Supplementary Material

I Efficiency curves for Box, Airplane and StrawHat kernels

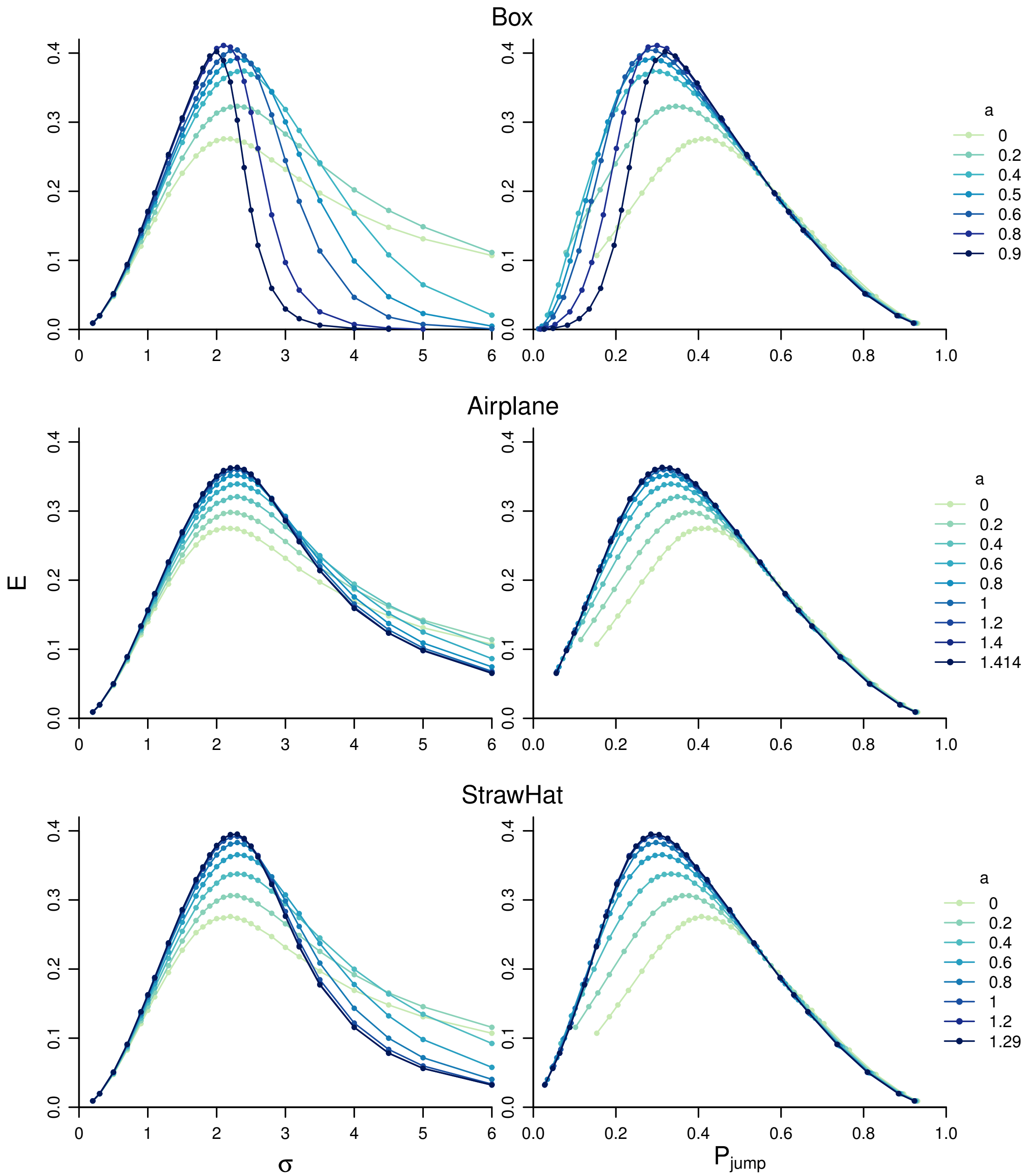

Figure S1: Effect of parameter a of the Box, Airplane and StrawHat kernels on efficiency for estimating the mean of $N(0, 1)$.

II Two-dimensional Gaussian target distributions

We consider two bivariate Gaussian targets $N_2(0, I)$ and $N_2(0, \Sigma)$ with $\Sigma = \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$ with $\rho = 0.9$. For $N_2(0, I)$, we compare proposals that are either a two-dimensional distribution, or a cycle of two one-dimensional distributions. For $N_2(0, \Sigma)$, we also use a variable transformation to deal with the correlation.

The proposal kernels considered are as follows. Let $x = (x_1, x_2)$ be the current value and $x' = (x'_1, x'_2)$ be the proposed value from $q(x'_1, x'_2|x_1, x_2)$.

- Two-dimensional proposals on $\mathbf{R}^2$:
  - K1. Gaussian:
    $$q(x'_1, x'_2|x_1, x_2) = N(x'_1, x'_2|(x_1, x_2), \sigma^2 I_2).$$
    This is a symmetric kernel, with $q(x'_1, x'_2|x_1, x_2) = q(x_1, x_2|x'_1, x'_2)$. Thus the proposal ratio is 1.
  - K2. Square:
    $$q(x'_1, x'_2|x_1, x_2) = \frac{1}{12\sigma^2} 1_S(x_1, x_2)$$
    where $S := \{(u, v) \in \mathbf{R}^2 : x_1 - \sqrt{3}\sigma \le u \le x_1 + \sqrt{3}\sigma, x_2 - \sqrt{3}\sigma \le v \le x_2 + \sqrt{3}\sigma\}$ is the square of side length $2\sqrt{3}\sigma$, centred at (x_1, x_2), and $1_S(x_1, x_2) := 1$ if $(x_1, x_2) \in S$, and 0 otherwise; this has mean (x_1, x_2) and covariance matrix I_2. To generate (x'_1, x'_2) from this kernel, draw $x'_i \sim U(x_i - \sqrt{3}\sigma, x_i + \sqrt{3}\sigma)$ independently for $i = 1, 2$. The proposal ratio is 1.
  - K3. Disc:
    $$q(x'_1, x'_2|x_1, x_2) = \frac{1}{4\pi\sigma^2} 1_S(x_1, x_2)$$
    where $S := \{(u, v) \in \mathbf{R}^2 : (x_1 - u)^2 + (x_2 - v)^2 \le 4\sigma^2\}$ is the disc of radius 2σ, centred at (x_1, x_2); this has mean (x_1, x_2) and covariance I_2. To generate (x'_1, x'_2) from this kernel, first draw $r \sim U(0, 1)$ and $\theta \sim U(0, 2\pi)$, then set $x'_1 := x_1 + 2\sigma\sqrt{r}\cos\theta$ and $x'_2 := x_2 + 2\sigma\sqrt{r}\sin\theta$. The proposal ratio is 1.
- Two one-dimensional proposals (one for each coordinate):
  - K4. Two 1D uniform proposals:
    1. First, draw $u \sim U(x_1 - \sqrt{3}\sigma, x_1 + \sqrt{3}\sigma)$ and set $(x'_1, x_2) = (u, x_2)$ with probability $\min\left(1, \frac{\pi(u, x_2)}{\pi(x_1, x_2)}\right)$, otherwise set $(x'_1, x_2) = (x_1, x_2)$. The proposal ratio is 1.
    2. Then, draw $v \sim U(x_2 - \sqrt{3}\sigma, x_2 + \sqrt{3}\sigma)$ and set $(x'_1, x'_2) = (x'_1, v)$ with probability $\min\left(1, \frac{\pi(x'_1, v)}{\pi(x'_1, x_2)}\right)$, otherwise set $(x'_1, x'_2) = (x'_1, x_2)$. The proposal ratio is 1.
  - K5. Two 1D Gaussian proposals. This is similar to the uniform one (K4), but with $u \sim N(x_1, \sigma^2)$ and $v \sim N(x_2, \sigma^2)$ instead.

For the $N_2(0, \Sigma)$ target, we consider four additional proposal kernels, based on the whitening transformation (8).

- K6. 2D Gaussian with transformation. Generate $y' \sim N(y, \sigma^2 I)$ and set $x' = \Sigma^{1/2} y'$. The proposal ratio is 1.
- K7. Two 1D Uniform with transformation. This is similar to K4, but uses the transformed variable y.
- K8. Two 1D Gaussian with transformation. This is similar to K7, but uses the Gaussian proposal instead of uniform.
- K9. Two 1D MirrorU with transformation. This is similar to K7, but uses the MirrorU proposal instead of uniform, with $\mu^* = 0.1$ for both components.

For the $N_2(0, I)$ target, the two two-dimensional versions of the uniform kernel, Square2D (K2) and Disc2D (K3), are more efficient than Gaussian2D (K1) (Table S1). This is apparently due to the fact that Gaussian2D is more concentrated on points close to the current point so that the samples are more strongly correlated. The efficiency is almost doubled when two one-dimensional proposals are used instead of a single two-dimensional move (compare Two1DUniform (K4) with Square2D and Disc2D, and Two1DGaussian (K5) with Gaussian2D in Table S1). The optimal σ and efficiency for these kernels agree with those for the $N(0, 1)$ target in Table 1. Note, however, that this improvement in statistical efficiency comes at an extra cost in computation that scales with the target's dimensionality. If the target is d-dimensional, a sequence of d 1D moves requires d evaluations of the target density and d MH acceptance steps instead of just one.

For the $N_2(0, \Sigma)$ target, applying kernels K1-K5 directly gives poor results, with efficiency of only $2 - 6\%$, compared with over 10% for the $N_2(0, I)$ target (Table S1 and Figure S2). This inefficiency is because these proposals fail to account for the high correlation ($\rho = 0.9$) between the variables in the target. When the correlation is removed via the whitening transformation (8) in TransfGaussian2D (K6), we recover the same efficiency of 0.134 as achieved by Gaussian2D (K1) on $N_2(0, I)$ target. The same pattern applies to the one-dimensional moves with transformation (8): Two1DTransfUnif (K7) achieves the same efficiency as Two1DUnif (K4) on $N_2(0, I)$ ($E = 0.276$), and Two1DTransfGaussian (K8) achieves the same efficiency as Two1DGaussian (K5) on $N_2(0, I)$ ($E = 0.228$). Note that simply using one-dimensional proposals without transformation can yield worse performance than the corresponding two-dimensional moves as correlations make it more difficult to make a large move along the axis-aligned directions. Finally, the Two1DTransfMirrorU (K9) kernel is several times more efficient than any other kernel considered.

III MCMC algorithms for the phylogenetic problem

	Kernel	$N_2(0, I)$					$N_2(0, \Sigma)$				
		optimal σ	P_{jump}	E	E_π^2	ρ_1	optimal σ	P_{jump}	E	E_π^2	ρ_1
K1	Gaussian2D	1.7	0.352	0.134	0.544	0.762	1.8	0.159	0.043	0.174	0.913
K2	Square2D	1.7	0.300	0.155	0.475	0.728	1.5	0.167	0.060	0.236	0.882
K3	Disc2D	1.7	0.294	0.156	0.552	0.724	1.5	0.153	0.048	0.192	0.904
K4	Two1DUniform	2.2	0.407	0.276	0.880	0.560	1.0	0.393	0.030	0.166	0.917
K5	Two1DGaussian	2.5	0.430	0.228	0.744	0.628	1.0	0.456	0.024	0.141	0.930
K6	TransfGaussian2D			n/a			1.7	0.352	0.134	0.475	0.762
K7	Two1DTransfUnif			n/a			2.2	0.407	0.276	0.880	0.560
K8	Two1DTransfGaussian			n/a			2.5	0.430	0.228	0.743	0.629
K9	Two1DTransfMirrorU ($\mu^* = 0.1$)			n/a			0.4	0.852	1.825	2.987	-0.494

Table S1:: Efficiency for estimating the mean of two-dimensional Gaussian targets.

NOTE: The $N_2(0, \Sigma)$ target has covariance $\Sigma = \left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right)$. Efficiency ($E$) is for estimating the mean of the first component $\mu_1 = \mathbf{E}(x_1)$.

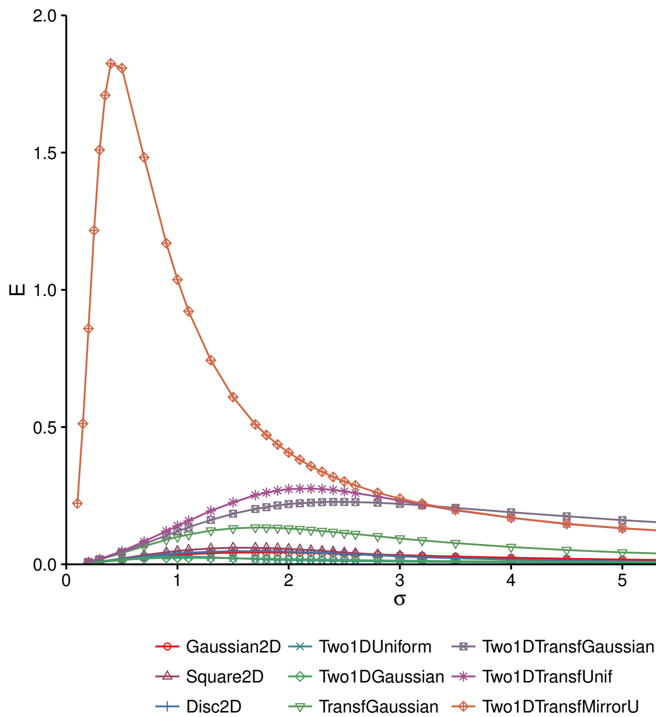

Figure S2: Efficiency of proposal kernels for the $N_2(0, \Sigma)$ target, with $\Sigma = \left(\begin{smallmatrix} 1 & 0.9 \\ 0.9 & 1 \end{smallmatrix}\right)$.

Algorithm A1 (1D Uniform on t, r). The algorithm consists of two MH steps. In step 1, draw $t'|t \sim U(t - \sigma_t\sqrt{3}, t + \sigma_t\sqrt{3})$. If $t' < 0$, set $t' \leftarrow -t'$ (proposal ratio is 1). In step 2, draw $r'|r \sim U(t - \sigma_r\sqrt{3}, t + \sigma_r\sqrt{3})$. If $r' < 0$, set $r' \leftarrow -r'$ (proposal ratio is 1).

The step-size parameters σ_t and σ_r are automatically tuned to achieve $P_{\text{jump}} = 0.4$ (see Section 2.3).

Algorithm A2 (1D Uniform on w, z). The algorithm consists of two MH steps. In step 1, draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $t' \leftarrow te^{\sigma_w u}$ (proposal ratio is $\frac{t'}{t}$). In step 2, draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $r' \leftarrow re^{\sigma_z v}$ (proposal ratio is $\frac{r'}{r}$).

The step-size parameters σ_w and σ_z are tuned to achieve $P_{\text{jump}} = 0.4$.

Algorithm A3 (2D Uniform on w, z). The algorithm uses a single two-dimensional proposal. First, draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $t' \leftarrow te^{\sigma_w u}$. Then draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $r' \leftarrow re^{\sigma_z v}$. The proposal ratio is $\frac{t'r'}{tr}$.

It is hard to adjust two step-sizes σ_w and σ_z in one proposal. We use $\sigma_w = s_w \times 2.2 \times \frac{1.7}{2.4}$ and $\sigma_z = s_z \times 2.2 \times \frac{1.7}{2.4}$, where s_w and s_z are the standard deviations of $w = \log t$ and $z = \log r$, estimated during burn-in. Here, 2.4 and 1.7 are optimal scales in 1D and 2D for the Gaussian kernel, while 2.2 is the optimal scale for the uniform kernel in 1D (tables 1 and S1).

Algorithm A4 (1D Uniform on w, z with whitening transformation (8)). Let $w := \log t, z := \log r$. Let $\widehat{\Sigma}$ denote the estimated covariance matrix of (w, z) during the burn-in. The algorithm consists of two one-dimensional MH steps on w and z.

1. Set $\binom{w}{z} \leftarrow \begin{pmatrix}\log t\\ \log r\end{pmatrix}$ and $\binom{\tilde{w}}{\tilde{z}} \leftarrow \widehat{\Sigma}^{-1/2}\binom{w}{z}$. Draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $\tilde{w}' \leftarrow \tilde{w} + \sigma_w u$ and $\tilde{z}' \leftarrow \tilde{z}$. Then set $\binom{w'}{z'} \leftarrow \widehat{\Sigma}^{1/2}\binom{\tilde{w}'}{\tilde{z}'}$ and $\binom{t'}{r'} \leftarrow \begin{pmatrix}e^{w'}\\ e^{z'}\end{pmatrix}$. The proposal ratio is $\frac{t'r'}{tr}$.
2. Set $\binom{w}{z} \leftarrow \begin{pmatrix}\log t\\ \log r\end{pmatrix}$ and $\binom{\tilde{w}}{\tilde{z}} \leftarrow \widehat{\Sigma}^{-1/2}\binom{w}{z}$. Draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $\tilde{w}' \leftarrow \tilde{w}$ and $\tilde{z}' \leftarrow \tilde{z} + \sigma_z v$. Then set $\binom{w'}{z'} \leftarrow \widehat{\Sigma}^{1/2}\binom{\tilde{w}'}{\tilde{z}'}$ and $\binom{t'}{r'} \leftarrow \begin{pmatrix}e^{w'}\\ e^{z'}\end{pmatrix}$. The proposal ratio is $\frac{t'r'}{tr}$.

The step-size parameters σ_w and σ_z are tuned to achieve $P_{\text{jump}} = 0.4$.

Algorithm A5 (1D Uniform on $x := \log(tr), y := \log(t/r)$). The algorithm consists of two one-dimensional MH steps on x and y.

1. Set $x \leftarrow \log(tr)$ and $y \leftarrow \log(t/r)$. Draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $x' \leftarrow x + \sigma_x u$ and $y' \leftarrow y$. Then set $t' \leftarrow e^{\frac{x'+y'}{2}}$ and $r' \leftarrow e^{\frac{x'-y'}{2}}$. The proposal ratio is $\frac{t'r'}{tr}$.

2. Set $x \leftarrow \log(tr)$ and $y \leftarrow \log(t/r)$. Draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $x' \leftarrow x$ and $y' \leftarrow y + \sigma_y v$. Then set $t' \leftarrow e^{\frac{x'+y'}{2}}$ and $r' \leftarrow e^{\frac{x'-y'}{2}}$. The proposal ratio is $\frac{t'r'}{tr} = 1$.

The step-size parameters σ_x and σ_y are tuned to achieve $P_{\text{jump}} = 0.4$.

Algorithm A6 (1D MirrorU on x, y). The algorithm consists of two one-dimensional MH steps on x and y.

1. Set $x \leftarrow \log(tr)$ and $y \leftarrow \log(t/r)$. Draw $x'|x \sim U(2\mu_x^* - x - \sigma_x\sqrt{3}, 2\mu_x^* - x + \sigma_x\sqrt{3})$ and set $y' \leftarrow y$. Then set $t' \leftarrow e^{\frac{x'+y'}{2}}$ and $r' \leftarrow e^{\frac{x'-y'}{2}}$. The proposal ratio is $\frac{t'r'}{tr}$.
2. Set $x \leftarrow \log(tr)$ and $y \leftarrow \log(t/r)$. Draw $y'|y \sim U(2\mu_y^* - y - \sigma_y\sqrt{3}, 2\mu_y^* - y + \sigma_y\sqrt{3})$ and set $x' \leftarrow x$. Then set $t' \leftarrow e^{\frac{x'+y'}{2}}$ and $r' \leftarrow e^{\frac{x'-y'}{2}}$. The proposal ratio is $\frac{t'r'}{tr} = 1$.

Here, μ_x^*, μ_y^* are set to the estimated means $\hat{\mu}_x, \hat{\mu}_y$ of x and y, respectively, and σ_x, σ_y are set to either $\hat{s}_x, \hat{s}_y$ (A6a) or $\frac{1}{2}\hat{s}_x, \frac{1}{2}\hat{s}_y$ (A6b), where $\hat{s}_x$ and $\hat{s}_y$ are the estimated standard deviations of x and y from the burn-in sample.

Algorithm A7 (1D MirrorU on w, z with whitening transformation). Let $\widehat{\Sigma}$ denote the estimated covariance matrix of (w, z) during burn-in. The algorithm consists of two MH steps.

1. Set $\binom{w}{z} \leftarrow \binom{\log t}{\log r}$ and $\binom{\tilde{w}}{\tilde{z}} \leftarrow \widehat{\Sigma}^{-1/2}\left(\binom{w}{z} - \binom{\hat{\mu}_w}{\hat{\mu}_z}\right)$. Draw $u \sim U(-\sqrt{3}, \sqrt{3})$ and set $\tilde{w}' \leftarrow -\tilde{w} + \sigma_w u$ and $\tilde{z}' \leftarrow \tilde{z}$. Then set $\binom{w'}{z'} \leftarrow \widehat{\Sigma}^{1/2}\binom{\tilde{w}'}{\tilde{z}'} + \binom{\hat{\mu}_w}{\hat{\mu}_z}$ and $\binom{t}{r} \leftarrow \binom{e^w}{e^z}$. The proposal ratio is $\frac{t'r'}{tr}$.
2. Set $\binom{w}{z} \leftarrow \binom{\log t}{\log r}$ and $\binom{\tilde{w}}{\tilde{z}} \leftarrow \widehat{\Sigma}^{-1/2}\left(\binom{w}{z} - \binom{\hat{\mu}_w}{\hat{\mu}_z}\right)$. Draw $v \sim U(-\sqrt{3}, \sqrt{3})$ and set $\tilde{w}' \leftarrow \tilde{w}$ and $\tilde{z}' \leftarrow -\tilde{z} + \sigma_z v$. Then set $\binom{w'}{z'} \leftarrow \widehat{\Sigma}^{1/2}\binom{\tilde{w}'}{\tilde{z}'} + \binom{\hat{\mu}_w}{\hat{\mu}_z}$ and $\binom{t}{r} \leftarrow \binom{e^w}{e^z}$. The proposal ratio is $\frac{t'r'}{tr}$.

The step-size parameters σ_w and σ_z are set to $\frac{1}{2}$.

Algorithm A8 (MALA with preconditioning on w, z). The algorithm uses a single two-dimensional proposal.

1. Set $(w, z) \leftarrow (\log t, \log r)$. Draw $(w', z') \sim N(m(w, z), \varepsilon^2 A)$, where
$$m(w, z) \coloneqq (w, z) + \frac{\varepsilon}{2} A \nabla \log p(w, z|x),$$
and the first derivatives are derived using (9). Then set $(t', r') \leftarrow (e^{w'}, e^{z'})$. The proposal ratio is $\frac{N((w,z)|(w',z') + \frac{\varepsilon^2}{2} A\nabla \log p(w',z'|x), \varepsilon^2 A)}{N((w',z')|(w,z) + \frac{\varepsilon^2}{2} A\nabla \log p(w,z|x), \varepsilon^2 A)} \frac{t'r'}{tr}$.

The scalar step-size parameter ε is tuned manually to achieve the highest efficiency. The matrix A is set to $\frac{1}{(\det \widehat{\Sigma})^{1/2}}\widehat{\Sigma}$, where $\widehat{\Sigma}$ is denotes the estimate of the target's covariance matrix from burn-in samples, following Marshall and Roberts (2012).

Algorithm A9 (HMC on w, z). Let $L_{\max}$ be the upper bound on the number of leapfrog steps and let ε be the leapfrog step-size. Let $\widehat{\Sigma}$ denote the estimated covariance matrix of (w, z) from burn-in. This algorithm uses a single two-dimensional proposal.

1. Set $(w, z) \leftarrow (\log t, \log r)$. Draw an auxiliary variable $\phi \sim N(0, \widehat{\Sigma})$. Set $(w', z') \leftarrow (w, z)$ and $\phi' \leftarrow \phi$. Draw $L \sim U\{1, \ldots, L_{\max}\}$. For $\ell = 1, \ldots, L$, (a) set $\phi' \leftarrow \phi' + \frac{\varepsilon}{2}\nabla \log p(w', z')$, (b) set $(w', z') \leftarrow (w', z') + \varepsilon\widehat{\Sigma}^{-1}\phi'$, and (c) set $\phi' \leftarrow \phi' + \frac{\varepsilon}{2}\nabla \log p(w', z')$. Then set $(t', r') \leftarrow (e^{w'}, e^{z'})$. The proposal ratio is $\frac{N(\phi'|0,\widehat{\Sigma})}{N(\phi|0,\widehat{\Sigma})}\frac{t'r'}{tr}$.

The parameters $L_{\max}$ and ε are tuned manually to achieve the highest efficiency.

Algorithm A10 (HMC (Stan) on w, z). NUTS algorithm. See Hoffman and Gelman (2014) for description.

Algorithm A11 (Manifold MALA on w, z). This algorithm uses a single two-dimensional proposal.

1. Set $(w, z) \leftarrow (\log t, \log r)$. Draw $(w', z') \sim N(m(w, z), \varepsilon^2 G^{-1}(w, z))$ where

$$m(w, z) \coloneqq (w, z) + \varepsilon^2\left(\frac{1}{2}G^{-1}(w, z)\nabla \log p(w, z|x) + \Omega(w, z)\right),$$

$$G(w, z) \coloneqq -\mathbf{E}_{p(x|w,z)}\nabla^2_{(w,z)} \log p(x|w, z) - \nabla^2_{(w,z)} \log p(w, z)$$

(the Fisher information matrix of the likelihood plus the negative Hessian of the log prior density), and

$$\Omega(w, z) \coloneqq \frac{1}{2}G^{-1}\begin{pmatrix}\operatorname{tr}\bigl(G^{-1}\partial_w G\bigr)\\ \operatorname{tr}\bigl(G^{-1}\partial_z G\bigr)\end{pmatrix} - \sum_{j=w,z}(G^{-1}\partial_j G)G^{-1}_{\cdot,j}.$$

Set $(t', r') \leftarrow (e^{w'}, e^{z'})$. The proposal ratio is $\frac{N((w,z)|m(w',z'),\varepsilon^2 G^{-1}(w',z'))}{N((w',z')|m(w,z),\varepsilon^2 G^{-1}(w,z))}\frac{t'r'}{tr}$.

The step-size parameter ε is tuned manually to achieve the highest efficiency.

Algorithm A12 (Manifold HMC on w, z). Let $L_{\max}$ be the upper bound on the number of leapfrog steps and let ε be the leapfrog step-size. Let M be the number of fixed point iterations for the generalized leapfrog integrator from Girolami and Calderhead (2011). This algorithm uses a single two-dimensional proposal.

1. Set $(w, z) \leftarrow (\log t, \log r)$. Draw an auxiliary variable $\phi \sim N(0, G)$. Set $(w', z') \leftarrow (w, z)$ and $\phi' \leftarrow \phi$. Draw $L \sim U\{1, \ldots, L_{\max}\}$. For $\ell = 1, \ldots, L$,

   (a) Set $\tilde{\phi} \leftarrow \phi'$. For $m = 1, \ldots, M$, set

   $$\tilde{\phi} \leftarrow \phi' + \frac{\varepsilon}{2}\left(\nabla \log p(w', z'|x) - \frac{1}{2}\mathrm{tr}(G^{-1}\nabla G) + \frac{1}{2}\tilde{\phi}^{\top} G^{-1}(\nabla G)G^{-1}\tilde{\phi}\right).$$

   Then set $\phi' \leftarrow \tilde{\phi}$.

   (b) Set $(\tilde{w}, \tilde{z}) \leftarrow (w', z')$. For $m = 1, \ldots, M$, set

   $$(\tilde{w}, \tilde{z}) \leftarrow (w', z') + \frac{\varepsilon}{2}\left(G^{-1}(w', z') + G^{-1}(\tilde{w}, \tilde{z})\right)\phi'.$$

   Then set $(w', z') \leftarrow (\tilde{w}, \tilde{z})$.

   (c) Set

   $$\phi' \leftarrow \phi' + \frac{\varepsilon}{2}\left(\nabla \log p(w', z'|x) - \frac{1}{2}\mathrm{tr}(G^{-1}\nabla G) + \frac{1}{2}\phi'^{\top} G^{-1}(\nabla G)G^{-1}\phi'\right).$$

   Then set $(t', r') \leftarrow (e^{w'}, e^{z'})$. The proposal ratio is $\frac{N(\phi'|0, G(w', z'))}{N(\phi|0, G(w, z))}\frac{t'r'}{tr}$.

The parameters $L_{\max}$ and ε are tuned manually to achieve the highest efficiency, and M is fixed to 3.

Note that the parameters of the model are t and r, as are the state of the Markov chain. Transformed variables w and z or x and y are used to design efficient moves in the t-r space.

IV Effect of μ^* on efficiency

We used the burn-in to estimate the means and variances of the posterior distribution. To assess the impact of this estimation on the efficiency of the chain, we performed 100 independent runs of algorithm A6b for the phylogenetic example. The means (μ_x, μ_y) and standard deviations (s_x, s_y) are estimated using four rounds during the burn-in, with each round consisting of 20,000 iterations. The estimates are then used to construct the mirror move. From Figure S3, we see that the mean and variance estimates from the burn-in are reasonably accurate, with mean efficiency 1.165 for t and 0.497 for r, slightly higher than values shown in Table 5.

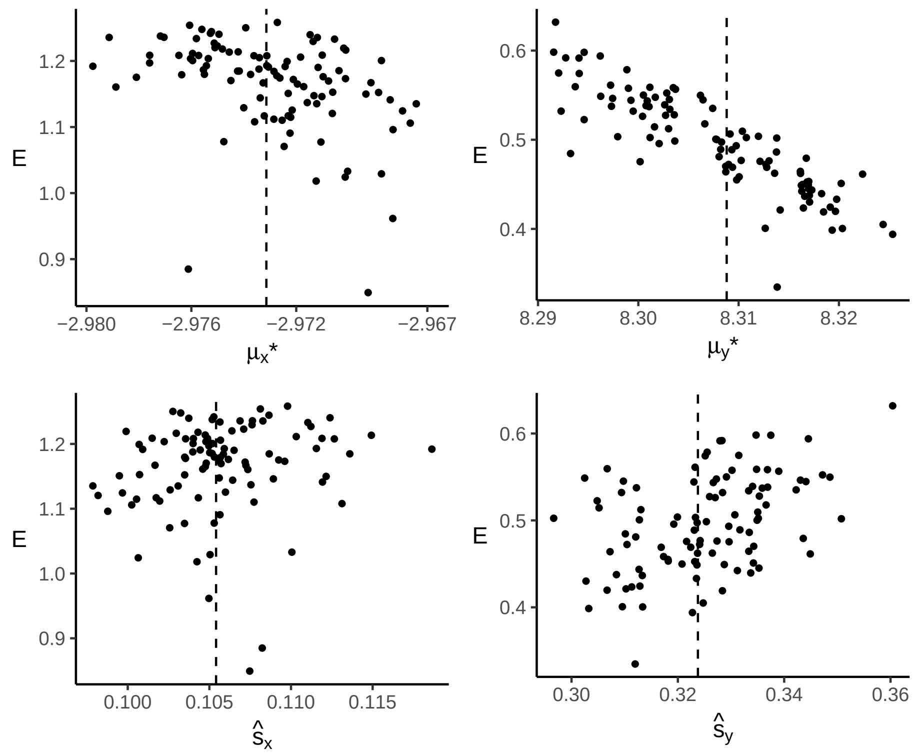

Figure S3: Efficiency (E) for estimating t (left column) and r (right column) over 100 replicate runs of kernel A6b in the phylogenetic example, plotted as a function of μ_x^*, μ_y^*, $\hat{s}_x$, and $\hat{s}_y$ estimates obtained from the burn-in.