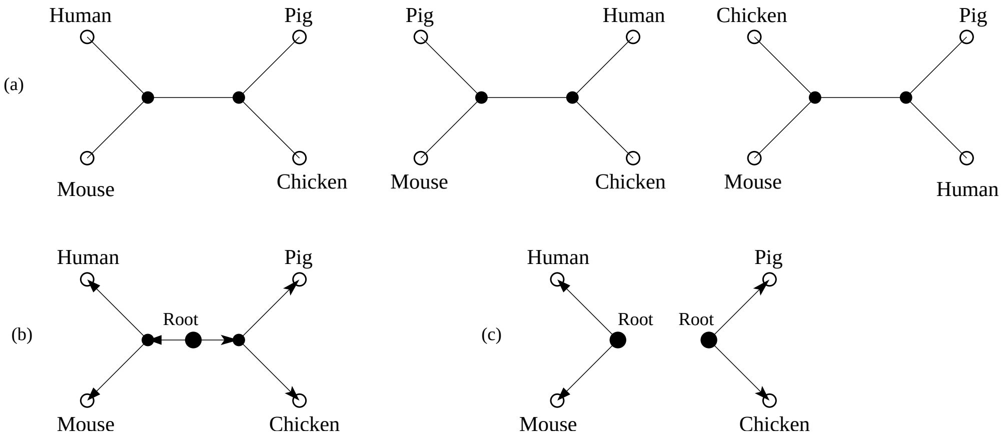

Figure 2: (a): Unrooted trees. (b): A rooted tree (the root shown is not the true root). (c): Two sub-trees obtained by deleting an edge.

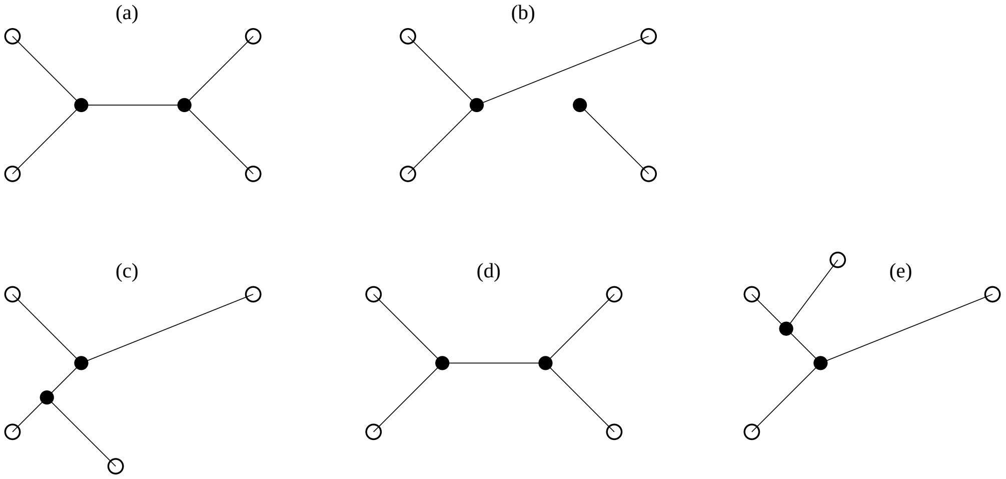

Figure 3: (a): Original Tree. (b): Tree with sub-tree removed. (c), (d) and (e): Trees obtained by inserting sub-tree into remaining edges.

placement of a sub-tree a (Figure 4, a and b). Removing a from x leaves two sub-trees b and c (Figure 4c). Similarly, removing a from y leaves two sub-trees d and e, where b is a sub-tree of d and e is a subtree of c (Figure 4d). Now, sub-tree a and all its sub-trees are unaltered by the transition $x \to y$, so they can be mapped to themselves. Moreover, the complements of these sub-trees in x can be mapped to their complements in y. (The *complement* of a sub-tree is the other sub-tree obtained by deleting the same edge as in Figure 2 (c).) Sub-tree b maps to d, and its complement maps to the complement of b in y. Proper sub-trees of b are unaltered by the transition, so they can be mapped to themselves and their complements can be mapped to themselves. Sub-tree c maps to e, and its complement maps to b. Proper sub-trees of c other than e are altered only by the possible insertion of a, so a natural mapping exists for them and for their complements. Finally, sub-tree e of x maps to the complement of

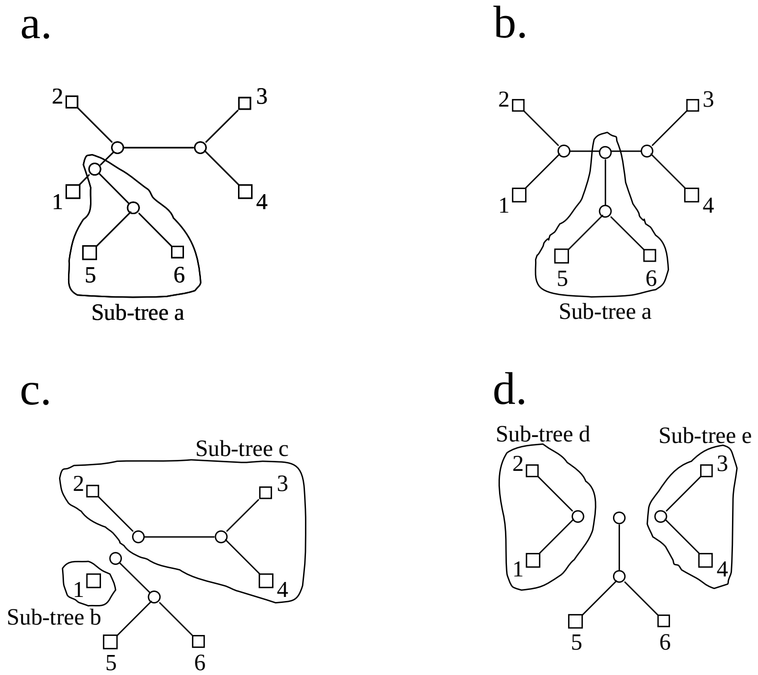

Figure 4: (a): Tree x and subtree a. (b): Tree y differs from tree x in the placement of subtree a. (c): Removing a from x leaves subtrees b and c. (d) Removing a from y leaves subtrees d and e.

d in y and the complement of e in x maps to the complement of e in y.

If we now substitute these terms into Equation 9 and Algorithm 3.1, the Q-step consists of determining the next sub-tree to be moved, whereas the R-step consists of selecting a new position for that sub-tree.

We are now almost ready to apply the sampler, but first we must define a scoring function S. To do this, we require some additional background on DNA and phylogenetic inference. DNA molecules consist of chemical components called *nucleotides*, arranged in a linear sequence. There are only four different nucleotides (labelled A, C, G and T), but they can be arranged in any order to generate a vast range of molecules. DNA is copied and passed from parents to offspring. Occasionally a mutation occurs during copying so that the DNA of the offspring differs from that of its parents. Mutations accumulate over time, and thus the DNA in different lineages diverges from that of the common ancestor.

In one approach to phylogenetic inference, the first step is to align the sequences. An example of a sequence alignment is shown in Figure 5. Each column of the alignment contains characters that are thought to be derived from a common ancestral character. The symbol '–' indicates that a character is absent at that position, either because it has been deleted, or because the other characters in that column have been inserted. The next step is to determine the tree that best explains the variations observed in the columns of the

```
GCAAGGTA---CCACAACTT
GTGAGGTA---CCACAAGTG
GTGAGGTA---CCACAAGTT
GTGAGGTA---CCACAGCTT
GCGACGTGGTACCAGAAGTG
GTGACGTG---CCACAAGTT
GTGACGTG---CCACAAGTC
```

Figure 5: Example of a sequence alignment

alignment. One way to score a tree is to sum the minimum number of *point mutations* (single-character insertions, deletions and substitutions) that must have occurred in each column if that tree is correct. Let the score for a tree x be $S(x)$. This score can be computed using an algorithm described by Fitch [3]. A tree that minimises S is said to have *maximum parsimony.*

We used Algorithm 3.1 in the context of simulated annealing to determine a maximum parsimony tree for a data set described by Murphy *et al.* [16]. The data set consisted of an alignment of 44 DNA sequences in 16,397 columns. The sequences were obtained by concatenating 19 gene sequences from the genomes of 42 placental mammals and 2 marsupials. Some sequences contained long deletions; indeed, whole genes have apparently been deleted from some of the genomes. This is inconvenient for parsimony analysis, since each occurrence of the symbol '–' is interpreted as a single-character insertion or deletion. We therefore replaced all instances of the gap character '–' with a wild-card character '?', indicating that the character at that position could be 'A', 'C', 'G', 'T' or '–'. (Wild-card characters are implemented as a set $\{A, C, G, T, -\}$, and are handled in the same way that sets at internal nodes are handled by Fitch's algorithm.) This adjustment discards some information, but it means that insertions and deletions make no contribution to the score.

The algorithm ran in less than a minute on a PC with a Pentium IV processor. In each iteration, $O(n)$ sub-trees are trialled in $O(n)$ positions, and the computation of each score takes $O(L)$ time (where n is the number of sequences, and L the alignment length). Each iteration therefore takes $O(n^2L)$ time. Independent runs of the algorithm invariably converged to one of the two trees shown in Figure 6. These two trees differ only slightly in their score. Trees were drawn using TreeView [18].

The variations between these trees and that of Murphy *et al.*, and the reasons for them, would make for an interesting discussion. However, these matters go beyond our current intentions.

3.2 The string sampler

In this section, we describe a Markov chain sampler for a density f on a space $\mathcal{X}$ consisting of *finite strings formed from characters of a finite alphabet* Σ. The authors have previously used this sampler to construct consensus sequences for families of DNA sequences [10] and to infer original DNA sequences from descendant sequences [11, 12]. We include this example here because it is a

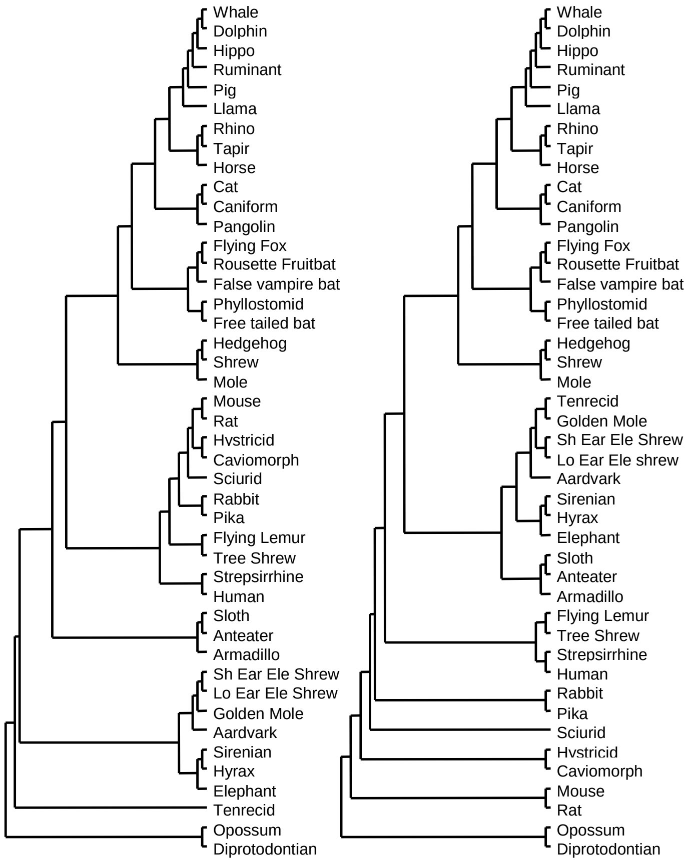

Figure 6: Maximum parsimony trees for the placental mammals.

simple illustration of the use of the generalised Gibbs sampler in a space where the dimension varies, and in order to elaborate on relevant technical details not covered in our earlier papers.

Note firstly that if the strings in $\mathcal{X}$ are all of a fixed length L then the conventional Gibbs sampler can be applied, since each string is a vector of L coordinates, with each coordinate taking a value in Σ. One may therefore cycle through the coordinates, updating each one in accordance with the corresponding conditional distribution. However, since $\mathcal{X}$ contains all strings of any finite length, the number of coordinates varies from point to point, and the conventional Gibbs sampler does not apply. The sampler must be able to step between strings of different lengths. We propose an algorithm that uses the new sampler to cycle through the coordinates just as in the Gibbs sampler, but allowing characters to be deleted or inserted between adjacent characters or at the ends of the string.

We first describe the algorithm using the notation of the previous section. The target set is $\mathcal{X}$. For ease of description, we append a terminating character to each string so that we may refer to "character $L+1$" of a string with length L.

For each string $x \in \mathcal{X}$, we define the following *index* states. In state (n, I), a character *insertion* may be considered immediately in front of character n, where $1 \le n \le |x| + 1$, $|x|$ denotes the length of x, and character $|x| + 1$ is the terminating character. In state (n, D), *deletion* or *substitution* of character n may be considered, where $1 \le n \le |x|$. Thus the index set is $\mathcal{I} = \{1, 2, \ldots\} \times \{I, D\}$. For each $x \in \mathcal{X}$ let

$$\mathcal{Q}(x) = \{(n, I, x) : n = 1, \ldots, |x| + 1\} \cup \{(n, D, x) : n = 1, \ldots, |x|\} .$$

Note here the abbreviations (n, I, x) and (n, D, x) for $((n, I), x)$ and $((n, D), x)$. Henceforth we will omit similar brackets where possible. The state space $\mathcal{U}$ is defined as the union of the sets $\mathcal{Q}(x)$. For each x we define the transition matrix Q_x on $\mathcal{Q}(x)$ by

$$Q_x((i, x), (j, x)) = \begin{cases} 1 & \text{for } i = (n, I),\ j = (n, D),\ n = 1, \ldots, |x|, \\ 1 & \text{for } i = (|x| + 1, I),\ j = (1, I), \\ 1 & \text{for } i = (n, D),\ j = (n + 1, I),\ n = 1, \ldots, |x|, \\ 0 & \text{otherwise.} \end{cases}$$

For fixed x, this transition matrix cycles through the $2|x| + 1$ states of $\mathcal{Q}(x)$ alternatively considering insertions and deletions (see Figure 7). The stationary distribution of Q_x is the discrete uniform distribution on $\mathcal{Q}(x)$, thus, $q_x(i, x) = 1/(2|x| + 1)$. The above specifies the Q-step of the sampler. Next, we specify the R-step. For each string x let $x_n^+(a)$ denote the string obtained from x by inserting character a immediately in front of the nth character of x. Similarly, let $x_n(a)$ denote the string obtained by replacing the nth character of x by a. Finally, let x_n^- be the string obtained by deleting the nth character of x. Now define

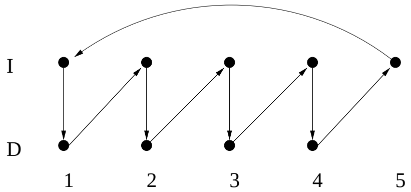

Figure 7: *The transition graph for Q_x*

$$\mathcal{R}(n, I, x) = \{(n, I, x)\} \cup \{(n, D, x_n^+(a)) : a \in \Sigma\},$$

and note that $\mathcal{R}(n, D, x) = \mathcal{R}(n, I, x_n^-)$. The transition function R is defined via (5) and (6) as

$$R((i, x), (j, y)) = \begin{cases} \dfrac{f(y)/(2|y| + 1)}{\displaystyle\sum_{(k,z) \in \mathcal{R}(i,x)} f(z)/(2|z| + 1)} & \text{for } (j, y) \in \mathcal{R}(i, x), \\ 0 & \text{otherwise.} \end{cases}$$

The discussion above leads to the following algorithm.

Algorithm 3.2 [String Sampler] Starting from an arbitrary $\boldsymbol{U}_0$, perform the following steps iteratively:

1. **[Q-step]** Given $\boldsymbol{U}_n = (i, x)$, let $(j, y) \in \mathcal{Q}(x)$ be the immediate successor of (i, x) in the transition graph of Figure 7.
2. **[R-step]** Generate $\boldsymbol{W} \in \mathcal{R}(j, y)$ by drawing from $R((j, y), \cdot)$. Specifically:
   (a) [**Insertion**] If $j = (n, I)$, randomly select a string Z from $\{y\} \cup \{y_n^+(a) : a \in \Sigma\}$ where y is weighted by $f(y)/(2|y| + 1)$ and $y_n^+(a)$ is weighted by $f(y_n^+(a))/(2|y| + 3)$ for each $a \in \Sigma$. If $Z = y$, then put $\boldsymbol{W} = (n, I, y)$; if $Z = y_n^+(a)$ put $\boldsymbol{W} = (n, D, y_n^+(a))$.
   (b) [**Deletion/Substitution**]. If $j = (n, D)$, randomly select a string Z from $\{y_n^-\} \cup \{y_n(a) : a \in \Sigma\}$ where y_n^- is weighted by $f(y^-)/(2|y| - 1)$ and $y_n(a)$ is weighted by $f(y_n(a))/(2|y| + 1)$ for each $a \in \Sigma$. If $Z = y_n^-$, then put $\boldsymbol{W} = (n, I, y_n^-)$; if $Z = y_n(a)$ put $\boldsymbol{W} = (n, D, y_n(a))$.
3. Let $\boldsymbol{U}_{n+1} = \boldsymbol{W}$.

Remark 3.1 Note that the characters in Σ can be any objects whatsoever, and hence the algorithm is not limited to biological sequences. It can be used in any space where the elements can be represented by text strings formed from a finite alphabet.

4 Generalising for non-denumerable spaces

In our third and final example (Section 5), the generalised Gibbs sampler is applied to a problem involving a non-denumerable target space. We therefore digress briefly to discuss the implementation of the sampler in non-denumerable spaces. We also show in this section that the reversible jump MCMC is a special case of the generalised Gibbs sampler.

The extension of the sampler to non-denumerable spaces is fairly straightforward. The sets $\mathcal{X}$, $\mathcal{I}$, $\mathcal{U}$, $\mathcal{Q}(x)$ and $\mathcal{R}(\boldsymbol{u})$ may all be defined as in Section 2, except that some or all of these sets may now be non-denumerable. Note also that f and q_x are densities with respect to reference measures that need not be counting measures. We find that an assumption has to be made to ensure that $R_{\boldsymbol{u}}(\boldsymbol{u}, \cdot)$ has a density with respect to a convenient reference measure.

Suppose that the target distribution has density f with respect to some reference measure φ on $\mathcal{X}$. Suppose further that for each x the stationary distribution of Q_x has density q_x with respect to some reference measure ψ_x on $\mathcal{Q}(x)$. We construct a reference measure ξ on $\mathcal{U}$ by putting

$$\xi(B) = \int_{\mathcal{X}} \psi_x(B \cap \mathcal{Q}(x))\, d\varphi(x)$$

for all measurable sets B. We assume that the following holds:

Assumption: There exists a measure ζ on the set $\mathcal{R} = \{\mathcal{R}(\boldsymbol{u}) : \boldsymbol{u} \in \mathcal{U}\}$ and measures η_r on r for each $r \in \mathcal{R}$ such that

$$\xi(A) = \int_{\mathcal{R}} \eta_r(A \cap r)\, d\zeta(r)$$

for all measurable sets A. In other words, the reference measure ξ can be decomposed into reference measures on the sets $\mathcal{R}(\boldsymbol{u})$.

Now we can define μ to be the measure on $\mathcal{U}$ with density $g(i,x) := f(x)\, q_x(i,x)$ with respect to ξ. Moreover, we set $R_{\boldsymbol{u}}(\boldsymbol{u},\cdot)$ to be the density with respect to η_r on $r = \mathcal{R}(\boldsymbol{u})$ given by

$$R_{\boldsymbol{u}}(\boldsymbol{u},\boldsymbol{v}) = \frac{f(y)\, q_y(\boldsymbol{v})}{\int_r f(z)\, q_z(\boldsymbol{w})\, d\eta_r(\boldsymbol{w})}\,, \tag{10}$$

with $\boldsymbol{u} = (i,x), \boldsymbol{v} = (j,y)$ and $\boldsymbol{w} = (k,z)$. Alternatively, we may define $R_{\boldsymbol{u}}(\boldsymbol{u},\cdot)$ to be the density with respect to η_r on $r \setminus \{\boldsymbol{u}\}$ given by

$$\frac{s(\boldsymbol{u},\boldsymbol{v})\, f(y)\, q_y(\boldsymbol{v})}{\int_r f(z)\, q_z(\boldsymbol{w})\, d\eta_r(\boldsymbol{w})}\,, \tag{11}$$

and assign probability mass

$$1 - \frac{\int_{r\setminus\{\boldsymbol{u}\}} s(\boldsymbol{u},\boldsymbol{v}) f(y)\, q_y(\boldsymbol{v}) d\eta_r(\boldsymbol{v})}{\int_r f(z)\, q_z(\boldsymbol{w})\, d\eta_r(\boldsymbol{w})}$$

to $\boldsymbol{u}$. The function s must be symmetric and non-negative, and must satisfy

$$\frac{\int_{r\setminus\{\boldsymbol{u}\}} s(\boldsymbol{u},\boldsymbol{v}) f(y)\, q_y(\boldsymbol{v}) d\eta_r(\boldsymbol{v})}{\int_r f(z)\, q_z(\boldsymbol{w})\, d\eta_r(\boldsymbol{w})} \leq 1.$$

Under the assumption above we can now extend Q_x and $R_{\boldsymbol{u}}$ into transition densities Q and R on $\mathcal{U}$ with respect to ξ. One may show, although we shall not do so here, that μ is stationary with respect to Q and R. Consequently, μ is the limiting distribution of the chain produced by Algorithm 2.1, provided the chain is irreducible. One may also show that the R-step is reversible, though the Q-step may not be.

Reversible jump MCMC sampler

In the remainder of this section, we show that the reversible jump MCMC sampler (Green, [6]) is an instance of the new sampler. Let $\mathcal{M}$ be the countable set of move types used in the reversible jump MCMC and let the index set $\mathcal{I}$ be $\mathcal{M} \times \mathcal{X}$. Thus each element of $\mathcal{I}$ consists of a move type $m \in \mathcal{M}$ and a proposed new element $x \in \mathcal{X}$. Put $\mathcal{U} = \mathcal{I} \times \mathcal{X}$ and $\mathcal{Q}(x) = \mathcal{I} \times \{x\}$. Define a probability distribution on $\mathcal{Q}(x)$ with density $q_x(m,y,x) = \sigma_x(m)\, A_m(x,y)$ with respect to some ψ_x, where (m,y,x) is an abbreviation for $((m,y),x)$, $\sigma_x(m)$ is the the probability of selecting move type m when at x, and $A_m(x,\cdot)$ is the density for move-type m at x. Note that by definition

$$\sum_m \int_{\mathcal{X}} q_x(m,y,x)\, d\psi_x(y) = 1, \quad \text{for all } x\ .$$

Define a transition density $Q_x((m_0,y_0,x_0),(m,y,x)) = q_x(m,y,x)$ on $\mathcal{Q}(x)$ and note that q_x is stationary with respect to Q_x. Extend Q_x to a transition density Q on $\mathcal{U}$. Selecting a new element $(m, y, x) \in \mathcal{U}$ in accordance with Q is equivalent to selecting a move type m with probability $\sigma_x(m)$ and proposing an element y in accordance with the density A_m, as in the reversible jump MCMC. This completes the Q-step.

For the R-step, we must first define a reference measure ξ on $\mathcal{U}$. We assume, as in Green [6], that there is a symmetric joint measure ξ_m on $\mathcal{X} \times \mathcal{X}$ for each move type m. Define $\xi(A) = \sum_m \xi_m(A \cap \mathcal{U}_m)$ for all measurable sets $A \subseteq \mathcal{U}$, where $\mathcal{U}_m = \{(m, y, x) \in \mathcal{U} : x, y \in \mathcal{X}\} \subset \mathcal{U}$. Note that ξ_m may legitimately be regarded as a measure on $\mathcal{U}_m$, since that space is isomorphic to $\mathcal{X} \times \mathcal{X}$. Let f be the finite density of the target distribution and define $f_m(x, y) = f(x)A_m(x, y)$. We may now define a measure μ with density $\sigma_m(x)f_m(x, y)$ at (m, y, x) with respect to ξ.

Next, define a partition of $\mathcal{U}$ consisting of the sets $\mathcal{R}(m, y, x) = \{(m, y, x), (m, x, y)\}$ for all $(m, y, x) \in \mathcal{U}$. It can be shown, using the symmetry of ξ_m, that ξ decomposes into counting measures on the sets $\mathcal{R}(m, y, x)$. That is, $\eta_r(\{(m, y, x)\}) = \eta_r(\{(m, x, y)\}) = 1$ for all $r \in \mathcal{R}$. Consequently, putting

$$s((m, y, x), (m, x, y)) = \min\left\{1 + \frac{\sigma_m(x)f_m(x, y)}{\sigma_m(y)f_m(y, x)}, 1 + \frac{\sigma_m(y)f_m(y, x)}{\sigma_m(x)f_m(x, y)}\right\}$$

and substituting this into (11) gives the transition matrix

$$R_{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{v}) = \begin{cases} \alpha_m(x, y) & \text{if } \boldsymbol{v} = \boldsymbol{u} \equiv (m, y, x) \\ 1 - \alpha_m(x, y) & \text{if } \boldsymbol{v} = (m, x, y) \end{cases}$$

where

$$\alpha_m(x, y) = \min\left\{1, \frac{\sigma_m(y)f_m(y, x)}{\sigma_m(x)f_m(x, y)}\right\}.$$

Thus, selecting a new element (m, x, y) in accordance with the global transition matrix R (defined as in (8)) is equivalent to accepting the proposed y with probability $\alpha_m(x, y)$, as in the reversible jump MCMC.

5 Example: Isochore delineation

In this section, the generalised Gibbs sampler is used to segment a long DNA sequence into intervals of approximately uniform composition. The genomes of complex organisms, including the human genome, are known to vary in GC content along their length. That is, they vary in the local proportion of the nucleotides G and C, as opposed to the nucleotides A and T. (The reason that G and C are grouped together, and A and T are grouped together, is that DNA is a double-stranded molecule in which G's on one strand bind to C's on the other, and similarly for A's and T's. Thus the proportions of G and C are always equal in double-stranded DNA, as are the proportions of A and T. These equalities are known as *Chargaff's rules.*) Changes in GC content are often abrupt, producing well-defined regions called *isochores*. An attempt to delineate isochores in several genomes is documented in [17] and the website `http://bioinfo2.ugr.es/isochores`. The example given here is not so ambitious, although the approach we develop seems a promising one for large-scale studies of isochore structure. We model the problem as a multiple change-point problem, that is, a problem in which sequential data is separated into segments by an unknown number of change-points, with each segment supposed to have been generated by a different process. Multiple change-point problems have previously been used to illustrate model-switching samplers [6, 19].

Firstly, let us formulate the problem in mathematical terms. A sequence $a = \{a_1, \ldots, a_L\}$ of length L is given, where $a_m \in \{A, C, G, T\}$. The sequence may be converted to a binary sequence $b = \{b_1, \ldots, b_L\}$ in which $b_m = 1$ if $a_m \in \{C, G\}$ and $b_m = 0$ otherwise. A segmentation of the sequence is specified by giving the number of change-points N and the positions of the change-points $\{c_1, \ldots, c_N\}$, where $1 < c_1 < \ldots < c_N \leq L$. In this context, a *change-point* is a boundary between two adjacent segments, and the value c_n is the sequence position of the leftmost character of the segment to the right of the nth change-point. A maximum number of change-points N_{max} is specified, where $0 \leq N \leq N_{max} \leq L$. It will also be convenient to define $c_0 = 1$ and $c_{N+1} = L + 1$. The model here assumed is that within each segment characters are generated by independent Bernoulli trials with probability of success (that is, a '1') θ_n, where $0 < \theta_n < 1$. (For brevity, we henceforth refer to the probability of success for a given segment as the Bernoulli parameter for that segment.) Thus a complete model of the process by which the sequence was generated consists of the elements $(N, c_1, \ldots, c_N, \theta_0, \ldots, \theta_N)$ and the space of all such models is $\mathcal{X} = \cup_{N=0}^{N_{max}} \{N\} \times \mathcal{C}_N \times (0,1)^{N+1}$, where $\mathcal{C}_N = \{(c_1, \ldots, c_N) \in \{2, \ldots, L\}^N : c_1 < \ldots < c_N\}$. We represent an element of $\mathcal{X}$ by (N, c, θ), where $c \in \mathcal{C}_N$ and $\theta \in (0,1)^{N+1}$.

To formulate the problem in terms of a Bayesian model, a prior distribution must be defined on $\mathcal{X}$. As a prior distribution on the number of change-points we take a truncated Poisson distribution. We assume a uniform prior on $\mathcal{C}_N$ and uniform priors on $(0,1)$ for each θ_n. Thus the overall prior is proportional to

$$\frac{\lambda^N}{N!} \binom{L-1}{N}^{-1},$$

at $x = (N, c, \theta)$, where λ is a hyper-parameter which is taken as given. The posterior distribution is therefore

$$f(x) \propto \lambda^N (L - 1 - N)! \prod_{n=0}^{N} \theta_n^{\mathbb{I}(c_n, c_{n+1})} (1 - \theta_n)^{\mathbb{O}(c_n, c_{n+1})},$$

where $\mathbb{I}(c_n, c_{n+1})$ is the number of ones in the segment bounded by sequence positions c_n and $c_{n+1} - 1$ and $\mathbb{O}(c_n, c_{n+1})$ the number of zeros in that same segment. Note that f is a density with respect to the implicitly assumed reference measure

$$\sum_{N=0}^{\infty} \sum_{c \in \mathcal{C}_N} \mathrm{Leb}_{N+1}(A \cap \mathcal{X}_{N,c}),$$

for measurable A, where Leb_{N+1} is the Lebesgue measure on $\mathcal{X}_{N,c} := \{(N, c, \theta) : \theta \in (0,1)^{N+1}\} \cong (0,1)^{N+1}$.

We here use the generalised Gibbs sampler in the context of simulated annealing to determine the maximum of f, attained at the “best model” x^*. However, the method can in principle provide substantially more information than this, such as uncertainties about the number and positions of the change-points.

A Gibbs-like sampler for this problem may be constructed as follows. Two broad classes of moves are allowed, labelled ‘I’ and ‘D’. The former considers the insertion of a new change-point, and the latter the deletion of one. For a model x with N change-points, there are $N+1$ segments into which a new change-point may be inserted, and N change-points that may be deleted. There are thus $2N+1$ move types, which we label $(0,I),\ldots,(N,I)$ and $(1,D),\ldots,(N,D)$. We therefore set $\mathcal{Q}(x) = \{(n,I,x) : n = 0,\ldots,N\} \cup \{(n,D,x) : n = 1,\ldots,N\}$. Note that $((n,D),x)$ and $((n,I),x)$ have been abbreviated to (n,I,x) and (n,D,x). Note that the state space is $\mathcal{U} = \cup_{x\in\mathcal{X}}\mathcal{Q}(x)$.

For the Q-step, define a transition matrix

$$Q_x((i,x),(j,x)) = \begin{cases} 1 & \text{if } i=(n,I),\ j=(n+1,D),\ n=0,\ldots,N-1 \\ 1 & \text{if } i=(N,I),\ j=(0,I) \\ 1 & \text{if } i=(n,D),\ j=(n,I),\ n=1,\ldots,N \\ 0 & \text{otherwise.} \end{cases}$$

Thus Q_x cycles through the $2N+1$ move types available at x. A global transition matrix Q can now be defined as in Section 2 for the Q-step. Note that the density of the stationary distribution of Q_x is $q_x(\boldsymbol{u}) = 1/(2N+1)$ for all $\boldsymbol{u} \in \mathcal{Q}(x)$.

For the R-step, define $\mathcal{R}(n,I,x)$ to be the set of models obtained by changing only a single segment of x, either by changing the Bernoulli parameter for that segment, or by splitting it into two segments with separate Bernoulli parameters. For given n and x, we define $r_-(n,x)$ to be the set of models obtained by changing the Bernoulli parameter for segment n, and $r_{c_*}(n,x)$ to be the set of models obtained by inserting a change-point at $c_* \in \{c_n+1,\ldots,c_{n+1}-1\}$ and setting two new Bernoulli parameters for the two segments thus created. Thus,

$$\mathcal{R}(n,I,x) = r_-(n,x) \bigcup_{c_*=c_n+1}^{c_{n+1}-1} r_{c_*}(n,x),$$

if x has fewer than N_{max} change-points and

$$\mathcal{R}(n,I,x) = r_-(n,x)$$

if x has N_{max} change-points. Note that $\mathcal{R}(n,D,x) = \mathcal{R}(n-1,I,y)$ for some $y \in \mathcal{X}$. (A suitable y may be obtained by deleting change-point n and choosing a new Bernoulli parameter for the segment thus created.) Also note that $r_-(n,x) \cong (0,1)$ and $r_{c_*}(n,x) \cong (0,1)^2$.

Define μ to be the measure on $\mathcal{U}$ with density $g(i,x) = f(x)\,q_x(i)$. Note that this density is defined with respect to the implicitly assumed reference measure

$$\xi(A) = \sum_{N=0}^{\infty} \sum_{c\in\mathcal{C}_N} \sum_i \mathrm{Leb}_{N+1}(A \cap \mathcal{U}_{i,N,c}),$$

for measurable A, where

$$\mathcal{U}_{i,N,c} = \{(i,x) \in \mathcal{U} : x = (N,c,\theta), \theta \in (0,1)^{N+1}\} \cong (0,1)^{N+1}.$$

The sum indexed by i is over the $2N+1$ move types $i = (0,I), \dots, (N,I), (1,D), \dots, (N,D)$. This reference measure on $\mathcal{U}$ can be decomposed into reference measures η_r on the sets $r = \mathcal{R}(n,I,x)$, given by

$$\eta_r(A) = \mathrm{Leb}(A \cap r_-(n,x)) + \sum_{c_*=c_n+1}^{c_{n+1}-1} \mathrm{Leb}_2(A \cap r_{c_*}(n,x)).$$

Thus, we may take the densities in the R-step proportional to g. In practice, we compute the integral $w_-(n,x)$ of g over $r_-(n,x)$, and integrals $w_{c_*}(n,x)$ of g over $r_{c_*}(n,x)$, for each $c_* \in \{c_n+1, \dots, c_{n+1}-1\}$. These integrals can be expressed in terms of gamma functions, and the normalisation constant is their sum. As an example, consider $w_-(k,x)$, where x has fewer than N_{max} changepoints. We have

$$w_-(k,x) = \frac{\lambda^N (L-1-N)!}{2N+1} \prod_{\substack{n=0\\ n\neq k}}^{N} \theta_n^{\mathbb{I}(c_n,c_{n+1})}(1-\theta_n)^{\mathbb{O}(c_n,c_{n+1})}$$
$$\times \int_0^1 \theta^{\mathbb{I}(c_k,c_{k+1})}(1-\theta)^{\mathbb{O}(c_k,c_{k+1})}\, d\theta\ .$$

When x has $N < N_{max}$, the R-step can be implemented as follows:

1. Calculate the weights $w_-(n,x)$ and $w_{c_*}(n,x)$ for $c_* = c_n+1, \dots, c_{n+1}-1$.

2. Select an element of $\{-, c_n+1, \dots, c_{n+1}-1\}$ with probabilities proportional to the weights calculated in Step 1.

3. If '$-$' is selected, update θ_n by sampling from a beta distribution with parameters $\alpha = \mathbb{I}(c_n,c_{n+1})+1$ and $\beta = \mathbb{O}(c_n,c_{n+1})+1$. Otherwise, if c_* is selected, insert a new change-point at c_* and select new Bernoulli parameters for the segments to the left and right of c_* by sampling from beta distributions with parameters $(\alpha = \mathbb{I}(c_n,c_*)+1, \beta = \mathbb{O}(c_n,c_*)+1)$ and $(\alpha' = \mathbb{I}(c_*,c_{n+1})+1, \beta' = \mathbb{O}(c_*,c_{n+1})+1)$ respectively.

When x has N_{max} change-points, the R-step is simply a conventional Gibbs coordinate update in which a new value for θ_n is generated by sampling from a beta distribution with parameters $\alpha = \mathbb{I}(c_n,c_{n+1})+1$ and $\beta = \mathbb{O}(c_n,c_{n+1})+1$.

The sampler was tested on a long DNA sequence containing a region known as the human major histocompatibility region. This sequence contains approximately 3.5 million characters and is of interest here because it contains two well-characterised isochores (see Oliver *et al.* [17]). The sampler just described was used in the context of simulated annealing to determine the most probable segmentation. The value of N_{max} was 1000, and results were obtained with $\lambda = 10^{-10}$ and $\lambda = 10^{-200}$. Such small values of λ impose a heavy penalty on

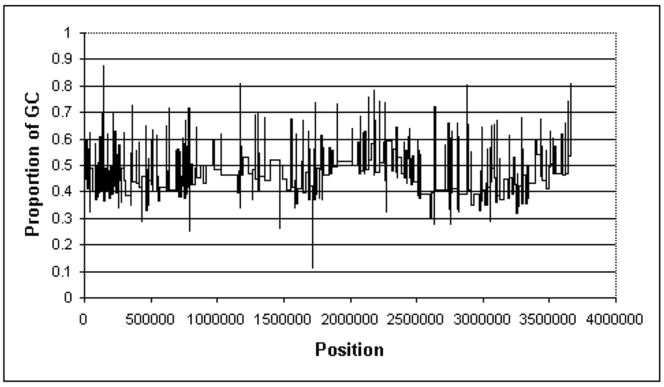

Figure 8: Optimal segmentation with $\lambda = 10^{-10}$

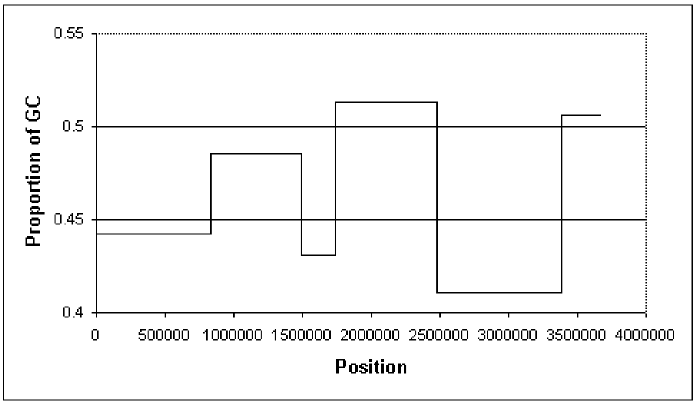

Figure 9: Optimal segmentation with $\lambda = 10^{-200}$

the addition of change-points, so a change-point will only appear if it is very well supported by the data. The results are shown in Figure 8 and Figure 9.

The value plotted for each segment is the most probable Bernoulli parameter for that segment, and may be interpreted as the GC proportion for that segment. We note that the segmentation for the smaller value of λ (Figure 9) is in excellent agreement with that obtained by Oliver *et al.* [17]. We also note that the segmentation for the larger value of λ (Figure 8) contains many more change-points. This indicates the existence of smaller segments with well-defined GC content. We suggest that a hierarchical segmentation model would fit the data better, with the traditional notion of an isochore corresponding to the highest level (coarsest) segmentation. This is consistent with the findings of other studies (eg. IHGSC [9]).

6 Concluding Remarks

The generalised Gibbs sampler enables Gibbs-like sampling in more general spaces than is possible with the conventional Gibbs sampler. In particular, it allows sampling from probability spaces in which there is uncertainty about not only the model parameters, but also the model itself. The new sampler therefore enables the Gibbs sampling approach to be used for model determination, evaluation and averaging.

The new sampler has turned out to be more than the generalised Gibbs sampler it was originally intended to be, in that it encompasses as particular cases all the Markov chain samplers mentioned in the introduction, including the reversible jump MCMC. The examples considered here were deliberately Gibbs-like, but it may be that many instances that do not resemble the Gibbs sampler remain to be explored.

Acknowledgements

The authors would like to thank Mark Springer for providing the mammalian sequences referred to in this paper.

References

[1] Barker, A.A., “Monte Carlo calculations of the radial distribution functions for a proton-electron plasma,” Aust. J. Phys. 18, 119–133, 1965.

[2] Carlin, B.P., and Chib, S., “Bayesian model choice via Markov chain Monte Carlo,” J. Am. Statist. Assoc., 88, 309-319.

[3] Fitch, W.M., “Toward defining the course of evolution: minimum change for a specific tree topology,” Syst. Zool. 20(4), 406-416, 1971.

[4] Gelfand, A.F. and Smith, A.F.M., “Sampling-based approaches to calculating marginal densities,” J. Am. Stat. Assoc. 85, 398–409, 1990.

[5] Geman, S. and Geman, D., “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” IEEE T. Pattern Anal. 6, 721–741, 1984.

[6] Green, P.J., “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” Biometrika 82(4), 711-732, 1995.

[7] Hastings, W.K., “Monte Carlo sampling methods using Markov chains and their applications,” Biometrika 57(1), 97–109, 1970.

[8] Huelsenbeck, J.P., Ronquist, F., “MrBayes: Bayesian inference of phylogenetic trees,” Bioinformatics 17 (8), 754 – 755, 2001.

[9] International Human Genome Sequencing Consortium, “Initial sequencing and analysis of the human genome,” Nature 409, 860-921, 2001.

[10] Keith, J.M., Adams, P., Bryant, D., Kroese, D.P., Mitchelson, K.R., Cochran, D.A.E., Lala, G.H., "A simulated annealing algorithm for finding consensus sequences," Bioinformatics 18, 1494-1499, 2002.

[11] Keith, J.M., Adams, P., Bryant, D., Mitchelson, K.R., Cochran, D.A.E., Lala, G.H., "Inferring an original sequence from erroneous copies: a Bayesian approach," Proceedings of the 1st Asia-Pacific Bioinformatics Conference (APBC2003), Conferences in Research and Practice in Information Technology, Vol. 19, Ed: Yi-Ping Phoebe Chen, 23-28, 2003.

[12] Keith, J.M., Adams, P., Bryant, D., Mitchelson, K.R., Cochran, D.A.E., Lala, G.H., "Inferring an original sequence from erroneous copies: two approaches," Asia-Pacific BioTech News 7(3), 107-114, 2003.

[13] Larget, B. and Simon, D., "Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees," Mol. Biol. Evol. 16, 750–759, 1999.

[14] Mau, B., and Newton, M., "Phylogenetic inference for binary data on dendrograms using Markov chain Monte Carlo," J. Comput. Graph. Stat. 6, 122–131, 1997.

[15] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., "Equations of State Calculations by Fast Computing Machines," J. Chem. Phys. 21(6), 1087–1092, 1953.

[16] Murphy, W.J., Eizirik, E., O'Brien, S.J., Madsen, O., Scally, M., Douady, C.J., Teeling, E., Ryder, O.A., Stanhope, M.J., de Jong, W.W., Springer, M.S., "Resolution of the early placental mammal radiation using Bayesian phylogenetics," Science, 294, 2348-2351.

[17] Oliver, J.L., Bernaolo-Galvin, P., Carpena, P., Roman-Roldan, R., "Isochore chromosome maps of eukaryotic genomes," Gene 276, 47-56, 2001.

[18] Page, R.D.M., "TREEVIEW: An application to display phylogenetic trees on personal computers," Computer Applications in the Biosciences 12, 357–358, 1996.

[19] Phillips, D.B., and Smith, A.F.M., "Bayesian model comparison via jump diffusions," in Markov Chain Monte Carlo in Practice, Ed. Gilks, W.R., Richardson, S.T., and Spiegelhalter, D.J., Chapman and Hall, 1995.

[20] Yang, Z., and Rannala, B., "Bayesian phylogenetic inference using DNA sequences: a Markov chain Monte Carlo method," Mol. Biol. Evol. 14, 717–724, 1997.