1 October 2013

Searching for Efficient Markov Chain Monte Carlo Proposal Kernels

Ziheng Yang and Carlos E. Rodríguez
Department of Genetics, Evolution and Environment, University College London, Darwin Building, Gower Street, London WC1E 6BT, UK

Running head: Efficient MCMC proposals
Key words: MCMC, proposal, efficiency, mixing

Correspondence to:
Ziheng Yang
Department of Genetics, Evolution and Environment
University College London
Darwin Building
Gower Street
London WC1E 6BT
England
Email: z.yang@ucl.ac.uk
Phone: +44 (20) 7679 4379
Fax: +44 (20) 7679 7096

Abstract

Markov chain Monte Carlo or the Metropolis-Hastings (MH) algorithm is a simulation algorithm that has made modern Bayesian statistical inference possible. Nevertheless, the efficiency of different MH proposal kernels has rarely been studied except for the Gaussian proposal. Here we propose a novel class of Bactrian kernels, which avoid proposing values that are very close to the current value, and compare their efficiency with a number of proposals for simulating different target distributions, with efficiency measured by the asymptotic variance of a parameter estimate. The uniform kernel is found to be more efficient than the Gaussian kernel, while the Bactrian kernel is even better. When optimal scales are used for both, the Bactrian kernel is at least 50% more efficient than the Gaussian. Implementation in a Bayesian program for molecular-clock dating confirms the general applicability of our results to generic MCMC algorithms. Our results refute a previous claim that all proposals had nearly identical performance and will prompt further research into efficient MCMC proposals.

Significance statement

Bayesian statistics is widely used in various branches of sciences. Its main computational method is the Markov chain Monte Carlo (MCMC) algorithm, which is used to simulate a sample on the computer, on which all Bayesian inference is based. The efficiency of the algorithm determines how long one should run the computer program to obtain estimates with acceptable precision. We compare different simulation algorithms and propose a new class of algorithms (called Bactrian algorithms), which are found to be ~50% more efficient than the current algorithms. Our tests suggest that the improvement is generic and may apply to all MCMC algorithms developed in various applications of Bayesian inference.

/body

Markov chain Monte Carlo (MCMC) algorithms can be used to simulate a probability distribution $\pi(x)$ that is known only up to a factor, that is, with only $\frac{\pi(y)}{\pi(x)}$ known. They are especially important in Bayesian inference where π is the posterior distribution. In a Metropolis-Hastings (MH) algorithm (1, 2), a proposal density $q(y|x)$, with $x, y \in \chi$, is used to generate a new state y given the current state x. The proposal is accepted with probability $\alpha(x, y)$. If the proposal is accepted, the new state becomes y; otherwise it stays at x. The algorithm generates a discrete-time Markov chain with state space χ and transition law P having transition probability density

$$p(x, y) = \begin{cases} q(y \mid x) \cdot \alpha(x, y), & y \neq x, \\ 1 - \int_\chi q(y \mid x) \cdot \alpha(x, y)\,\mathrm{d}y, & y = x. \end{cases} \tag{1}$$

The acceptance probability α is chosen so that the detailed balance condition is satisfied: $\pi(x)p(x, y) = \pi(y)p(y, x)$, for all $x, y \in \chi$. The MH choice of α is

$$\alpha(x, y) = \min\left\{1, \frac{\pi(y)}{\pi(x)} \times \frac{q(x \mid y)}{q(y \mid x)}\right\}. \tag{2}$$

The proposal kernel $q(y|x)$ can be very general. As long as it specifies an irreducible aperiodic Markov chain, the algorithm will generate a reversible Markov chain with stationary distribution π. Here we assume that χ is a subset of $\mathbb{R}^k$ and both $\pi(y)$ and $q(y|x)$ are densities on χ.

Given a sample $x_1, x_2, \ldots, x_n$ simulated from P, the expectation of any function $f(x)$ over π

$$I = E_\pi\{f(x)\} = \int_\chi \pi(x) f(x)\,\mathrm{d}x \tag{3}$$

can be approximated by the time average over the sample

$$\tilde{I} = \frac{1}{N} \sum_{i=1}^{N} f(x_i). \tag{4}$$

This is an unbiased estimate of I, and converges to I according to the central limit theorem, independent of the initial state x_0 (e.g., 3, p.99). The asymptotic variance of $\tilde{I}$ is

$$\nu = \lim_{N\to\infty} N\,\mathrm{var}\{\tilde{I}\} = V_f \times \left[1 + 2(\rho_1 + \rho_2 + \rho_3 + \cdots)\right] = \frac{V_f}{E}, \quad (5)$$

where $V_f = \mathrm{var}_\pi\{f(X)\}$ is the variance of $f(X)$ over π, and $\rho_k = \mathrm{corr}\{f(x_i), f(x_{i+k})\}$ is the lag-k autocorrelation (e.g., 4, pp. 87-92), and the variance ratio $E = V_f / \nu$ is the efficiency: an MCMC sample of size N is as informative about I as an independent sample of size NE. Thus NE is known as the *effective sample size*.

Given π and q, the MH choice of (2) maximizes $\alpha(x, y)$, minimizes ν of eq. (5) and is optimal (5). The choice of q given π is less clear. The independent sampler $q(y|x) = \pi(y)$ has $E = 1$, but is often impossible or impractical to sample from. The choice of scale of the Gaussian kernel applied to the Gaussian target, in both univariate and multivariate cases, has been studied (6, 7). For example, the optimal scale σ for the 1-D kernel $y|x \sim \mathrm{N}(x, \sigma^2)$ applied to the N(0, 1) target is 2.5, in which case 44% of proposals are accepted (6), with $P_\mathrm{jump} = 0.44$. Note that

$$P_\mathrm{jump} = \int_\chi \int_\chi \pi(x) q(y \mid x) \alpha(x, y)\,\mathrm{d}x\,\mathrm{d}y\,. \quad (6)$$

However, there has been little research into alternative kernels. In this paper, we examine the mixing efficiency of a number of proposal kernels applied to several representative targets. We focus on 1-D proposals. We develop a Bactrian kernel, which uses a mixture of two Gaussian distributions and which has the overall shape of a Bactrian camel, to reduce the positive autocorrelation (ρ_1 in eq. 5). The proposal density is

$$q(y|x;\, \sigma^2, m) = \frac{1}{2\sigma\sqrt{2\pi(1-m^2)}} \left[\exp\left\{ -\frac{(y - x + m\sigma)^2}{2(1-m^2)\sigma^2} \right\} + \exp\left\{ -\frac{(y - x - m\sigma)^2}{2(1-m^2)\sigma^2} \right\} \right], \quad (7)$$

where $0 \le m < 1$ controls how spiky the two humps are (fig. 1a). This has mean x and variance σ^2. We also consider mixtures of two triangle or two Laplace distributions (fig. 1b & 1c and SI Appendix, fig. S1).

We compare the efficiency of the Bactrian as well as many other proposals for simulating several target distributions. We demonstrate the general nature of our results with a major MCMC application in Bayesian molecular-clock dating of species divergences.

Results

Mixing efficiency for simple targets. Figure 2a shows the performance of three proposal kernels applied to estimate the mean of the target N(0, 1). First, the scale (σ) of each kernel has a great impact on the performance of the kernel. The Gaussian kernel $x'|x \sim \mathrm{N}(x, \sigma^2)$ achieves highest efficiency at $\sigma = 2.5$, in which case $E = 0.23$ (6). In other words, if the target is $\mathrm{N}(\mu, \tau^2)$, the optimal proposal should be over-dispersed with standard deviation 2.5τ. At this optimal scale one has to take an MCMC sample of size $N/0.23$ to achieve the accuracy of an independent sample of size N. Second, the different kernels have different performance. At the optimal scales, the uniform kernel is 21% more efficient than the Gaussian (0.28/0.23 = 1.21), and the Bactrian kernel is 65% more efficient than the Gaussian (0.38/0.23 = 1.65) (table 1). The Bactrian kernel uses a mixture of two single-moded distributions and avoids values very close to the current value (fig. 1 and SI Appendix, text S1). *To propose something new, propose something different.*

As the efficiency E (eq. 5) may depend on the target π or the function f, we have included in our evaluation nine different proposals combined with five targets (figs. S1 and 3). The efficiency E is calculated directly using the transition matrix P on a discretized parameter space (Methods and materials). The five targets are $N(0, 1)$, a mixture of two normals, a mixture of two t_4 distributions, the gamma, and the uniform, all having unit variance (figs. 3). For the gamma and uniform targets, reflection is used to propose new states in the MCMC and to calculate the proposal density (See SI Appendix, text S2 and fig. S2). For all targets except the uniform, every proposal is optimal at an intermediate scale (σ), achieved when the acceptance proportion is $P_\mathrm{jump} \approx 30\%$ for the Bactrian kernels ($m = 0.95$) and ≈40% for the other six kernels (fig. **S3**, table 1).

When each proposal is used at the optimal scale, the relative performance of the kernels does not vary among targets: the uniform kernel is more efficient than the Gaussian, while the Bactrian kernels are the best for all targets examined (tables 1 and S1). For the 2-normals target, the Bactrian kernel (with m

≥ 0.95) is nearly twice as efficient as the Gaussian (table 1). The Bactrian kernel is better if the two modes are far apart (e.g., when m is close to 1). However, if m is too close to 1, efficiency drops off quickly with the increase of σ. We use $m = 0.95$ as it appears to strike the right balance, whereby the optimal $P_{\text{jump}} \approx 0.3$. With this m, the Bactrian kernels have better performance than the uniform and Gaussian for almost the whole range of P_{jump} for the N(0, 1) target (fig. 2b).

The uniform target is revealing for understanding the differences of the kernels (table S1). Due to reflection, all proposals considered here will become the independent sampler with efficiency $E \approx 1$, if $\sigma \to \infty$. The uniform kernel achieves best performance at $\sigma = 3$, in which case $E = 1.52$. In other words, if the target is $U(0, 1)$, a uniform sliding window of width 3 is optimal, and is 52% more efficient than sampling directly from $U(0, 1)$. This 'super-efficiency' is because the samples from the Markov chain tend to be negatively-correlated, like antithetic variables. For this target, the Bactrian kernels (with $m \geq 0.95$) are 4-20 times more efficient than the independent sampler or the Gaussian kernel.

The uniform may not be representative of posteriors in real applications. For the other four targets, the Bactrian kernels are ~50-100% more efficient than the Gaussian kernel (tables 1 and **S1**). Figure 4 shows the transition matrix P for the optimal Gaussian ($\sigma = 2.5$) and Bactrian ($m = 0.95$, $\sigma = 2.3$) kernels applied to the N(0, 1) target. The two kernels are very different. With the Gaussian the next step x' tends to be around the current value x, leaning towards the mode at 0. With the Bactrian the next step x' has a good chance of being in the interval (–1, 1) if x is outside of it, but if x is around 1 (or –1), x' tends to be around –1 (or 1).

Alternative measure of efficiency. A different measure of mixing efficiency (8) is $E_\pi^2 = E(X_i - X_{i-1})^2$, where X_i are the sampled values from the Markov chain. This always ranks the proposals (both different scales for the same kernel and different kernels) the same as does E (tables 1 and S1 and fig. S3). This may be expected since $E(X_i - X_{i-1})^2 = 2\text{var}(X)(1 - \rho_1)$, where ρ_1 is the lag-one autocorrelation of eq. (5). E_π^2 may be useful when searching for efficient kernels in an MCMC algorithm since it is easier to calculate than E although it does not have the nice interpretation of E.

Rate of convergence. We also examined the rate of convergence, measured by δ_8, the difference between the distribution reached after 8 steps and the steady-state distribution (Materials and methods). For the uniform and Gaussian kernels, the optimal σ for fast convergence is slightly greater than for efficient mixing, and one should take larger steps during the burn-in than after (fig. S3). For the Bactrian kernels, optimal σ for fast convergence is about the same as that for mixing. As the burn-in constitutes a small portion of the computational effort, it appears to be adequate to optimize σ for fast mixing only. It is generally understood that a small λ_2 (the second largest eigenvalue of P) is associated with efficient mixing, and a smaller $|\lambda_2|$ (the second largest eigenvalue of P in absolute value) is associated with fast convergence (e.g., 9). Nevertheless, we found that λ_2 and $|\lambda_2|$ are too crude to be used to choose the optimal scale for a given kernel (fig. S3) or to rank different kernels (tables 1 and S1, fig. S3).

Application to Bayesian phylogenetics. To confirm the general applicability of our results, we implement the major proposal kernels in the MCMCTREE program for molecular clock dating (10, 11), and use it to date divergences among six primate species. The tree with fossil calibrations is shown in fig. 5. The large genomic dataset is analyzed under the HKY+Γ_5 model (12, 13). There are 8 parameters: the five times t_1-t_5 (fig. 5), the evolutionary rate μ, the ratio of transition to transversion rates (κ) and the shape parameter (α) of the gamma distribution of rates among sites. As the likelihood depends on the products of times and rate but not on times and rate separately, the model is not fully identifiable. As a result, the posterior for t_1-t_5 and μ involves considerable uncertainties while κ and α are estimated with extremely high precision (10, 14).

All proposals generate the same posterior but their efficiencies differ (fig. 5, table 2). For t_j and μ, the relative efficiencies of the proposals are very similar to those seen earlier for simple targets (tables 1 and S1), with the uniform being better than the Gaussian and the Bactrian better than both. The results confirm the general nature of the results found for simple targets. Efficiency is however very low for κ and α, especially for the Bactrian kernels. This is due to the use of the same scale to update both parameters in the MCMCTREE program. The data are far more informative about κ than about α, so that the same scale is too large for κ and too small for α. As we apply the proposals on the logarithms of κ and α, a relevant scale is the ratio of posterior credibility interval width to the posterior mean (Note that if $y = \log(x)$, then $\sigma_y \approx \sigma_x \left|\frac{\mathrm{d}y}{\mathrm{d}x}\right| = \sigma_x/x$). This is 1.3% for κ and 8.9% for α (table 2). Ideally the scale for the κ move should be 8.9/1.3 = 6.8 times as large as that for α. This is not pursued here, as both parameters are estimated with high precision and the small variations have virtually no effect on the posterior of times and rate, but we note the danger of using one scale for parameters of different (relative) posterior precisions.

Discussion

Automatic scale adjustment. We have evaluated different kernels at nearly optimal scales. In real applications, calculating E at different σs (eq. 5) may be too expensive. Instead we can adjust the scale σ by using the realized P_{jump}, which typically has a monotonic relationship with σ. The $N(0, 1)$ target is of particular interest since in large datasets the posterior will be approximately Gaussian. With this target,

$$P_{\text{jump}} = \tfrac{2}{\pi}\tan^{-1}\left(\tfrac{2}{\sigma}\right) \qquad (8)$$

for the Gaussian kernel (6). Thus given the current σ and P_{jump}, the optimal scale is

$$\sigma^* = \sigma \times \frac{\tan\left(\frac{\pi}{2}P_{\text{jump}}\right)}{\tan\left(\frac{\pi}{2}P_{\text{jump}}^*\right)}, \qquad (9)$$

where the optimal $P_{\text{jump}}^* \approx 0.44$.

Similarly we have

$$P_{\text{jump}} = \tfrac{2}{3\sigma}\sqrt{\tfrac{6}{\pi}}\left(1 - \mathrm{e}^{-\frac{3}{8}\sigma^2}\right) + 2\left(1 - \Phi(\tfrac{\sqrt{3}}{2}\sigma)\right) \qquad (10)$$

for the uniform kernel and

$$P_{\text{jump}} = \frac{2}{\pi}\int_0^a \frac{1}{1+t^2}\exp\left\{-\frac{b^2(1+t^2)}{2(1+at)^2}\right\}\mathrm{d}t\,, \qquad (11)$$

where $a = \frac{2}{\sigma\sqrt{1-m^2}}$ and $b = \frac{m}{\sqrt{1-m^2}}$ for the Bactrian kernel (SI Appendix, text S3 and fig. 6). Eqs. (10) and (11) can be used for automatic scale adjustment for those two kernels through a linear search or look-up table. The P_{jump} vs. σ curves, however, have similar shapes for all the kernels over a wide range of P_{jump} (fig. 6), so that eq. (9) can be used effectively for all kernels. We use $P_{\text{jump}}^* \approx 0.4$ for the uniform, triangle, Laplace, and Gaussian kernels and ≈0.3 for the three Bactrian kernels in the MCMCTREE program, to apply four rounds of automatic scale adjustment during the burn-in. The strategy appears effective and is simpler than adaptive Monte Carlo algorithms (8).

In search for efficient MCMC proposals. We suspect that the Bactrian kernel can be improved further, as it is based on the simple intuition to reduce the autocorrelation. We also note that alternatives to MH algorithms have been suggested, such as the slice sampler (15) and Langevin diffusion, which requires derivatives of the posterior density (16). The efficiency of good MH algorithms relative to the MH alternatives will be interesting topics for future research.

We have here focused on 1-D proposals, and the case of multi-dimensional proposals is yet to be studied. Nevertheless, 1-D proposals may be the most important. Note that one may use a 1-D kernel to change many parameters along a curve in the multidimensional space to accommodate strong correlations in the posterior, as in our clock-dating algorithm. There has been much interest in the optimal choice of scale for Gaussian kernels, especially in infinite dimensions (7, 8). In contrast virtually no study has examined alternative kernels. This appears to be due to the influence of ref. (6), which claimed that different kernels had "nearly identical" performance. This conclusion is incorrect. Note that even the uniform is more efficient than the Gaussian and also much cheaper to simulate. We hope that our results will motivate researches into efficient MCMC proposals.

Methods and materials

Calculation of MCMC mixing efficiency. We used the *initial positive sequence* method (17) to calculate the asymptotic variance ν of eq. (5). More efficiently, for the simple target, we calculated ν from the transition matrix P on a discretized state space directly (6). We use $K = 500$ bins over the range $(x_L, x_U) = (-5, 5)$, with each bin having the width $\Delta = (x_U - x_L)/K$, and with bin i represented by $x_i = x_L + (i - ½)\Delta$. The steady-state distribution on the discrete space is then $\pi_i = \pi(x_i)\Delta$, $i = 1, 2, \ldots, K$, renormalized to sum to 1. The $K \times K$ transition matrix $P = \{p_{ij}\}$ is constructed to mimic the MCMC algorithm

$$p_{ij} = q(x_j|x_i)\alpha(x_i, x_j)\Delta,\ \ 1 \le i, j \le K,\ i \ne j, \qquad (12)$$

with $p_{ii} = 1 - \sum_{j \ne i} p_{ij}$. We then have

$$P_{\text{jump}} = \sum_{i=1}^{K} \pi_i (1 - p_{ii}) . \qquad (13)$$

Let $f = (f_1, f_2, \ldots, f_K)^T$, with $f_i = f(x_i)$. Define $A = \{a_{ij}\}$, with $a_{ij} = \pi_j$, to be the 'limiting matrix', $Z = [I - (P - A)]^{-1}$ to be the 'fundamental matrix' (18, p.74), and $B = \text{diag}\{\pi_1, \pi_2, \ldots, \pi_K\}$. Then eq. (5) becomes

$$v(f, \pi, P) = f^T \cdot (2BZ - B - BA) \cdot f \qquad (14)$$

(18, p.84, 5). This can equivalently be calculated using the spectral decomposition of P, as

$$v(f, \pi, P) = \sum_{k \ge 2}^{K} \frac{1 + \lambda_k}{1 - \lambda_k} \left(E^T B f\right)_k^2 , \qquad (15)$$

where $1 = \lambda_1 > \lambda_2 \ge \lambda_3 \ge \ldots \lambda_K > -1$ are the eigenvalues of P and columns of E are the corresponding right eigenvectors, normalized so that $E^{\text{T}}BE = I$ (19, 20, 9). We calculate the eigenvalues and eigenvectors of P as follows. Define $T = B^{1/2}PB^{-1/2}$ to be a symmetrical matrix (so that its eigenvalues are all real and can be calculated using standard algorithms). Let $T = R\Lambda R^T$, where $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_K\}$, and $R^T = R^{-1}$. Then

$$P = B^{-1/2}TB^{1/2} = (B^{-1/2}R)\Lambda(R^TB^{1/2}) = E\Lambda E^TB, \qquad (16)$$

with $E = B^{-1/2}R$.

Thus we calculated the efficiency both using MCMC simulation on the continuous space (Eq. 5) and using Eqs. 14 and 15 in the discretized space.

The rate of convergence of the chain is often measured by $|\lambda|_2 = \max_{i \ge 2} |\lambda_i|$, the second largest eigenvalue of P in absolute value (9). For a more precise measure, we used

$$\delta_n = \max_i \sum_j \left| p_{ij}^n - \pi_j \right|, \qquad (17)$$

where p_{ij}^n is the ijth element of P^n. We use $n = 8$, with P^8 calculated using the spectral decomposition of P or using three rounds of matrix squaring. The maximum is typically achieved with the initial state in the tail ($i = 0$ or $K - 1$).

All computations are achieved using an ANSI C program written by Z.Y. This is available at http://abacus.gene.ucl.ac.uk/software/MCMCjump.html.

Proposals and targets evaluated. We evaluate nine different MCMC proposal kernels (fig. S1). They are (a) uniform sliding window, (b) triangle proposal, (c) Laplace proposal, (d) Gaussian sliding window, (e) t_4 (t distribution with df = 4) sliding window, (f) Cauchy sliding window, and (g-i) three Bactrian proposals based on the Gaussian, triangle, and Laplace kernels. Note that all proposal densities except the Cauchy have mean x and variance σ^2. The density functions for all kernels are given in SI Appendix, Text S1 (fig. S1).

We use five target densities (fig. 3). They are (i) Gaussian $N(0, 1)$; (ii) mixture of two Gaussian distributions, $\frac{1}{4}N(-1,\frac{1}{4}) + \frac{3}{4}N(1,\frac{1}{4})$, with mean $\frac{1}{2}$ and variance 1; (iii) mixture of two t_4 distributions: $\frac{3}{4}t_4(-\frac{3}{4}, s) + \frac{1}{4}t_4(\frac{3}{4}, s)$, with $s = \sqrt{37}/8$; (iv) gamma distribution $G(4, 2)$ with mean 2 and variance 1; and (v) uniform $U(-\sqrt{3}, \sqrt{3})$. They all have variance 1. With the gamma and uniform targets, the variable x is bounded but most proposal kernels considered in this study have support over $(-\infty, \infty)$. Thus reflection is necessary. For the gamma, if the proposed value $x' < 0$, it is simply reset to $-x'$. For the uniform, we implemented reflection algorithms to calculate the proposal density and to sample from it (SI Appendix, *Text* S2).

Each kernel is implemented using ~30 different scales (σ) to identity the optimum. Thus with 5 targets and 9 kernels, we constructed ~1350 Markov transition matrices and MCMC algorithms.

Application to molecular clock dating. We implemented the major proposal kernels in the MCMCTREE program for dating species divergences (10, 11), and applied it to date divergences among six primate species under the molecular clock (fig. 5). The sequence data consist of the 1st and 2nd codon positions of 14,631 protein-coding genes, with 8,708,584 base pairs in the sequence alignment. The data were analyzed previously by dos Reis and Yang (14, 21). The likelihood is calculated under the HKY+Γ_5 model (12, 13). There are 8 parameters: the five times or node ages t_1-t_5, the rate μ, the ratio (κ) of the transition to transversion rates, and the shape parameter (α) of the gamma distribution of rates among sites. The prior on times t_1-t_5 incorporates calibration information based on the fossil record (see 21). The rate is assigned the prior $\mu \sim$ G(2, 80), with mean 0.025 or 2.5×10^{-8} changes per site per year. Two gamma priors are assigned on the substitution parameters: $\kappa \sim$ G(2, 0.5) with mean 4, and $\alpha \sim$ G(2, 20) with mean 0.1.

Each iteration of the MCMC algorithm consists of four steps. The first three update (i) the times t_j, (ii) the rate μ, and (iii) parameters κ and α. Step (iv) multiplies the times t_j and divides the rate μ by the same random variable c that is close to 1. This is a 1-D move and accommodates the positive correlation among t_1-t_5 and the negative correlation between t_1-t_5 and μ. All four proposals are proportional scalers, equivalent to apply a sliding window (which can be any of the proposal kernels evaluated in this paper) on the logarithm of the parameter. The scale for each proposal is adjusted automatically to achieve optimal P_{jump}. We use a burn-in of 4000 iterations to estimate P_{jump} and apply four rounds of automatic scale adjustments (eq. 9), setting the target P_{jump} to 0.4 for the uniform, triangle, Laplace, and Gaussian kernels and to 0.3 for the three Bactrian kernels. After the burn-in, the chain is run for 40000 iterations, and the collected samples are used to calculate the asymptotic variance of parameter estimates (eq. 5) (17).

ACKNOWLEDGMENTS. Z.Y. was supported by a Biotechnology and Biological Sciences Research Council grant (BB/K000896/1) and a Royal Society-Wolfson Merit Award.

References

1. Metropolis, N, Rosenbluth, AW, Rosenbluth, MN, Teller, AH & Teller, E. (1953) Equations of state calculations by fast computing machines. *J Chem Physi* 21:1087-1092.
2. Hastings, WK. (1970) Monte Carlo sampling methods using Markov chains and their application. *Biometrika* 57:97-109.
3. Chung, KL (1960) *Markov Processes with Stationary Transition Probabilities* (Springer-Verlag, Heidelberg).
4. Diggle, PJ (1990) *Time Series: a Biostatistical Introduction* (Oxford University Press, Oxford, England).
5. Peskun, PH. (1973) Optimum Monte-Carlo sampling using Markov chains. *Biometrika* 60:607-612.
6. Gelman, A, Roberts, GO & Gilks, WR (1996) in *Bayesian Statistics 5*, eds. Bernardo, J. M., Berger, J. O., Dawid, A. P., et al. (Oxford University Press, Oxford), Vol. 5, pp. 599-607.
7. Roberts, GO, Gelman, A & Gilks, WR. (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann Appl Prob* 7:110-120.
8. Rosenthal, JS (2011) in *Handbook of Markov Chain Monte Carlo*, eds. Brooks, S., Gelman, A., Jones, G. L., et al. (Taylor & Francis, New York), pp. 93-112.
9. Green, PJ & Han, XL (1992) in *Stochastic Models, Statistical Methods & Algorithms in Image Analysis*, eds. Barone, P., Frigessi, A. & Piccioni, M. (Springer, New York), pp. 142-164.
10. Yang, Z & Rannala, B. (2006) Bayesian estimation of species divergence times under a molecular clock using multiple fossil calibrations with soft bounds. *Mol Biol Evol* 23:212-226.
11. Yang, Z. (2007) PAML 4: Phylogenetic analysis by maximum likelihood. *Mol Biol Evol* 24:1586-1591.
12. Hasegawa, M, Kishino, H & Yano, T. (1985) Dating the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol* 22:160-174.
13. Yang, Z. (1994) Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J Mol Evol* 39:306-314.
14. dos Reis, M & Yang, Z. (2013) The unbearable uncertainty of Bayesian divergence time estimation. *J Syst Evol* 51:30-43.
15. Damien, P, Wakefield, J & Walker, S. (1999) Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *J R Statist Soc B* 61:331-344.
16. Roberts, GO & Tweedie, RL. (1996) Exponential convergence of Langevin distributions and their discrete approximations. *Bernouli* 2:341-363.
17. Geyer, CJ. (1992) Practical Markov chain Monte Carlo. *Stat Sci* 7:473-511.
18. Kemeny, JG & Snell, JL (1960) *Finite Markov Chains* (Princeton, Van Nostrand).
19. Sokal, AD (1989) *Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms* (Lecture Notes for the Cours de Troisieme Cycle de la Physique en Suisse Romande, Lausanne, Switzerland (June 1989).
20. Frigessi, A, Hwang, CR & Younes, L. (1992) Optimal spectral structure of reversible stochastic matrices, Monte Carlo methods and the simulation of Markov random fields. *Ann Appl Prob* 2:610-628.
21. dos Reis, M, Inoue, J, Hasegawa, M, Asher, R, Donoghue, PC & Yang, Z. (2012) Phylogenomic data sets provide both precision and accuracy in estimating the timescale of placental mammal evolution. *Proc R Soc Lond B Biol Sci* 279:3491-3500.
22. Inoue, J, Donoghue, PCH & Yang, Z. (2010) The impact of the representation of fossil calibrations on Bayesian estimation of species divergence times. *Syst Biol* 59:74-89.

Table 1. Mixing efficiency and convergence rate for different target and proposal distributions

	(a) Gaussian N(0, 1)						(b) 2 Gaussians					
Proposal	σ	P_{jump}	E	E_π^2	δ_8	$\lvert\lambda\rvert_2$	σ	P_{jump}	E	E_π^2	δ_8	$\lvert\lambda\rvert_2$
(1) Uniform	2.2	0.405	0.276	0.879	0.230	0.671	1.9	0.385	0.227	0.771	0.454	0.746
(2) Triangle	2.4	0.451	0.233	0.760	0.312	0.649	1.9	0.417	0.178	0.627	0.561	0.738
(3) Laplace	3.2	0.439	0.185	0.625	0.383	0.702	3.0	0.383	0.136	0.500	0.541	0.785
(4) Gaussian	2.5	0.439	0.228	0.744	0.302	0.652	2.2	0.388	0.171	0.608	0.457	0.750
(5) t_4	3.2	0.422	0.207	0.686	0.330	0.676	3.0	0.367	0.154	0.555	0.501	0.769
(6) t_1 (Cauchy)	2.0	0.381	0.157	0.542	0.505	0.738	1.8	0.329	0.116	0.431	0.634	0.813
(7) Bactrian												
m = 0.80	2.3	0.403	0.269	0.856	0.229	0.672	2.1	0.361	0.209	0.722	0.422	0.768
m = 0.90	2.3	0.348	0.327	1.004	0.213	0.757	2.2	0.304	0.259	0.873	0.470	0.836
m = 0.95	**2.3**	**0.304**	**0.378**	**1.137**	**0.458**	**0.832**	**2.3**	**0.259**	**0.303**	**1.026**	**0.719**	**0.882**
m = 0.98	2.2	0.292	0.408	1.234	0.698	0.875	2.4	0.232	0.332	1.182	0.892	0.903
m = 0.99	2.2	0.282	0.413	1.273	0.887	0.923	2.5	0.217	0.323	1.264	1.222	0.946
(8) Bactrian triangle												
m = 0.80	2.3	0.408	0.261	0.834	0.239	0.662	2.0	0.380	0.202	0.701	0.462	0.747
m = 0.90	2.3	0.353	0.319	0.985	0.193	0.748	2.1	0.323	0.253	0.853	0.395	0.818
m = 0.95	**2.3**	**0.304**	**0.377**	**1.131**	**0.442**	**0.829**	**2.2**	**0.271**	**0.303**	**1.010**	**0.705**	**0.880**
m = 0.98	2.2	0.292	0.408	1.233	0.687	0.874	2.4	0.231	0.330	1.177	0.886	0.903
m = 0.99	2.2	0.282	0.413	1.273	0.883	0.922	2.5	0.217	0.321	1.260	1.221	0.945
(9) Bactrian Laplace												
m = 0.80	2.5	0.355	0.301	0.937	0.182	0.741	2.5	0.297	0.232	0.799	0.431	0.827
m = 0.90	2.4	0.319	0.348	1.061	0.345	0.802	2.4	0.272	0.278	0.940	0.568	0.856
m = 0.95	**2.3**	**0.300**	**0.384**	**1.160**	**0.530**	**0.843**	**2.4**	**0.249**	**0.315**	**1.071**	**0.726**	**0.880**
m = 0.98	2.2	0.292	0.408	1.238	0.730	0.888	2.5	0.222	0.338	1.208	1.035	0.920
m = 0.99	2.2	0.282	0.412	1.274	0.901	0.930	2.5	0.219	0.325	1.278	1.237	0.949

Note.— E and E_π^2 measure the mixing efficiency of the chain, with large values for high efficiency, while δ_8 measures the convergence rate (with small values for fast convergence). $|\lambda_2|$ is the second largest eigenvalue in absolute value of P. In all cases, $|\lambda_2| = \lambda_2$, the second largest eigenvalue of P.

Table 2. The efficiency (*E*) of five proposal kernels implemented in the MCMCTree program

				Bactrian (m = 0.95)		
Parameters	Posterior	Uniform	Gaussian	Gaussian	Triangle	Laplace
t_1 root	57.6 (47.1, 65.3)	0.242	0.186	0.295	0.310	0.272
t_2 human/macaque	30.6 (25.1, 34.7)	0.242	0.186	0.295	0.310	0.272
t_3 human/orang	17.4 (14.2, 19.7)	0.242	0.186	0.295	0.310	0.273
t_4 human/gorilla	8.98 (7.35, 10.2)	0.241	0.186	0.295	0.309	0.272
t_5 human/chimp	7.12 (5.82, 8.08)	0.241	0.186	0.295	0.308	0.272
μ rate ($\times10^{-8}$/site/year)	2.70 (2.36, 3.27)	0.244	0.185	0.302	0.316	0.279
κ	3.64 (3.61, 3.66)	0.090	0.102	0.028	0.028	0.034
α	0.107 (0.103, 0.112)	0.130	0.143	0.185	0.169	0.183

Figure Legends

Figure 1. The Bactrian proposals proposed in this study are mixtures of two component distributions. We tested the (**a**) normal, (**b**) triangle, and (**c**) Laplace component distributions.

Figure 2. The efficiency of the Gaussian, uniform and Bactrian ($m = 0.95$) proposals plotted (**a**) against the proposal step length (standard deviation σ) and (**b**) against P_{jump}. The target is N(0, 1).

Figure 3. Five target distributions used to evaluate the efficiency of different proposal kernels: (**a**) The Gaussian $N(0, 1)$; (**b**) mixture of two Gaussians: $\frac{1}{4}N(-1,\frac{1}{4})+\frac{3}{4}N(1,\frac{1}{4})$; (**c**) mixture of two t_4 distributions: $\frac{3}{4}t_4(-\frac{3}{4},s)+\frac{1}{4}t_4(\frac{3}{4},s)$, with $s = 0.53765$; (**d**) Gamma G(4, 2), and (**e**) Uniform $U(-\sqrt{3},\sqrt{3})$. All have unit variance.

Figure 4. Heat map representation of the transition matrix $P = \{p(x, x')\}$ for (**a**) the Gaussian ($\sigma = 2.5$) and (**b**) the Bactrian ($m = 0.95$, $\sigma = 2.3$) kernels, applied to the $N(0, 1)$ target. $K = 200$ bins are used to discretize the state space (–3, 3). Red (dark) is for high values and white (light) for low values. For any given x, $p(x, x')$ is not smooth at $x' = -x$ (where the contours have sharp corners), and is discontinuous at $x' = x$ due to rejected proposals.

Figure 5. The tree of six primate species showing fossil calibrations and posterior time estimates. The tree is drawn using the posterior means of node ages estimated under the molecular clock, while the bars represent the 95% credibility intervals (CI). Fossil calibrations are implemented using soft bounds, with a 1% soft tail for minima and 5% for maxima (see 10: figure 2). For node 4 (Homininae), a truncated Cauchy distribution is used as the calibration density with $p = 0.1$ and $c = 1$ (22).

Figure 6. The acceptance probability (P_{jump}) as a function of the step length σ for the Gaussian, uniform and Bactrian proposals when the target is $N(0, 1)$.

Supplementary Information for
Searching for Efficient Markov Chain Monte Carlo Proposal Kernels

Ziheng Yang (z.yang@ucl.ac.uk) and Carlos E. Rodríguez

Supplementary Text S1: Proposal and target densities

The Bactrian proposal uses a mixture of two single-moded distributions. Suppose $\psi(m, s^2)$ is a location-scale distribution with mean m and variance s^2. Then the $p : (1 - p)$ mixture of $\psi(m_1, s_1^2)$ and $\psi(m_2, s_2^2)$ has mean $pm_1 + (1 - p)m_2$ and variance $ps_1^2 + (1-p)s_2^2 + p(1-p)(m_1 - m_2)^2$. In particular, the mixture

$$h(y; m) = ½\psi(y; -m, 1 - m^2) + ½\psi(y; m, 1 - m^2) \qquad (18)$$

has mean 0 and variance 1 and may be called the 'standard' Bactrian distribution. Here $0 \le m < 1$ is a parameter, with a larger m meaning that the two modes are farther apart (fig. 1). To simulate a variate y from h, generate $z \sim \psi(x; 0, 1)$ and set

$$y = \begin{cases} m + z\sqrt{1-m^2}, & \text{with probability } \tfrac{1}{2}, \\ -m + z\sqrt{1-m^2}, & \text{with probability } \tfrac{1}{2}. \end{cases} \qquad (19)$$

Then

$$x' = x + y\sigma. \qquad (20)$$

will be the new proposed value given the current value x, where the scale σ is adjusted to achieve efficient mixing. The cost of sampling from the Bactrian kernel is thus nearly the same as the cost of sampling from the component distribution $\psi(0, 1)$.

If ψ is Gaussian, the proposal density is given in equation (7). We can replace the Gaussian with any location-scale distribution. We have considered two: the triangle and the Laplace, with the proposal densities

$$q(x' \mid x; \sigma^2, m) = \frac{\sigma\sqrt{6(1-m^2)} - |x' - x + m\sigma|}{12\sigma^2(1-m^2)} I_{|x'-x+m\sigma|<\sigma\sqrt{6(1-m^2)}} + \frac{\sigma\sqrt{6(1-m^2)} - |x' - x - m\sigma|}{12\sigma^2(1-m^2)} I_{|x'-x-m\sigma|<\sigma\sqrt{6(1-m^2)}}. \qquad (21)$$

for the Bactrian-triangle and

$$q(x'|x; \sigma^2, m) = \frac{1}{2\sigma\sqrt{2(1-m^2)}}\left[\exp\left\{-\sqrt{2}\left|\frac{x'-x+m\sigma}{\sigma\sqrt{1-m^2}}\right|\right\} + \exp\left\{-\sqrt{2}\left|\frac{x'-x-m\sigma}{\sigma\sqrt{1-m^2}}\right|\right\}\right]. \qquad (22)$$

for the Bactrian-Laplace.

We have evaluated nine proposal kernels, as follows.

(1) Uniform sliding window $q(x'|x; \sigma^2) = \frac{1}{2\sqrt{3}\sigma}$, $x - \sqrt{3}\sigma \le x' \le x + \sqrt{3}\sigma$, with window size $2\sqrt{3}\sigma$ and variance σ^2.

(2) Triangle proposal, with $q(x'|x; \sigma^2) = \frac{\sqrt{6}\sigma - |x'-x|}{6\sigma^2}$, for $x - \sqrt{6}\sigma \le x' \le x + \sqrt{6}\sigma$. Inversion can be used to simulate from this kernel. Let $u \sim U(0, 1)$ and set $z = -\sqrt{6} + 2\sqrt{3u}$ if $u < ½$ or $\sqrt{6} - 2\sqrt{3(1-u)}$ otherwise. Then $x' = x + z\sigma$ is the desired variable.

(3) Laplace proposal, with $q(x'|x; \sigma^2) = \frac{1}{\sigma\sqrt{2}}\exp\left\{-\sqrt{2}\left|\frac{x'-x}{\sigma}\right|\right\}$. Again inversion can be used to simulate from this kernel. Let $u \sim U(0, 1)$ and set $z = \log(2u)/\sqrt{2}$ if $u < ½$ or $-\log(2(1-u))/\sqrt{2}$ otherwise. Then set $x' = x + z\sigma$.

(4) Gaussian sliding window, with $q(x'|x; \sigma^2) = \phi(x'; x, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}(x'-x)^2\right\}$.

(5) A t_4 (t distribution with df = 4) sliding window $x'|x \sim t_4(x, \sigma)$ with variance σ^2. The proposal density is

$$q(x'|x;\ \sigma^2) = \frac{3}{4\sqrt{2}\sigma}\left[1+\tfrac{(x'-x)^2}{2\sigma^2}\right]^{-\frac{5}{2}}. \tag{23}$$

(6) A Cauchy sliding window $x'|x \sim C(x, \sigma)$, with

$$q(x'|x;\ \sigma) = \frac{1}{\pi\sigma}\left[1+\left(\tfrac{x'-x}{\sigma}\right)^2\right]^{-1}. \tag{24}$$

This does not have finite mean or variance.

(7) The Bactrian proposal with parameter m and variance σ^2 (eq. 7).

(8) The Bactrian-triangle proposal with parameter m and variance σ^2 (eq. 21).

(9) The Bactrian-Laplace proposal with parameter m and variance σ^2 (eq. 22).

Note that all proposal densities $q(x'|x)$ except the Cauchy have mean x and variance σ^2.

We have evaluated five targets, as follows.

(i) Gaussian $N(0, 1)$.

(ii) A mixture of two Gaussian distributions, $\frac{1}{4}N(-1,\frac{1}{4})+\frac{3}{4}N(1,\frac{1}{4})$, with mean $\frac{1}{2}$ and variance 1.

(iii) A mixture of two t_4 distributions: $\frac{3}{4}t_4(-\frac{3}{4},s)+\frac{1}{4}t_4(\frac{3}{4},s)$, with $s = \sqrt{37}/8$. This has mean $-\frac{3}{8}$ and variance 1. This is heavy-tailed, so $K = 1000$ bins are used over the range (–10, 10).

(iv) The gamma distribution $G(4, 2)$ with mean 2 and variance 1.

(v) The uniform distribution $U(-\sqrt{3},\sqrt{3})$ with mean 0 and variance 1.

Supplementary Text S2. Reflection for bounded parameters

When the parameters have bounds, reflection may be necessary for proposing new values in the MCMC algorithm and for calculating the proposal density. We consider reflection in the MCMC algorithm first. Let $x \in (a, b)$. We assume that the proposal is symmetrical, with $q(x'|x) = q(x|x')$. Then if the proposed value is outside the interval, the excess is reflected back into the interval. In other words, if $x' < a$, x' is reset to $a + (a - x') = 2a - x'$; and if $x' > b$, x' is reset to $b - (x' - b) = 2b - x'$. This process may have to be repeated until x' is finally inside the interval. The Hastings ratio is $\frac{q(x|x')}{q(x'|x)} = 1$, because if x can reach x' through a number of reflections, it must be possible for x' to reach x through the same number of reflections, so that $q(x'|x) = q(x|x')$ always hold. If the proposal window is much larger than the range $b - a$, it may be simpler to jump over the interval 'virtually' and calculate the final value x' directly. Let y be the initial proposed value. If $y < a$, the excess $e = a - y$ is on the left; and if $y > b$, the excess $e = y - b$ is on the right. Then $n = \lfloor e/(b - a)\rfloor$, the integer part of $e/(b - a)$, is the number of jumps over the interval, and the final excess is $e' = e - n(b - a)$. If n is odd one changes sides. If the final excess e' is on the left, set $x' = a + e'$; and if it is on the right, set $x' = b - e'$.

We also need consider reflection in the calculation of the proposal density $q(y|x)$ on the discretized space. We developed a parcel-delivery algorithm for this purpose (fig. S2). The algorithm assumes that the initial proposal kernel (before reflection), $q^0(x'|x)$, is symmetrical with $q^0(x'|x) = q^0(x|x') = q^0(|x' - x|)$. As an example, consider the parameter to be a probability so that the target range is $(a, b) = (0, 1)$, and the initial proposal kernel to be $q^0(x'|x) = \phi(x'; x, \sigma^2)$, over the range $(x_L, x_U) = (-5, 5)$, say. Our algorithm breaks the target range (a, b) into K bins, each of length $\Delta x = (b - a)/K$. The range of the initial kernel q^0 is also broken into bins of size Δx. For any given x_i, we start from the centre bin of the sliding window (which corresponds to bin i in the target), and move left to deliver the proposal density mass (the 'parcels') onto the target bins. It we hit the boundary a or b, we turn back. The process is repeated until all the bins (parcels) on the left half of the proposal window are delivered. Then we start from the centre of the proposal window and move right to deliver the right half of the proposal window onto the target bins. The pseudo-code of the algorithm is shown below.

Algorithm S1. Parcel-delivery algorithm of reflection to calculate the proposal density $q(y|x)$. The algorithm assumes that the initial proposal is symmetrical with $q^0(y|x) = q^0(y|x) = q^0(|x - y|)$. Simple modifications are needed if this is not the case.

Input and definitions:

(a, b): range of the target, e.g., (0, 1) if the parameter is a probability, $\theta \in (0, 1)$.
(x_L, x_U): range of the proposal, e.g., (–5, 5) for the Gaussian kernel.
K: the number of bins for discretizing the target (say, $K = 500$).
$\Delta x = (b - a)/K$ is bin size.
nwbins: number of bins in the proposal window.

Output: The proposal density in matrix Q

```
dx  =  (b - a)/K;
for i = 0 to (K - 1) do
   $x_i$ = a + (i + 0.5) dx;          # The state space of the Markov chain
end for
w =  $x_U$ - $x_L$;                   # Size of the proposal window
nwbins  =  int[w/dx + ½];
for i = 0 to nwbins/2 + 2 do         # $q_i$ has $q^0(|y - x|)$ where $x$ and $y$ are $i$ bins apart, with $|y - x| = i\Delta x$.
   $q_i$ = q($x_0$ + i*dx|$x_0$, $\sigma^2$) dx;
end for
for i = 0 to K - 1 do
   for leftright = 0 to 1 do         # leftright = 0 (1) for the left (right) half of the proposal window
      if leftright = 0 then          # direction = –1 (1) to move left (right)
         nsteps = nwbins/2;
         direction = -1;
      else
         nsteps = nwbins - nwbins/2;
         direction = 1;
      end if
      loc = i;                       # Start from centre of proposal window
      for j = 0 to nsteps - 1 do
         $Q_{i,\,loc}$ = $Q_{i,\,loc}$ + $q_j$;
         loc = loc + direction;
         if (loc < 0 or loc ≥ K) then   # We hit the boundary.  Turn back.
            direction = -direction;
            if loc == -1 then
               loc =  0;
            else
               loc =  K - 1;
            end if
         end if
      end for
      $Q_{i,\,loc}$  = $Q_{i,\,loc}$ + $q_j$/2;   # Add half of a bin at the end of the proposal window
   end for
   $Q_{i,\,i}$ = $Q_{i,\,i}$ - $q_0$;           # Remove the extra bin added at the centre of the proposal window
end for
```

Supplementary Text S3. Acceptance probability (P_{jump}) for the uniform and Bactrian kernels applied to the N(0, 1) target

Note that both uniform and Bactrian kernels are symmetrical with $q(y|x) = q(x|y) = q(|y - x|)$. Thus

$$\begin{aligned} P_{\text{jump}} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(x)q(y\,|\,x)\,\mathrm{I}_{\phi(y)>\phi(x)}\,\mathrm{d}x\,\mathrm{d}y + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \phi(x)q(y\,|\,x)\,\mathrm{I}_{\phi(y)\le\phi(x)}\,\frac{\phi(y)}{\phi(x)}\,\mathrm{d}x\,\mathrm{d}y \\ &= 2\iint_{\phi(y)>\phi(x)} \phi(x)q(y\,|\,x)\,\mathrm{d}y\,\mathrm{d}x \qquad (25) \\ &= 4\int_{0}^{\infty}\int_{-x}^{x} \phi(x)q(y\,|\,x)\,\mathrm{d}y\,\mathrm{d}x. \end{aligned}$$

The mass of $\phi(x)q(y|x)$ left of the y axis is the same as right of the y axis. Also right of the y axis, $\phi(y) > \phi(x)$ iff $-x < y < x$.

For the uniform kernel, we have

$$\begin{aligned} P_{\text{jump}} &= \frac{2}{\sqrt{3}\sigma}\left[\int_0^{\frac{\sqrt{3}\sigma}{2}}\int_{-x}^{x}\phi(x)\,\mathrm{d}y\,\mathrm{d}x + \int_{\frac{\sqrt{3}\sigma}{2}}^{\infty}\int_{x-\sqrt{3}\sigma}^{x}\phi(x)\,\mathrm{d}y\,\mathrm{d}x\right] \\ &= \frac{2}{\sqrt{3}\sigma}\left[2\int_0^{\frac{\sqrt{3}\sigma}{2}} x\phi(x)\,\mathrm{d}x + \sqrt{3}\sigma\int_{\frac{\sqrt{3}\sigma}{2}}^{\infty}\phi(x)\,\mathrm{d}x\right] \\ &= \frac{2}{\sqrt{3}\sigma}\left[\sqrt{\frac{2}{\pi}}\left(1-\mathrm{e}^{-\frac{3\sigma^2}{8}}\right) + \sqrt{3}\sigma\left(1-\Phi(\tfrac{\sqrt{3}\sigma}{2})\right)\right] \\ &= \sqrt{\frac{8}{3\pi\sigma^2}}\left(1-\mathrm{e}^{-\frac{3\sigma^2}{8}}\right) + 2\left(1-\Phi(\tfrac{\sqrt{3}\sigma}{2})\right). \end{aligned} \tag{26}$$

Before we consider the Bactrian kernel, we give the following integral, which we need later.

$$I(a, b) = \int_0^{\infty}\phi(x)\Phi(ax+b)\,\mathrm{d}x = \int_0^{\infty}\int_{-\infty}^{ax+b}\tfrac{1}{2\pi}\exp\{-\tfrac{x^2+y^2}{2}\}\,\mathrm{d}y\,\mathrm{d}x,\ \ a > 0, -\infty < b < \infty. \tag{27}$$

We change to the polar system. Then if $b \ge 0$,

$$\begin{aligned} I_+(a,b) &= \frac{1}{2\pi}\int_{-\frac{\pi}{2}}^{\tan^{-1}(a)}\int_0^{\infty}\mathrm{e}^{-\frac{r^2}{2}}\,r\,\mathrm{d}r\mathrm{d}\theta + \frac{1}{2\pi}\int_{\tan^{-1}(a)}^{\frac{\pi}{2}}\int_0^{\frac{b}{\sin\theta-a\cos\theta}}\mathrm{e}^{-\frac{r^2}{2}}\,r\,\mathrm{d}r\mathrm{d}\theta \\ &= \frac{1}{2} - \frac{1}{2\pi}\int_{\tan^{-1}(a)}^{\frac{\pi}{2}}\exp\left\{-\tfrac{b^2}{2(\sin\theta-a\cos\theta)^2}\right\}\mathrm{d}\theta, \end{aligned} \tag{28}$$

and if $b \le 0$,

$$I_{-}(a, b) = \frac{1}{2\pi}\int_{-\frac{\pi}{2}}^{\tan^{-1}(a)}\int_{\frac{b}{\sin\theta-a\cos\theta}}^{\infty}\mathrm{e}^{-\frac{r^2}{2}}\,r\,\mathrm{d}r\mathrm{d}\theta = \frac{1}{2\pi}\int_{-\frac{\pi}{2}}^{\tan^{-1}(a)}\exp\left\{-\tfrac{b^2}{2(\sin\theta-a\cos\theta)^2}\right\}\mathrm{d}\theta. \tag{29}$$

If $b = 0$, both $I_+(a, b)$ and $I_-(a, b)$ reduce to

$$\int_0^{\infty}\phi(x)\Phi(ax)\,\mathrm{d}x = \frac{1}{4} + \frac{1}{2\pi}\tan^{-1}(a)\text{, with } a > 0. \tag{30}$$

Note that the proposal density for the Bactrian kernel (eq. 7) can be written as

$$q(y|x) = \tfrac{1}{2}\,\phi(y;\, x + m\sigma^2,\, (1 - m^2)\sigma^2) + \tfrac{1}{2}\,\phi(y;\, x - m\sigma^2,\, (1 - m^2)\sigma^2). \tag{31}$$

Thus from eq. (25),

$$\begin{aligned} P_{\text{jump}} &= 2\int_0^{\infty}\int_{-x}^{x}\phi(x)\times\left[\phi(y;x+m\sigma,(1-m^2)\sigma^2)+\phi(y;x-m\sigma,(1-m^2)\sigma^2)\right]\mathrm{d}y\,\mathrm{d}x \\ &= 1-2\int_0^{\infty}\phi(x)\times\left[\Phi\left(\tfrac{-m}{\sqrt{1-m^2}}\right)-\Phi\left(\tfrac{-2x-m\sigma}{\sqrt{1-m^2}\sigma}\right)+\Phi\left(\tfrac{m}{\sqrt{1-m^2}}\right)-\Phi\left(\tfrac{-2x+m\sigma}{\sqrt{1-m^2}\sigma}\right)\right]\mathrm{d}x \\ &= -1+2\int_0^{\infty}\phi(x)\left[\Phi\left(\tfrac{2x+m\sigma}{\sqrt{1-m^2}\sigma}\right)+\Phi\left(\tfrac{2x-m\sigma}{\sqrt{1-m^2}\sigma}\right)\right]\mathrm{d}x. \end{aligned} \tag{32}$$

Now define $a = \frac{2}{\sigma\sqrt{1-m^2}} > 0$, $b = \frac{m}{\sqrt{1-m^2}} > 0$. Eq. (32) then becomes

$$\begin{aligned} P_{\text{jump}} &= -1+2[I_+(a,b)+I_-(a,b)] \\ &= \frac{1}{\pi}\left[\int_{-\frac{\pi}{2}}^{\tan^{-1}(a)}\exp\left\{-\frac{b^2}{2(\sin\theta-a\cos\theta)^2}\right\}\mathrm{d}\theta - \int_{\tan^{-1}(a)}^{\frac{\pi}{2}}\exp\left\{-\frac{b^2}{2(\sin\theta-a\cos\theta)^2}\right\}\mathrm{d}\theta\right]. \end{aligned} \tag{33}$$

The integrand is the same in the two integrals and is 0 at $\theta = c = \tan^{-1}(a)$, and symmetrical around c; in other words it is the same at $c - d$ and at $c + d$, for $0 < d < \pi/2 - c$. Thus

$$P_{\text{jump}} = \frac{1}{\pi}\int_{-\frac{\pi}{2}}^{-\frac{\pi}{2}+2\tan^{-1}(a)}\exp\left\{-\frac{b^2}{2(\sin\theta-a\cos\theta)^2}\right\}\mathrm{d}\theta. \tag{34}$$

Now the integral is over an interval of length $2c$, and the integrand is symmetrical around the center: it is the same at $(-\pi/2 + c) - d'$ and at $(-\pi/2 + c) + d'$ for $0 < d' < c$. Thus

$$\begin{aligned} P_{\text{jump}} &= \frac{2}{\pi}\int_{-\frac{\pi}{2}}^{-\frac{\pi}{2}+\tan^{-1}(a)}\exp\left\{-\frac{b^2}{2(\sin\theta-a\cos\theta)^2}\right\}\mathrm{d}\theta \\ &= \frac{2}{\pi}\int_0^{\tan^{-1}(a)}\exp\left\{-\frac{b^2}{2(\cos\theta+a\sin\theta)^2}\right\}\mathrm{d}\theta \end{aligned} \tag{35}$$

Let $t = \tan\theta$, with $\mathrm{d}\theta = 1/(1 + t^2)\,\mathrm{d}t$. Then

$$P_{\text{jump}} = \frac{2}{\pi}\int_0^a \frac{1}{1+t^2}\exp\left\{-\frac{b^2(1+t^2)}{2(1+at)^2}\right\}\mathrm{d}t\,. \tag{36}$$

This reduces to eq. (**8**) if $b = 0$ (if $m = 0$).

Table S1. Mixing efficiency and convergence rate for different target and proposal distributions

	(c) 2 t_4						(d) Gamma $G(4, 2)$						(e) Uniform $U(-\sqrt{3}, \sqrt{3})$					
Proposal kernel	σ	P_{jump}	E	E_π^2	δ_8	$\|\lambda\|_2$	σ	P_{jump}	E	E_π^2	δ_8	$\|\lambda\|_2$	σ	P_{jump}	E	E_π^2	δ_8	$\|\lambda\|_2$
(1) Uniform	2.2	0.366	0.218	0.760	1.276	0.794	3.2	0.464	0.300	0.996	0.204	0.646	3.0	1	1.523	2.417	0.000	0.212
(2) Triangle	2.4	0.386	0.193	0.666	1.229	0.797	3.2	0.479	0.257	0.873	0.262	0.666	∞	1	1	2.0	0.000	0.001
(3) Laplace	3.5	0.381	0.164	0.565	1.047	0.790	4.5	0.459	0.208	0.727	0.327	0.703	∞	1	1	2.0	0.000	0.024
(4) Gaussian	2.6	0.377	0.192	0.659	1.157	0.791	3.5	0.464	0.251	0.857	0.257	0.666	∞	1	1	2.0	0.000	0.000
(5) t_4	3.2	0.362	0.180	0.615	1.039	0.783	4.5	0.450	0.230	0.793	0.287	0.684	∞	1	1	2.0	0.000	0.000
(6) t_1 (Cauchy)	2.0	0.338	0.142	0.496	1.081	0.810	2.6	0.400	0.177	0.633	0.445	0.740	∞	1	1	2.0	0.000	0.000
(7) Bactrian																		
$m = 0.80$	2.4	0.351	0.219	0.749	1.172	0.784	3.2	0.463	0.286	0.972	0.216	0.653	3	1	1.347	2.296	0.000	0.150
$m = 0.90$	2.4	0.296	0.257	0.873	1.065	0.837	3.5	0.418	0.336	1.111	0.139	0.648	3.2	1	2.226	2.764	0.001	0.388
$m = 0.95$	**2.3**	**0.268**	**0.290**	**0.993**	**1.052**	**0.880**	**3.5**	**0.408**	**0.375**	**1.244**	**0.120**	**0.674**	**3.2**	**1**	**4.011**	**3.212**	**0.026**	**0.615**
$m = 0.98$	2.2	0.259	0.308	1.090	1.074	0.904	4.0	0.383	0.394	1.303	0.275	0.780	3.5	1	9.157	3.618	0.257	0.809
$m = 0.99$	2.1	0.271	0.307	1.120	1.173	0.927	4.0	0.392	0.378	1.354	0.569	0.854	3.5	1	17.68	3.793	0.308	0.905
(8) Bactrian triangle																		
$m = 0.80$	2.4	0.357	0.212	0.727	1.176	0.785	3.2	0.467	0.279	0.949	0.221	0.657	3.0	1	1.262	2.232	0.000	0.118
$m = 0.90$	2.3	0.316	0.251	0.858	1.146	0.817	3.5	0.422	0.329	1.089	0.142	0.649	3.2	1	2.088	2.708	0.000	0.359
$m = 0.95$	**2.3**	**0.267**	**0.289**	**0.986**	**1.054**	**0.881**	**3.5**	**0.404**	**0.378**	**1.242**	**0.117**	**0.665**	**3.2**	**1**	**3.875**	**3.190**	**0.022**	**0.604**
$m = 0.98$	2.2	0.259	0.307	1.088	1.073	0.903	4.0	0.381	0.396	1.299	0.249	0.768	3.5	1	8.897	3.610	0.249	0.815
$m = 0.99$	2.1	0.271	0.307	1.120	1.171	0.926	4.0	0.392	0.383	1.351	0.544	0.848	3.5	1	17.334	3.790	0.568	0.904
(9) Bactrian Laplace																		
$m = 0.80$	2.6	0.306	0.245	0.826	1.032	0.820	3.5	0.432	0.305	1.041	0.167	0.653	3.5	1	1.805	2.577	0.000	0.293
$m = 0.90$	2.4	0.284	0.274	0.932	1.048	0.854	4.0	0.391	0.342	1.136	0.142	0.706	3.5	1	2.858	2.968	0.004	0.490
$m = 0.95$	**2.3**	**0.266**	**0.295**	**1.020**	**1.045**	**0.884**	**4.0**	**0.385**	**0.371**	**1.230**	**0.192**	**0.748**	**3.5**	**1**	**4.790**	**3.316**	**0.049**	**0.665**
$m = 0.98$	2.2	0.260	0.307	1.098	1.076	0.909	4.0	0.388	0.381	1.320	0.394	0.821	3.5	1	10.293	3.652	0.296	0.833
$m = 0.99$	2.1	0.272	0.306	1.124	1.181	0.933	4.0	0.393	0.350	1.363	0.671	0.877	3.5	1	19.136	3.805	0.597	0.909

Note.— See legend to table 1. For the two-t_4 and gamma targets, $\lambda_2 = |\lambda_2|$, but for the uniform target, the two are sometimes different.

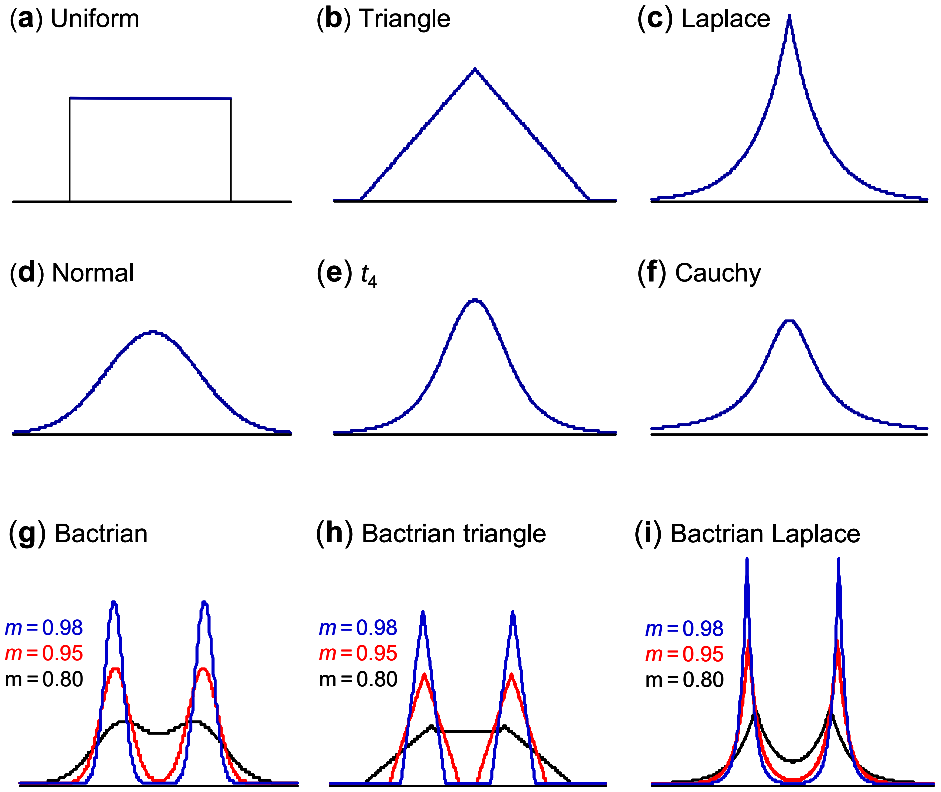


Figure S1. Nine proposal kernels evaluated in this study. The Bactrian proposals (g-i) are shown in figure 1 as well.

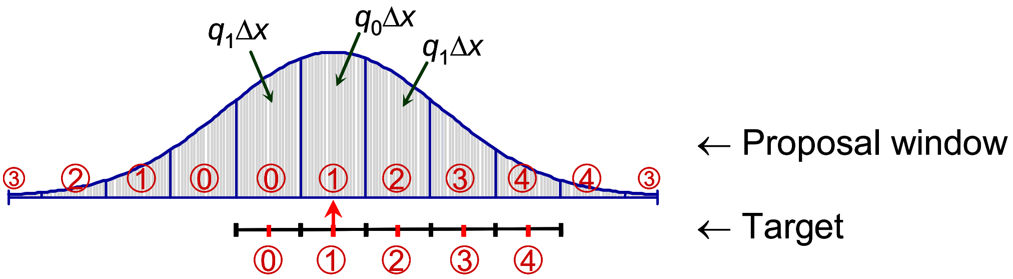


Figure S2. Parcel-delivery algorithm of reflection to calculate the proposal density $q(x'|x)$. The target (0, 1) is broken into $K = 5$ bins, numbered 0, 1, …, 4, with each bin represented by the mid-value $x_i = i\Delta x$. The proposal sliding window has width 2, broken into 10 bins. It is centred above the current value, assumed to be 0.15, at the centre of the second bin (bin 1) in the target. Suppose the Gaussian density is used to propose new values, so that the proposal density mass in each bin in the window is approximated by $q_i\Delta x = \phi(i\Delta x; 0, \sigma^2)\Delta x$. This is the weight of each parcel. The algorithm starts from the center of the proposal window, and moves left to deliver the left half of the window into the target bins. It then starts from the center of the proposal window, and moves right to deliver the right half of the window into the target bins.

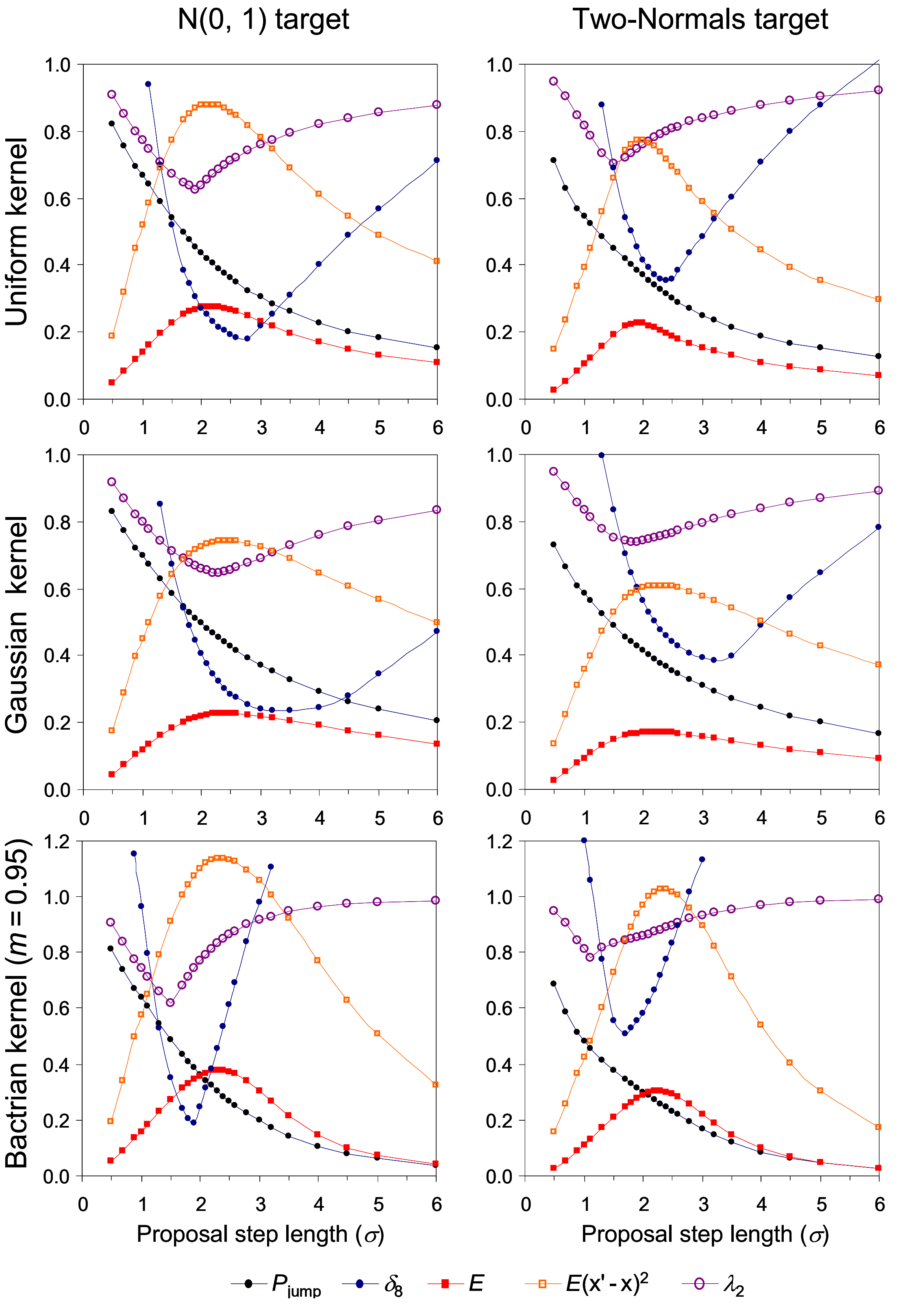


Figure S3. The impact of scale (step length σ) in the uniform, Gaussian, and Bactrian (m = 0.95) kernels applied to the N(0, 1) and two-normal mixture targets. The acceptance proportion (P_{jump}) always decreases with the increase of σ, but the measures of convergence (δ_8) and mixing efficiency (E and E_π^2) as well as the second-largest eigenvalue (λ_2) achieve their optima at intermediate values of σ. For those kernel-target combinations, λ_2 is also $|\lambda|_2$, the second largest eigenvalue by absolute value.

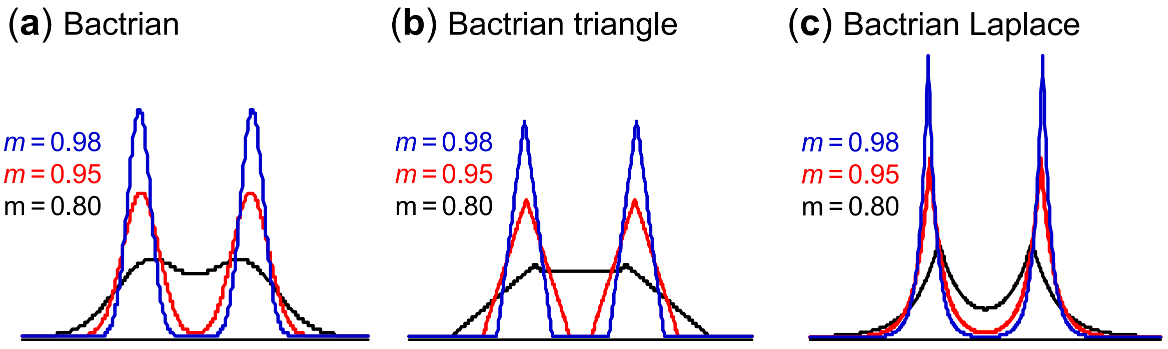

(a) Bactrian
m = 0.98
m = 0.95
m = 0.80
(b) Bactrian triangle
m = 0.98
m = 0.95
m = 0.80
(c) Bactrian Laplace
m = 0.98
m = 0.95
m = 0.80

Figure 1

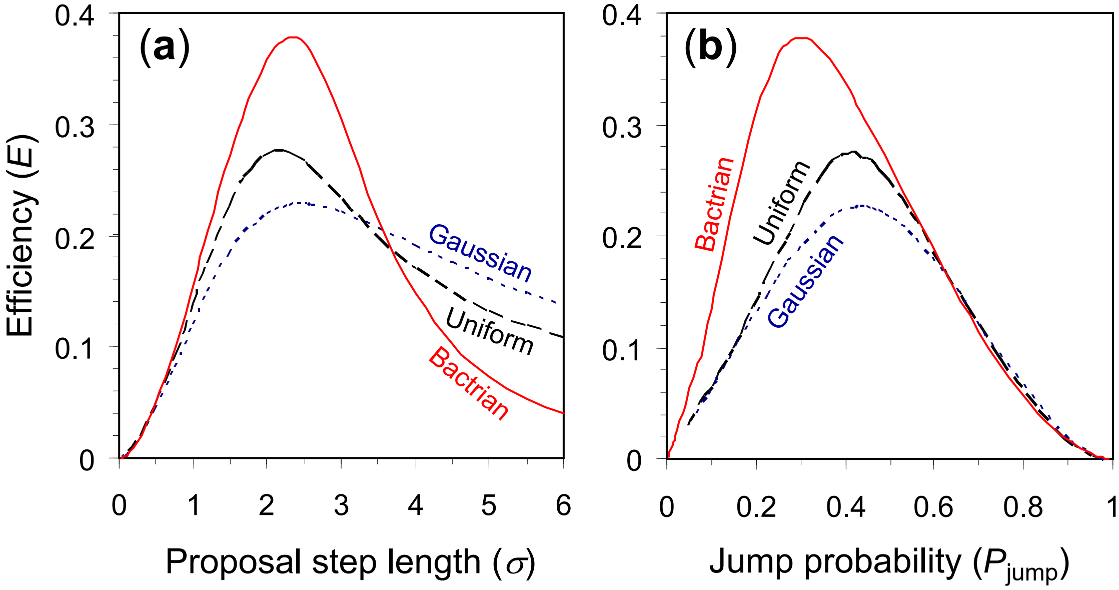

0.4
0.3
0.2
0.1
0
Efficiency (E)
(a)
Gaussian
Uniform
Bactrian
0
1
2
3
4
5
6
Proposal step length (σ)
0.4
0.3
0.2
0.1
0
(b)
Bactrian
Uniform
Gaussian
0
0.2
0.4
0.6
0.8
1
Jump probability (P_{jump})

Figure 2

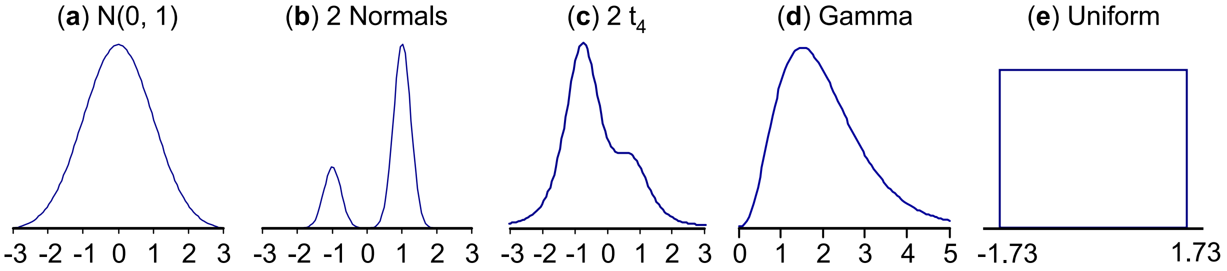
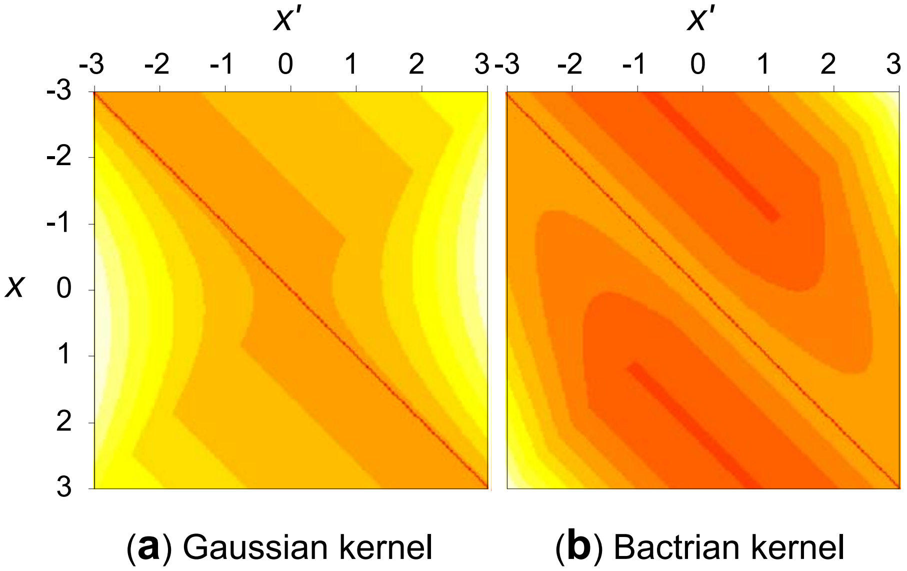
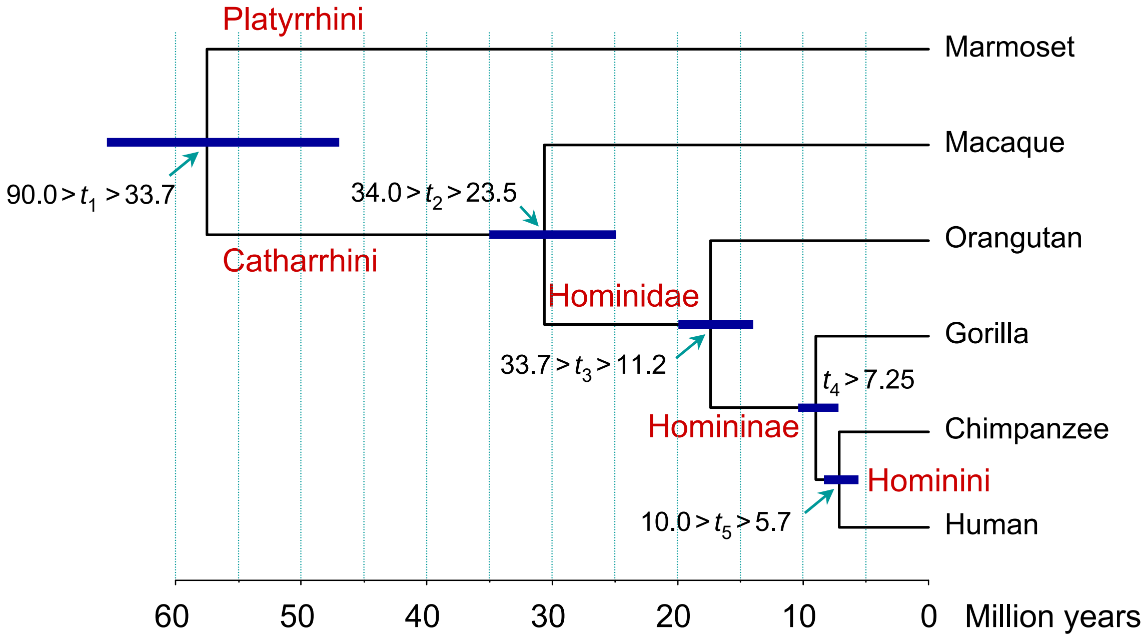

(**a**) N(0, 1)
-3 -2 -1 0 1 2 3
(**b**) 2 Normals
-3 -2 -1 0 1 2 3
(**c**) 2 t_4
-3 -2 -1 0 1 2 3
(**d**) Gamma
0 1 2 3 4 5
(**e**) Uniform
-1.73
1.73

Figure 3



(**a**) Gaussian kernel (**b**) Bactrian kernel

Figure 4



Platyrrhini
Catharrhini
Hominidae
Homininae
Hominini
Marmoset
Macaque
Orangutan
Gorilla
Chimpanzee
Human
90.0 > t_1 > 33.7
34.0 > t_2 > 23.5
33.7 > t_3 > 11.2
t_4 > 7.25
10.0 > t_5 > 5.7
60
50
40
30
20
10
0
Million years

Figure 5

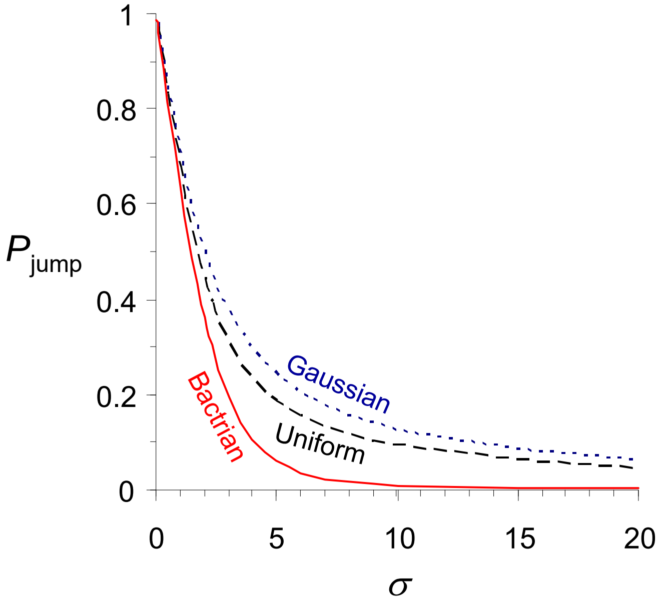

1
0.8
0.6
0.4
0.2
0
P_{jump}
Gaussian
Uniform
Bactrian
0
5
10
15
20
σ

Figure 6