The Generalized Gibbs Sampler and the Neighborhood Sampler

Jonathan M. Keith[1], George Yu. Sofronov[2], and Dirk P. Kroese[3]

[1] Department of Mathematics, University of Queensland, St. Lucia, Qld. 4072, Australia j.keith1@uq.edu.au
[2] Department of Mathematics, University of Queensland, St. Lucia, Qld. 4072, Australia georges@maths.uq.edu.au
[3] Department of Mathematics, University of Queensland, St. Lucia, Qld. 4072, Australia kroese@maths.uq.edu.au

The Generalized Gibbs Sampler (GGS) is a recently proposed Markov chain Monte Carlo (MCMC) technique that is particularly useful for sampling from distributions defined on spaces in which the dimension varies from point to point or in which points are not easily defined in terms of co-ordinates. Such spaces arise in problems involving model selection and model averaging and in a number of interesting problems in computational biology. Such problems have hitherto been the domain of the Reversible-jump Sampler, but the method described here, which generalizes the well-known conventional Gibbs Sampler, provides an alternative that is easy to implement and often highly efficient.

The GGS provides a very general framework for MCMC simulation. Not only the conventional Gibbs Sampler, but also a variety of other well known samplers emerge as special cases. These include the Metropolis-Hastings sampler, the Reversible-jump Sampler and the Slice Sampler. We also present a new special case of the GGS, called the Neighborhood Sampler (NS), which does not conform to any of the other existing MCMC frameworks. We illustrate use of the GGS and the NS with a number of examples. In particular, we use the NS to sample from a discrete state space represented as a graph in which nodes have varying degree. Finally, we introduce a technique for improving convergence and mixing between sub-spaces of different dimension.

1 Markov Samplers

MCMC is a technique for approximately sampling from a target distribution with pdf f. Almost any distribution can be sampled via MCMC, and often it is the only technique capable of generating a sample from a given distribution. There are numerous types of MCMC sampler, but the two most frequently

encountered in practice are the Metropolis-Hastings algorithm [13, 5] and the Gibbs sampler [3, 2]. Another sampler, which is distinct from these and gaining in importance, is the slice sampler [14]. The slice sampler is often the most efficient MCMC method for sampling from one-dimensional distributions, and it is used as the standard technique for sampling from a general distribution in a one-dimensional sub-space in the popular MCMC software package BUGS (available at `http://www.mrc-bsu.cam.ac.uk/bugs/`).

1.1 The Discrete Case

All of the above-mentioned MCMC algorithms can be described within a framework that we call the Generalized Gibbs Sampler (GGS), although we have also called it the Generalized Markov Sampler [8]. The authors have used the GGS to develop highly efficient new samplers for a range of problems arising in computational biology [6, 7, 8, 9, 10, 11].

Consider a Markov chain $\{(\mathbf{X}_n, \mathbf{Y}_n), n = 0, 1, 2, \ldots\}$ on the set $\mathscr{X} \times \mathscr{Y}$, where $\mathscr{X}$ is the target set and $\mathscr{Y}$ is an auxiliary set. For the sake of simplicity, we initially assume that both $\mathscr{X}$ and $\mathscr{Y}$ are finite. We extend the GGS to the general case in Section 1.2. Let $f(\mathbf{x})$ be the target pdf, defined on $\mathscr{X}$. Each transition of the Markov chain consists of two parts. The first is $(\mathbf{x}, \tilde{\mathbf{y}}) \to (\mathbf{x}, \mathbf{y})$, according to a transition matrix $\mathbf{Q}$; the second is $(\mathbf{x}, \mathbf{y}) \to (\mathbf{x}', \mathbf{y}')$, according to a transition matrix $\mathbf{R}$. In other words, the transition matrix $\mathbf{P}$ of the Markov chain is given by the product $\mathbf{Q}\,\mathbf{R}$. Both steps are illustrated in Figure 1, and further explained below.

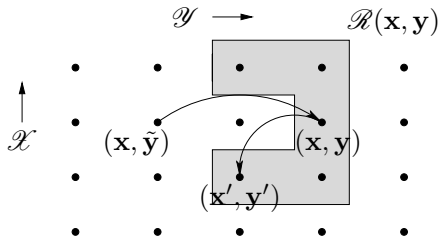

Fig. 1. Each transition of the Markov chain consists of two steps: the Q-step, followed by the R-step.

The first step, the *Q-step*, only changes the $\mathbf{y}$-coordinate, but leaves the $\mathbf{x}$-coordinate as it is. In particular, $\mathbf{Q}$ is of the form $\mathbf{Q}((\mathbf{x}, \tilde{\mathbf{y}}), (\mathbf{x}, \mathbf{y})) = \mathbf{Q}_{\mathbf{x}}(\tilde{\mathbf{y}}, \mathbf{y})$, where $\mathbf{Q}_{\mathbf{x}}$ is a transition matrix on $\mathscr{Y}$. Let $q_{\mathbf{x}}$ be a stationary distribution for $\mathbf{Q}_{\mathbf{x}}$, assuming that this exists.

The second step, the *R-step*, is determined by (a) the stationary distribution $q_{\mathbf{x}}$ and (b) a partition of the set $\mathscr{X} \times \mathscr{Y}$. Specifically, we define for each point $(\mathbf{x}, \mathbf{y})$ a set $\mathscr{R}(\mathbf{x}, \mathbf{y})$ containing $(\mathbf{x}, \mathbf{y})$ such that *if* $(\mathbf{x}', \mathbf{y}') \in \mathscr{R}(\mathbf{x}, \mathbf{y})$

then $\mathscr{R}(\mathbf{x}', \mathbf{y}') = \mathscr{R}(\mathbf{x}, \mathbf{y})$; see Figure 1, where the shaded area indicates the neighborhood set of $(\mathbf{x}, \mathbf{y})$. The crucial step is now to define the transition matrix $\mathbf{R}$ as

$$\mathbf{R}[(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')] = c(\mathbf{x}, \mathbf{y})\, f(\mathbf{x}')\, q_{\mathbf{x}'}(\mathbf{y}'), \quad \text{for all} \quad (\mathbf{x}', \mathbf{y}') \in \mathscr{R}(\mathbf{x}, \mathbf{y}),$$

where $c(\mathbf{x}, \mathbf{y})$ is a normalisation constant. Note that $c(\mathbf{x}', \mathbf{y}') = c(\mathbf{x}, \mathbf{y})$ if $(\mathbf{x}', \mathbf{y}') \in \mathscr{R}(\mathbf{x}, \mathbf{y})$. The distribution

$$\mu(\mathbf{x}, \mathbf{y}) = f(\mathbf{x})\, q_{\mathbf{x}}(\mathbf{y}), \tag{1}$$

is trivially stationary with respect to $\mathbf{Q}$, and satisfies detailed balance with respect to $\mathbf{R}$, and hence is a stationary distribution with respect to the Markov chain. It will also be the limiting distribution, provided that the chain is ergodic. In particular, by ignoring the $\mathbf{y}$-coordinate, we see that the limiting pdf of $\mathbf{X}_n$ is the required target $f(\mathbf{x})$. This leads to the following algorithm:

Algorithm 1.1 (Generalized Gibbs Sampler) Starting with an arbitrary $(\mathbf{X}_0, \mathbf{Y}_0)$, perform the following steps iteratively:

[Q**-step:**] Given $(\mathbf{X}_n, \mathbf{Y}_n)$, generate $\mathbf{Y}$ from $\mathbf{Q}_{\mathbf{x}}(\mathbf{Y}_n, \mathbf{y})$.

[R**-step:**] Given $\mathbf{Y}$ generate $(\mathbf{X}_{n+1}, \mathbf{Y}_{n+1})$ from $\mathbf{R}[(\mathbf{X}_n, \mathbf{Y}), (\mathbf{x}, \mathbf{y})]$.

Denoting $\mathscr{R}^-(\mathbf{x}, \mathbf{y}) = \mathscr{R}(\mathbf{x}, \mathbf{y}) \setminus \{(\mathbf{x}, \mathbf{y})\}$, the sampler can be generalized further (without disturbing detailed balance) by redefining $\mathbf{R}$ as:

$$\mathbf{R}[(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')] = \begin{cases} s((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}'))\, f(\mathbf{x}')\, q_{\mathbf{x}'}(\mathbf{y}') & \text{if } (\mathbf{x}', \mathbf{y}') \in \mathscr{R}^-(\mathbf{x}, \mathbf{y}) \\ 1 - \displaystyle\sum_{(\mathbf{z}, \mathbf{w}) \in \mathscr{R}^-(\mathbf{x}, \mathbf{y})} \mathbf{R}[(\mathbf{x}, \mathbf{y}), (\mathbf{z}, \mathbf{w})] & \text{if } (\mathbf{x}', \mathbf{y}') = (\mathbf{x}, \mathbf{y}) \end{cases} \tag{2}$$

where s is any symmetric function such that the quantities above are indeed probabilities.

1.2 GGS in a General Space

In order to relax the requirement that $\mathscr{X}$ and $\mathscr{Y}$ be finite, one must first specify the reference measure ϕ with respect to which the target density f is defined, and the reference measures $\psi_{\mathbf{x}}$ with respect to which the densities $\mathbf{Q}_{\mathbf{x}}$ and $q_{\mathbf{x}}$ are defined. Moreover, one must specify reference measures η_r with respect to which the density $\mathbf{R}[(\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')]$ will be defined for each set $r = \mathscr{R}(\mathbf{x}, \mathbf{y})$. It will be necessary to make the following assumption:

Assumption: There exists a measure ζ for the set $\mathscr{R} = \{\mathscr{R}(\mathbf{x}, \mathbf{y}) : (\mathbf{x}, \mathbf{y}) \in \mathscr{X} \times \mathscr{Y}\}$ such that the measure $\phi(d\mathbf{x})\psi_{\mathbf{x}}(d\mathbf{y})$ has a finite density $g(\mathbf{x}, \mathbf{y})$ with respect to the measure $\zeta(dr)\eta_r(d\mathbf{x}, d\mathbf{y})$.

We can now define

$$\mathbf{R}[(\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}')] = c(\mathbf{x},\mathbf{y})\, f(\mathbf{x}')\, q_{\mathbf{x}'}(\mathbf{y}')\, g(\mathbf{x}',\mathbf{y}'), \quad \text{for all} \quad (\mathbf{x}',\mathbf{y}') \in \mathscr{R}(\mathbf{x},\mathbf{y}),$$

where again $c(\mathbf{x},\mathbf{y})$ is a normalisation constant. More generally, we can define:

$$\mathbf{R}[(\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}')] = \begin{cases} s((\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}'))\, f(\mathbf{x}')\, q_{\mathbf{x}'}(\mathbf{y}')\, g(\mathbf{x},\mathbf{y}) & \text{if } (\mathbf{x}',\mathbf{y}') \in \mathscr{R}^{-}(\mathbf{x},\mathbf{y}) \\ 1 - \displaystyle\sum_{(\mathbf{z},\mathbf{w}) \in \mathscr{R}^{-}(\mathbf{x},\mathbf{y})\}} \mathbf{R}[(\mathbf{x},\mathbf{y}),(\mathbf{z},\mathbf{w})] & \text{if } (\mathbf{x}',\mathbf{y}') = (\mathbf{x},\mathbf{y}) \end{cases} \tag{3}$$

where s is any symmetric function such that $\mathbf{R}$ is indeed a pdf.

Note that if $\mathscr{X}$ and $\mathscr{Y}$ are finite, then all of the above measures may be assumed to be counting measures. Moreover, in that case ζ always exists (it, too, is a counting measure) and $g(\mathbf{x},\mathbf{y}) = 1$.

2 Special cases

The GGS framework makes it possible to obtain many different samplers in a simple and unified manner; we give the slice sampler as an example. Other instances of the GCS include (see [8]) the Metropolis-Hastings sampler, the Gibbs sampler and the Reversible-jump sampler [4]. Here, we also introduce a new sampler — the Neighborhood sampler — and use it to sample from a discrete space represented as a graph.

2.1 Slice Sampler

The slice sampler [14] has numerous variants, all of which can be conveniently described within the GGS framework. Here we discuss a fairly general form of the slice sampler. Suppose we wish to generate samples from the pdf

$$f(\mathbf{x}) = b \prod_{k=1}^{m} f_k(\mathbf{x}), \tag{4}$$

where b is a known or unknown constant, and the $\{f_k\}$ are known positive functions — not necessarily densities. We employ Algorithm 1.1, where at the Q-step we generate, for a given $\mathbf{X} = \mathbf{x}$, a vector $\mathbf{Y} = (Y_1, \ldots, Y_m)$ by independently drawing each component Y_k from the uniform distribution on $[0, f_k(\mathbf{x})]$. Thus, $q_{\mathbf{x}}(\mathbf{y}) = 1/\prod_{k=1}^{m} f_k(\mathbf{x}) = b/f(\mathbf{x})$. Secondly, we let $\mathscr{R}(\mathbf{x},\mathbf{y}) = \{(\mathbf{x}',\mathbf{y}) : f_k(\mathbf{x}') \geqslant y_k,\ k = 1,\ldots,m\}$. Then, (note that $f(\mathbf{x}')\, q_{\mathbf{x}'}(\mathbf{y}) = b$)

$$\mathbf{R}[(\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y})] = \frac{1}{|\mathscr{R}(\mathbf{x},\mathbf{y})|}\,.$$

where $|\mathscr{A}|$ means the measure of set $\mathscr{A}$: the cardinality in the discrete case, or the area/volume in the continuous case. In other words, in the R-step, given

$\mathbf{x}$ and $\mathbf{y}$, we draw $\mathbf{X}'$ uniformly from the set $\{\mathbf{x}' : f_k(\mathbf{x}') \geqslant y_k, k = 1, \ldots, m\}$. This gives the following *slice sampler*, in which N is a predetermined number of iterations:

Algorithm 2.1 (Slice Sampler) *Let $f(\mathbf{x})$ be of the form (4).*

1. *Initialize $\mathbf{X}_0$. Set $t = 1$.*
2. *For $k = 1$ to m draw $U_k \sim \mathsf{U}(0, 1)$ and let $Y_k = U_k\, f_k(\mathbf{x}_{t-1})$.*
3. *Draw $\mathbf{X}_t$ uniformly from the set $\{\mathbf{x} : f_k(\mathbf{x}) \geqslant Y_k, k = 1, \ldots, m\}$.*
4. *If $t = N$ stop. Otherwise set $t = t + 1$ and repeat from step 2.*

Suppose we want to generate a sample from the target pdf

$$f(x) = c\,\frac{x\,\mathrm{e}^{-x}}{1+x}, \quad x \geqslant 0\,,$$

using the slice sampler with $f_1(x) = x/(1+x)$ and $f_2(x) = \mathrm{e}^{-x}$. Suppose that at iteration t, $X_{t-1} = z$, and u_1 and u_2 are generated in step 2. In step 3, X_t is drawn uniformly from the set $\{x : f_1(x)/f_1(z) \geqslant u_1,\ f_2(x)/f_2(z) \geqslant u_2\}$, which implies the bounds $x \geqslant \frac{u_1\, z}{1+z-u_1\, z}$, and $x \leqslant z - \ln u_2$. Since for $z > 0$ and $0 \leqslant u_1, u_2 \leqslant 1$, the latter bound is larger than the former, the interval to be drawn from in step 3 is $(\frac{u_1\, z}{1+z-u_1\, z},\ z - \ln u_2)$. Figure 2 depicts a histogram of $N = 10^5$ samples generated via the slice sampler, along with the true pdf $f(x)$. We see that the two are in close agreement.

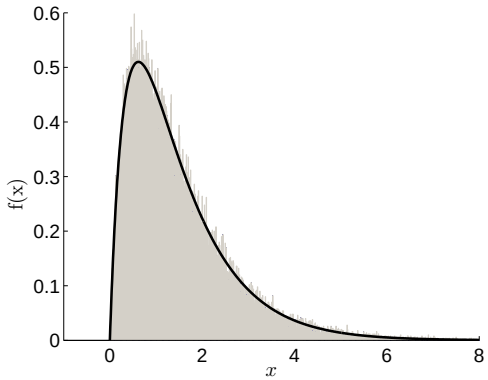

Fig. 2. True density and histogram of samples produced by the slice sampler.

2.2 The Neighborhood Sampler

The Neighborhood Sampler (NS) is a new instance of the GGS that resembles the slice sampler and in certain cases corresponds to it. The NS can be used to sample from a target distribution f on some measure space $(\mathscr{X}, \Sigma, \mu)$

consisting of a target space $\mathscr{X}$ with σ-algebra Σ and measure μ. The aim of the NS to reduce sampling from a complicated distribution function f to sampling from uniform distributions over local neighborhoods. To construct a NS, we must first assign a unique neighborhood $\mathcal{N}_{\mathbf{x}}$ to each element $\mathbf{x} \in \mathscr{X}$. These neighborhoods must satisfy the following three conditions:

1. $\mathbf{x} \in \mathcal{N}_{\mathbf{x}}$ for all $\mathbf{x} \in \mathscr{X}$,
2. $0 < \mu(\mathcal{N}_{\mathbf{x}}) < \infty$ for all $\mathbf{x} \in \mathscr{X}$, and
3. $\mathbf{y} \in \mathcal{N}_{\mathbf{x}}$ iff $\mathbf{x} \in \mathcal{N}_{\mathbf{y}}$ for all $\mathbf{x}, \mathbf{y} \in \mathscr{X}$.

In what follows we use the notation $\mathcal{N}(\mathbf{x})$ synonymously with $\mathcal{N}_{\mathbf{x}}$ to avoid placing subscripts on subscripts.

To sample from an arbitrary distribution having density f with respect to μ, the NS consists of the following steps performed iteratively, starting with an arbitrary element $\mathbf{x}_0$ and with $t = 0$:

Algorithm 2.2 (Neighborhood Sampler)
Given the current state $\mathbf{X}_t = \mathbf{x}$*:*

1. *Generate* $\mathbf{Y} \sim \mathsf{U}(\mathcal{N}_{\mathbf{x}})$ *where* $\mathsf{U}(\mathcal{N}_{\mathbf{x}})$ *is the uniform distribution (with respect to* μ*) on* $\mathcal{N}_{\mathbf{x}}$*. Set* $H = \mathcal{N}_{\mathbf{Y}}$*.*
2. *Generate* $\mathbf{U} \sim \mathsf{U}(0, f(\mathbf{x})/\mu[\mathcal{N}_{\mathbf{x}}])$*.*
3. *Generate* $\mathbf{Z}_1 \sim \mathsf{U}(H)$*.*
4. *Set* $k = 1$ *and iterate the following steps until* $f(\mathbf{Z}_k)/\mu[\mathcal{N}(\mathbf{Z}_k)] \geqslant \mathbf{U}$*:*
   a) *Optionally reduce* H *so that it excludes* $\mathbf{Z}_k$ *while still containing* $\mathbf{x}$*.*
   b) *Generate* $\mathbf{Z}_{k+1} \sim \mathsf{U}(H)$ *and set* $k := k + 1$*.*
5. *Set* $\mathbf{X}_{t+1} = \mathbf{Z}_k$*.*

The reduction in Step 4a) must be done so that $H(\mathbf{x}', \mathbf{y}, \mathbf{z}_1, \ldots, \mathbf{z}_k) = H(\mathbf{x}, \mathbf{y}, \mathbf{z}_1, \ldots, \mathbf{z}_k)$ for all $\mathbf{x}' \in H(\mathbf{x}, \mathbf{y}, \mathbf{z}_1, \ldots, \mathbf{z}_k)$, where the notation $H(\mathbf{x}, \mathbf{y}, \mathbf{z}_1, \ldots, \mathbf{z}_k)$ indicates the neighborhood obtained by reducing $\mathcal{N}_{\mathbf{y}}$ in such a way as to include $\mathbf{x}$ and exclude $\mathbf{z}_1, \ldots, \mathbf{z}_k$. The details of this reduction depend on the specific application. We give some examples below.

If we do not implement the reduction of H in Step 4a), then the Q-step in this algorithm consists of selecting the pair $(\mathbf{Y}, \mathbf{U})$ in Steps 1 and 2. The R-step consists of uniform sampling of the subset of H for which $f(\mathbf{z})/\mu[\mathcal{N}(\mathbf{z})] \geqslant \mathbf{U}$ in Steps 3 to 5. If we do implement the reduction of H, then the Q-step consists of selecting $(\mathbf{Y}, \mathbf{U}, \mathbf{Z}_1, \ldots, \mathbf{Z}_k)$ in Steps 1 to 4 and the R-step consists of accepting $\mathbf{Z}_k$ with probability 1 in Step 5.

It is not difficult to show that if $\mu(\mathcal{N}_{\mathbf{x}})$ is constant for all $\mathbf{x}$, then the denominators in Steps 2 and 4 above can be replaced by 1. With $\mathcal{N}_{\mathbf{x}} = \mathscr{X}$ for all $\mathbf{x}$, the NS reduces to the variant of the slice sampler described in Section 2.1 with $m = 1$. If $\mathscr{X}$ is $\mathbb{R}^n$ with Lebesbue measure and $\mathcal{N}_{\mathbf{x}}$ is a hypercube of side s for all $\mathbf{x} \in \mathscr{X}$, then the NS reduces to a variant of the slice sampler described in [14]. In this case, H is reduced to a smaller hyper-rectangle with $\mathbf{Z}_k$ as a corner in Step 4a). Another interesting case is obtained if we suppose that $\mathscr{X}$ is a discrete space represented by a connected graph in which nodes represent

states and edges represent allowed transitions. Let μ be counting measure and let $\mathcal{N}_{\mathbf{x}}$ consist of $\mathbf{x}$ and all of its neighbors, that is, all nodes adjacent to $\mathbf{x}$. Then $\mathbf{U}$ is chosen between 0 and $f(\mathbf{x})/|\mathcal{N}_{\mathbf{x}}|$ at step 2. The optional reduction of H in step 4a can be achieved by simply excluding $\mathbf{Z}_k$. The fact that previously rejected elements are excluded from being chosen a second time at Step 4a) should in principle make the neighborhood sampler faster than a random walk sampler with a uniform proposal function on the same neighborhoods, since the amount of computation is otherwise comparable.

2.3 Resequencing

Resequencing is the practice of determining the sequence of a biological molecule — usually DNA — by assembling short sub-sequences using related sequences that are already known to aid the assembly. For example, sequencing of some part of the genome of an individual human can be achieved by assembling short sub-sequences using the corresponding part of the already sequenced reference genome to guide the assembly. Similarly, parts of the genomes of other species can be assembled with reference to known genomes of related species. The problem of resequencing is important because modern high-throughput sequencing technologies determine only very short subsequences, or reads. For example, the technology known as *Sequencing By Hybridization* (SBH) determines the subsequence content of an unknown DNA by identifying all probes of a given length (often around 10 nucleotides) that bind to it [1]. More recently, a number of groups have developed fast, high-throughput technologies that use short reads [12, 16]. The use of sequences known to be similar can greatly facilitate the assembly process.

Fig. 3. An idealized picture of resequencing. The sequence in the centre is unknown. A similar sequence, shown at the top, is known, as is the complete set of subsequences of length 10.

We propose the following idealized model of resequencing. Suppose that we have a known sequence S of length L as shown in Figure 3. Suppose further that we model the generation of the unknown sequence $\mathbf{x}$ as the result of independent transitions at each base, with a known transition matrix M. Finally, suppose that we know all contiguous sub-sequences of length k contained in the unknown sequence. Let the set of such sub-sequences be denoted D. The target space $\mathscr{X}$ here thus consists of all sequences of length L with precisely the same set of length k sub-sequences. The posterior distribution on this space, given the sub-sequences, is the restriction to $\mathscr{X}$ of the distribution over all sequences of length L defined by the transition matrix. That is:

$$p(\mathbf{x}|D,S,M) = \prod_{i=1}^{L} M(S_i, \mathbf{x}_i)$$

restricted to $\mathscr{X}$.

To avoid having to estimate the transition matrix M, we can specify a prior probability distribution for each row of the transition matrix. Here we use a Dirichlet distribution:

$$g(M_i) \propto M_{i1}^{\alpha-1} M_{i2}^{\alpha-1} M_{i3}^{\alpha-1} M_{i4}^{\alpha-1}$$

for each row i, with $\alpha = 0.001$. Integrating over M_{ij} for $i,j = 1,\ldots,4$ results in the target distribution:

$$f(\mathbf{x}|D,S) \propto \prod_{i=1}^{4} \frac{\Gamma(C_{i1}+\alpha)\Gamma(C_{i2}+\alpha)\Gamma(C_{i3}+\alpha)\Gamma(C_{i4}+\alpha)}{\Gamma(C_{i1}+C_{i2}+C_{i3}+C_{i4}+4\alpha)}$$

where $C_{ij} = C_{ij}(\mathbf{x})$ is the number of positions at which sequence S has character i and sequence $\mathbf{x}$ has character j.

We sample from the distribution using the NS for a discrete space described above. Let the nodes of the graph be all sequences that have the same set of length k sub-sequences. The edges of the graph connect sequences that are related by transformations of the following form:

1. **Transpositions:** Sequences of the form $y_1z_1y_2z_2y_3z_1y_4z_2y_5$ where z_1 and z_2 are length $k-1$ subsequences and y_1, y_2, y_3, y_4 and y_5 are sequences of any length can be *transposed* by swapping the sequences y_2 and y_4. Sequences of the form $y_1z_1y_2z_1y_4z_1y_5$ can also be transposed by swapping the sequences y_2 and y_4.
2. **Rotations:** Sequences of the form $z_1y_1z_2y_2z_1$ where z_1 and z_2 are length $k-1$ subsequences and y_1 and y_2 are subsequences of any length can be *rotated* by forming the sequence $z_2y_2z_1y_1z_2$.

Some subtlety is required here: the sub-sequences labelled y may be null sequences, and in fact the sequences labelled z may even overlap. It has been shown [15] that sequences related by these transformations have the same

set of length k subsequences and that graphs formed as described above are connected. Note that rotations are only possible if the sequence begins and ends with the same $k-1$ characters. Here we consider only sequences that do not begin and end with the same sub-sequence, so that we need only consider transpositions. Hence the neighborhood $N_{\mathbf{x}}$ of a sequence $\mathbf{x}$ consists of $\mathbf{x}$ plus all sequences that can be obtained from $\mathbf{x}$ by transpositions.

We implemented the NS for discrete spaces for this problem and tested it on a known human DNA sequence of length 4009 nucleotides containing an exon of the breast cancer associated BRCA1 gene (specifically locus 38,496,578-38,500,586 of the March 2006 assembly of human chromosome 17). We also obtained a known sequence of the chimpanzee genome of the same length, aligned to this section of BRCA1 without any insertions or deletions. The human sequence was used as the reference and the chimpanzee sequence was treated as unknown. Figure 4 shows the log-likelihood values for 400 iterations of the algorithm, showing rapid convergence to a single sequence which on inspection turns out to be identical to the known chimpanzee sequence.

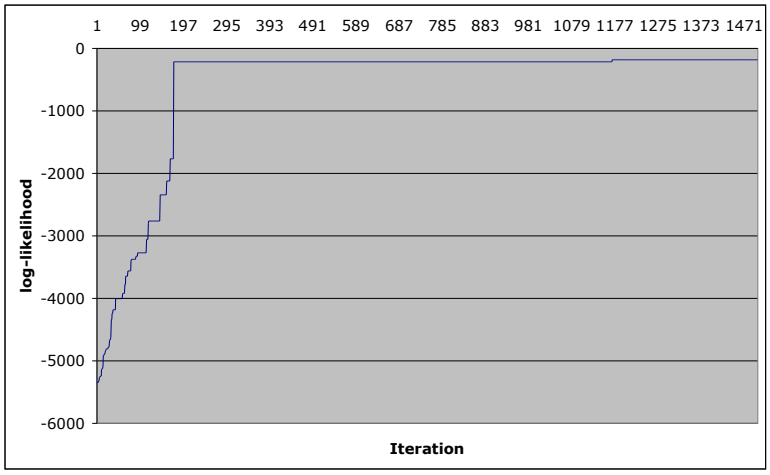

Fig. 4. log-likelihood for resequencing application of the Neighborhood sampler.

Note, however, that the sampler spends many iterations at a nearly optimal sequence, only making the final transposition at iteration 1170. It is not clear whether it is possible for this sampler to become stuck in a local mode for infeasible lengths of time. However, one possible way to improve mixing would

be to expand each neighborhood $N_{\mathbf{x}}$ to include sequences that can be obtained from $\mathbf{x}$ by two successive transpositions.

3 Discussion

The GGS provides a general framework within which all of the commonly used MCMC samplers can be described. This generality raises the interesting possibility of theoretically determining the sampler within this framework with optimal convergence and/or mixing rate for a specific distribution or family of distributions. The GGS also supplies a framework within which new samplers can be generated by exploring various possibilities for the sets $\mathscr{R}(\mathbf{x},\mathbf{y})$ and other parameters. The Neighborhood Sampler is an example of a new sampler generated in this manner.

To conclude this paper, we describe a technique that we recently developed to improve the convergence and mixing rate of the genome segmentation sampler that we described in [11]. The technique is based on the fact that typical trans-dimensional sampling (that is, sampling of spaces in which the dimension varies from point to point) involves conditional sampling of sets $\mathscr{R}(\mathbf{x},\mathbf{y})$ in which the dimension of the x component varies by at most one. In other words, each set $\mathscr{R}(\mathbf{x},\mathbf{y})$ can be subdivided naturally into a set $\mathscr{R}_1(\mathbf{x},\mathbf{y})$ in which all x components have dimension k, say, and a set $\mathscr{R}_2(\mathbf{x},\mathbf{y})$ in which all x components have dimension $k+1$.

We can now define a symmetric function s as follows. Let $s[(\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}')] = 0$ if $(\mathbf{x},\mathbf{y})$ and $(\mathbf{x}',\mathbf{y}')$ are both in $\mathscr{R}_1(\mathbf{x},\mathbf{y})$ or both in $\mathscr{R}_2(\mathbf{x},\mathbf{y})$. Otherwise, let

$$s[(\mathbf{x},\mathbf{y}),(\mathbf{x}',\mathbf{y}')] = \frac{1}{\max\{\sum\limits_{(\mathbf{z},\mathbf{w})\in\mathscr{R}_1(\mathbf{x},\mathbf{y})} f(\mathbf{z})q_{\mathbf{z}}(\mathbf{w}),\ \sum\limits_{(\mathbf{z},\mathbf{w})\in\mathscr{R}_2(\mathbf{x},\mathbf{y})} f(\mathbf{z})q_{\mathbf{z}}(\mathbf{w})\}}$$

if $(\mathbf{x}',\mathbf{y}') \in \mathscr{R}^{-}(\mathbf{x},\mathbf{y})$. Consequently, if $(\mathbf{x},\mathbf{y}) \in \mathscr{R}_1(\mathbf{x},\mathbf{y})$ and

$$\sum_{(\mathbf{z},\mathbf{w})\in\mathscr{R}_1(\mathbf{x},\mathbf{y})} f(\mathbf{z})q_{\mathbf{z}}(\mathbf{w}) \leqslant \sum_{(\mathbf{z},\mathbf{w})\in\mathscr{R}_2(\mathbf{x},\mathbf{y})} f(\mathbf{z})q_{\mathbf{z}}(\mathbf{w})$$

then the probability of transition to $\mathscr{R}_2(\mathbf{x},\mathbf{y})$, obtained by summing 2 over $\mathscr{R}_2(\mathbf{x},\mathbf{y})$, in one. Similarly, if $(\mathbf{x},\mathbf{y}) \in \mathscr{R}_1(\mathbf{x},\mathbf{y})$ and

$$\sum_{(\mathbf{z},\mathbf{w})\in\mathscr{R}_1(\mathbf{x},\mathbf{y})} f(\mathbf{z})q_{\mathbf{z}}(\mathbf{w}) \geqslant \sum_{(\mathbf{z},\mathbf{w})\in\mathscr{R}_2(\mathbf{x},\mathbf{y})} f(\mathbf{z})q_{\mathbf{z}}(\mathbf{w})$$

then the probability of transition to $\mathscr{R}_1(\mathbf{x},\mathbf{y})$ is one. The probability of a change in dimension is therefore high.

References

1. Drmanac, R., Drmanac, S., Strezoska, A., Paunesku, T., Labat, I., Zeremski, M., Snoddy, J., Funkhouser, W.K., Koop, B., Hood, L., Crkvenjakov, R.: DNA sequence determination by hybridization: a strategy for efficient large-scale sequencing. Science, **260**, 1649–1952 (1993)
2. Gelfand, A.F., Smith, A.F.M.: Sampling-based approaches to calculating marginal densities. J. Am. Stat. Assoc., **85**, 398–409 (1990)
3. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE T. Pattern Anal., **6**, 721–741 (1984)
4. Green, P.J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika, **82**, 711–732 (1995)
5. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika, **57**, 97–109 (1970)
6. 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)
7. 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 First Asia-Pacific Bioinformatics Conference: Conferences in Research and Practice in Information Technology, **19**, 23–28 (2003)
8. Keith, J.M., Kroese, D.P., Bryant, D.: A Generalized Markov Sampler. Meth. Comp. Appl. Prob., **6**, 29–53 (2004)
9. Keith, J.M., Adams, P., Bryant, D., Cochran, D.A.E., Lala, G.H., Mitchelson, K.R.: Algorithms for sequence analysis via mutagenesis. Bioinformatics", **20**, 2401–2410 (2004)
10. Keith, J.M., Adams, P., Ragan, M.A., Bryant, D.: Sampling phylogenetic tree space with the generalized Gibbs sampler. Molecular Phylogenetics and Evolution, **34**, 459–468 (2005)
11. Keith, J.M.: Segmenting eukaryotic genomes with the generalized Gibbs sampler. Journal of Computational Biology, **13**, 1369–1383 (2006)
12. Margulies, M., Egholm, M., Altman, W.E., et al.: Genome sequencing in microfabricated high-density picolitre reactors. Nature, **437**, 376–380 (2005)
13. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H.: Equations of state calculations by fast computing machines. J. Chem. Phys., **21**, 1087–1092 (1953)
14. Neal, R.M.: Slice sampling. Ann. Stat., **31**(3), 705–767 (2003)
15. Pevzner, P.A.: DNA Physical Mapping and Alternating Eulerian Cycles in Colored Graphs. Algorithmica, **13**, 77–105 (1995)
16. Shendure, J., Porreca, G.J., Reppas, N.B., Lin, X., McCutcheon, J.P., Rosenbaum, A.M., Wang, M.D., Zhang, K., Mitra, R.D., Church, G.M.: Accurate Multiplex Polony Sequencing of an Evolved Bacterial Genome. Science, **309**, 1728–1732 (2005)