5 Supplementary Methods for *A surrogate function for one-dimensional phylogenetic likelihoods* (doi:10.1093/molbev/msx253)

5.1 *Parameter regimes for the surrogate function*

For brevity, we define

$$\theta = \exp(r(t+b)).$$

We will assume that $r > 0$ and $b \geq 0$, so that θ as a function of non-negative t goes from some value greater than or equal to 1 up to infinity. Also note that $d\theta/dt = r\theta$ and $d\theta^{-1}/dt = -r\theta^{-1}$.

The surrogate function is defined as

$$\begin{aligned} &f(c, m, r, b; t) \\ &\quad = c\log((1+\theta^{-1})/2) + m\log((1-\theta^{-1})/2) \\ &\quad = c\log(1+\theta^{-1}) + m\log(1-\theta^{-1}) - (c+m)\log 2. \end{aligned}$$

As t goes to infinity, this has limit $-(c+m)\log 2$.

Taking the derivative,

$$\begin{aligned} df/dt &= -cr\theta^{-1}/(1+\theta^{-1}) + mr\theta^{-1}/(1-\theta^{-1}) \\ &= \frac{-cr}{\theta+1} + \frac{mr}{\theta-1} \\ &= r(-c\theta + c + m\theta + m)/(\theta^2 - 1) \\ &= r((m-c)\theta + m + c)/(\theta^2 - 1). \end{aligned}$$

So the first derivative is zero when (using subscript zero to denote maximum) $\theta_0 = (c+m)/(c-m)$; this gives a finite real solution for t when $c > m$. This is equivalent to

$$t_0 = -b + \log[(c+m)/(c-m)]/r. \tag{3}$$

For a more complete characterization of f, we also take the second derivative:

$$\frac{d^2 f}{dt^2} = r^2\theta\frac{(c-m)(\theta^2+1) - 2(c+m)\theta}{(\theta^2-1)^2}$$

This is zero when

$$\exp(r(t+b)) = \theta = \frac{(\sqrt{c} \pm \sqrt{m})^2}{c-m};$$

or

$$t = -b + \frac{1}{r}\log\left(\frac{(\sqrt{c} \pm \sqrt{m})^2}{c-m}\right); \tag{4}$$

We also note that $c > m$ implies $c - m > (\sqrt{c} - \sqrt{m})^2$, meaning that there can never be two positive solutions. With this we distinguish between four regimes:

1. one negative and one positive root of (4), $f(t)$ diverges at $t = 0$: $b = 0$, $c > m$ and $\exp(br) \leq (\sqrt{c} + \sqrt{m})^2)/(c-m)$

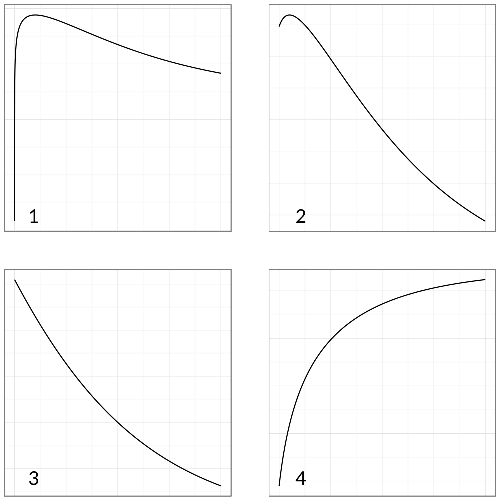


Figure S1: The various regimes of a likelihood function for the BSM parameterized by branch length.

2. one negative and one positive root of (4), $f(t)$ finite for all $t \geq 0$: $b > 0$, $c > m$ and $\exp(br) \leq (\sqrt{c} + \sqrt{m})^2)/(c - m)$
3. two negative solutions of (4): $c > m$ and $\exp(br) > (\sqrt{c} + \sqrt{m})^2/(c - m)$
4. no real solutions of (4): $c < m$.

This determines the shape of the likelihood curve up to the sign of the second derivative (Fig. S1) for positive t. Only in cases (1) and (2) are there inflection points. Only in cases (1) and (4) is the limit as t goes to zero infinite. In (3) and (4) the ML t is zero and infinity, respectively. Assuming a tree with finite branch lengths, note that the probability of having something in (4) goes to zero as sequences become long.

5.2 *lcfit2 parameterization*

An alternative parameterization to (1) can be useful. The "lcfit2" parameterization is in terms of c, m, the branch length t_0 giving the maximum value of the surrogate, and the second derivative at t_0. We assume that we are in parameter regime 1 or 2, so $c > m$.

We can re-express everything in terms of the difference from the ML branch length t_0 and eliminate b. Let $\tilde{t}$ be $t - t_0$ and $\tilde{\theta} = \exp(r(t - t_0))$. Note that $\theta = \tilde{\theta}\theta_0$, so we can re-express f in these terms, recalling that

$\theta_0 = (c+m)/(c-m)$:

$$\begin{aligned}
&f(c,m,r,t_0;\tilde{t})\\
&\quad = c\log\left(1+(\tilde{\theta}\theta_0)^{-1}\right) + m\log\left(1-(\tilde{\theta}\theta_0)^{-1}\right)\\
&\qquad\qquad - (c+m)\log 2\\
&\quad = c\log\left(1+\frac{c-m}{\tilde{\theta}(c+m)}\right) + m\log\left(1-\frac{c-m}{\tilde{\theta}(c+m)}\right)\\
&\qquad\qquad - (c+m)\log 2\\
&\quad = c\log\left(c+m+\frac{c-m}{\tilde{\theta}}\right) + m\log\left(c+m-\frac{c-m}{\tilde{\theta}}\right)\\
&\qquad\qquad - (c+m)\log(c+m) - (c+m)\log 2\\
&\quad = c\log\left(c+m+\frac{c-m}{\tilde{\theta}}\right) + m\log\left(c+m-\frac{c-m}{\tilde{\theta}}\right)\\
&\qquad\qquad - (c+m)\log(2(c+m))
\end{aligned}$$

Also recall

$$\begin{aligned}
f'(t) &= \frac{d}{dt}f(c,m,r,b;t) = \frac{-cr}{\theta+1} + \frac{mr}{\theta-1}\\
f''(t) &= \frac{d^2}{dt^2}f(c,m,r,b;t) = \frac{cr^2\theta}{(\theta+1)^2} + \frac{-mr^2\theta}{(\theta-1)^2}.
\end{aligned}$$

At the ML point t_0, note

$$\theta_0 + 1 = \frac{2c}{c-m} \qquad \theta_0 - 1 = \frac{2m}{c-m}$$

so

$$\frac{\theta_0}{(\theta_0+1)^2} = \frac{(c-m)(c+m)}{4c^2};\ \frac{\theta_0}{(\theta_0-1)^2} = \frac{(c-m)(c+m)}{4m^2}$$

and

$$\begin{aligned}
f''(t_0) &= r^2\left(\frac{(c-m)(c+m)}{4c} - \frac{(c-m)(c+m)}{4m}\right)\\
&= r^2\frac{(c-m)(c+m)}{4}\left(\frac{1}{c}-\frac{1}{m}\right)\\
&= r^2\frac{(c-m)(c+m)}{4}\left(\frac{m-c}{cm}\right)\\
&= -r^2\frac{(c-m)^2(c+m)}{4cm}.
\end{aligned}$$

So

$$r = \frac{2}{c-m}\sqrt{\frac{-f''(t_0)cm}{c+m}}.$$

5.3 *Sampling from the PDF corresponding to the surrogate function*

In the context of a Bayesian Monte Carlo algorithm, we can use the fit likelihood curve to quickly draw proposals from an approximate unnormalized posterior, which is simply the lcfit likelihood function times a prior. For example, we have found this useful in the context of sts. To draw such proposals, we can first use the procedure detailed above to fit an approximate likelihood curve and then use rejection sampling to draw from the approximate posterior.

Rejection sampling generates samples from an arbitrary distribution $h(x)$ using a proposal distribution $g(x)$ subject only to the constraint that $h(x) \leq cg(x)$ for some constant $c > 0$. Because the surrogate likelihood function is bounded above, we can use the prior as a proposal distribution. We now specialize to the case of an exponential prior. Let $h(t)$ be the unnormalized posterior on branch lengths

$$h(t) = \lambda e^{-\lambda t} F(t)$$

where $F(t) = e^{f(t)}$ is the surrogate likelihood function for some set of fit parameters. Let $g(t)$ be the PDF of the exponential distribution with rate λ,

$$g(t) = \lambda e^{-\lambda t}.$$

Clearly the ratio $h(t)/g(t) = F(t)$, so we choose c to be the maximum likelihood value

$$c = F(t_0)$$

where t_0 is the mode of the surrogate function and can be computed directly using (2). Then we have the ratio

$$\frac{h(t)}{cg(t)} = \frac{F(t)}{F(t_0)} \leq 1$$

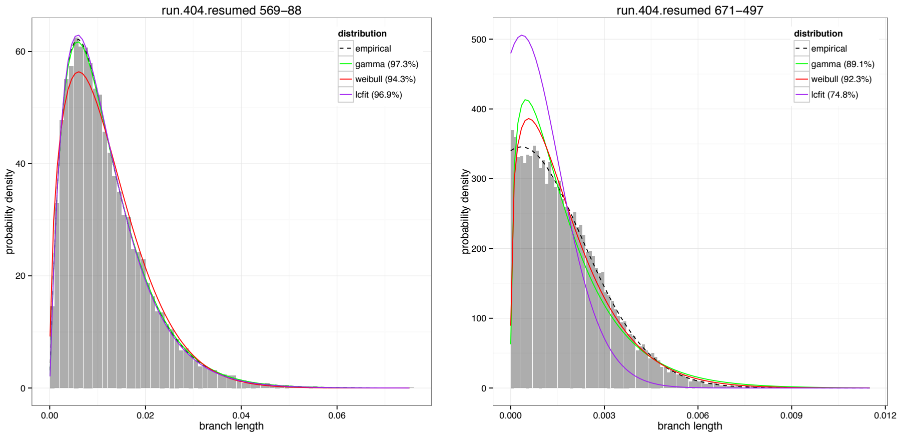
which satisfies the requirement for rejection sampling.

The procedure for generating a sample from the distribution begins by drawing a branch length t from the exponential distribution with rate λ and a value u from the uniform distribution over $(0, 1]$. If

$$u \leq \frac{h(t)}{cg(t)} = \frac{F(t)}{F(t_0)}$$

the sample is accepted; otherwise, the sample is rejected and the procedure is repeated. We note that eliminating the prior $g(t)$ from the acceptance calculation allows sampling from the distribution even when the maximum lcfit branch length is infinite (i.e., regime 4), since the asymptotic maximum likelihood can still be calculated.

Although this rejection sampling scheme can require many tries for certain curve shapes, individual evaluations of the surrogate function are very cheap so we have not found this to be a problem in practice.



(a) Good

(b) Bad

Figure S2: (a) An example good fit and (b) the worst fit of the surrogate function to empirical log-likelihood curves from the Aberer *et al.* (2016) data, presented as approximate posterior distributions given an exponential prior. Also shown are histograms of sampled branch lengths obtained from the ExaBayes MCMC sample after discarding burn-in. Expected acceptance rates of the corresponding lcfit, gamma, and Weibull distributions are also given.

5.4 *Fitting methods*

We have found a combination of two fitting methods to be useful. The first, which we call lcfit4, simply applies standard nonlinear least-squares optimization to find parameters for f using a sample of true values from the original likelihood function. The second, which we call lcfit2, uses the parameterization in terms of c, m, t_0, and $f''(t_0)$. In this case we simply set the t_0 and $f''(t_0)$ values to their values in the original function, then use least-squares fitting for c and m with a useful set of sampled points (inspired by Aberer *et al.* (2016); see below for details). For lcfit4, we first try unconstrained optimization using the Levenberg-Marquardt (L-M) algorithm (Levenberg, 1944; Marquardt, 1963) implemented in the GNU Scientific Library version 1.16 (Galassi and Gough, 2003). If the L-M algorithm fails to converge to a valid model, we fall back on constrained optimization using the SLSQP algorithm (Kraft, 1994) implemented in NLopt version 2.4.2 (Johnson, 2014). We have found that trying the L-M algorithm first yields better results in the case of four-parameter optimization than using SLSQP alone. For lcfit2, we use only the SLSQP algorithm, as we did not find the L-M step necessary to achieve good results. These two methods are used together as described below.

Next we describe the fitting process for these two methods in more detail. If one only wants a rough estimate of the likelihood curve, one can simply take a number of pre-chosen points, such as 0.05, 0.1, 0.5, and 1, calculate the corresponding likelihoods, and fit parameters of the curve using least squares as previously described. On the other hand, if a more accurate likelihood curve is desired, one can use an iterative algorithm to obtain an improved estimate of the likelihood curve around the maximum likelihood branch length. The idea of this process is to sample until the points enclose the maximum likelihood point. We will call this method “lcfit4 fitting”.

First, we fit the initial model:

1. Initialize with four values of t, and corresponding log likelihoods ℓ.
2. If the ℓ values are monotonically increasing, add a point: $t = 2\max(t)$, with corresponding log likelihood.
3. If the ℓ values are monotonically decreasing, add a point: $t = \min(t)/10$ with corresponding log-likelihood.
4. Repeat until the points enclose a maximum.

lcfit expects a minimum branch length $t_{\min}$ and maximum branch length $t_{\max}$ to consider.[1] Some phylogenetic libraries and applications enforce their own values. When such values are not available, we have found that a small but nonzero value for $t_{\min}$ (such as 10^{-6}) works well. For $t_{\max}$, choose a significantly large value at which the log-likelihood function can be expected to be nearly flat; we used 20. Note that excessively large values of $t_{\max}$ can affect numerical stability. The current implementation uses 0.1, 0.5, 1.0, and $t_{\max}$ as the first four starting points for t. $t_{\max}$ is included in these points to ensure that the fitted model exhibits good asymptotic performance. The procedure from the previous section is then used to find a starting point of BSM parameters for the optimization algorithm.

We then enter the second phase, which is repeated until the estimate of the ML branch length changes less than some fixed number. The first step is to find the maximum-likelihood branch length using (2) for the current BSM parameter estimates, and add it to the set of sampled values. The model is then re-fit using the optimization algorithm.

When the ML branch length is non-zero, we have found this method to be less robust to corner cases than we desired. Thus we have developed an alternative means of fitting, which we call “lcfit2 fitting”, that requires finding the ML value and the second derivative. As described in the main text and derived below, one can re-express the surrogate function in terms of the c and m parameters from before, along with the ML branch length t_0 of the surrogate function and its second derivative there. Then, one can simply set the t_0 and $f''(t_0)$ values of the surrogate function equal to the values found on the original function.

The procedure to find c and m for the lcfit2 surrogate after plugging in t_0 and $f''(t_0)$ is as follows. Starting with a default c and m,

[1] A reviewer has pointed out that likelihoods can be calculated directly at $t = 0$ and $t = +\infty$ because transition matrices take a simple form in both of these cases. This is an excellent point, however calculating these likelihoods would require knowing the partial likelihoods on either side of the edge, which are not available using the design of our software library. Indeed, our library assumes that only the likelihood function be available for evaluation at requested points. In any case, we also require a bounded range such that the log likelihood maximum is either in the range or less than $t_{\min}$ so that we can attempt to find the maximum using numerical methods. An alternative library design may be able to use log likelihood values at these extreme points.

1. calculate the inflection point t_i for the model.
2. define $\Delta = |t_0 - t_i|$.
3. let our four t values for fitting be $\{t_0 - \Delta, t_0, t_0 + \Delta, t_{\max}\}$; if either of $t_0 \pm \Delta$ are outside the interval $(t_{\min}, t_{\max})$ then replace them with half the distance from t_0 to the interval boundary.
4. fit c and m using these four points.
5. repeat steps 1–4 once more to refine the model.

Our complete fitting routine, using both lcfit2 and lcfit4, is as follows. First, maximize the original function on the set of non-negative t values. The maximum is found using Brent's method (Brent, 1973). Next, estimate the first and second derivatives at the maximum using fourth-order finite difference approximations (Davis and Polonsky, 1964, Table 25.2). If the first derivative is nonzero, use lcfit4 fitting, which we have found converges quickly in this case. If not, then use lcfit2 fitting. All least-squares fitting is done using the following gradient of f:

$$\begin{aligned}
df/dc &= \log\left(\frac{1}{2}(1+\theta^{-1})\right) \\
df/dm &= \log\left(\frac{1}{2}(1-\theta^{-1})\right) \\
df/dr &= (b+t)\frac{m(\theta+1)-c(\theta-1)}{\theta^2-1} \\
df/db &= r\frac{(m-c)\theta+c+m}{\theta^2-1}
\end{aligned}$$

We have also found it very advantageous to standardize the height of the surrogate function by subtracting the peak of the original function, so that we are fitting a curve that has maximum value zero. This leaves the asymptote free to vary.

Bad fits of the surrogate to the empirical log-likelihood function, such as the one seen in Fig. S2, generally appear to occur when the maximum-likelihood branch length t_0 is very short, and the second derivative there is negative (thus a positive inflection point t_i exists). These characteristics place the curve in regime 1 or 2 and are fit using the lcfit2 strategy. It is possible that these bad fits result from a deficiency in the fitting procedure. Recall that one of the points sampled during the lcfit2 procedure is to the left of t_0, ideally $t_0 - \Delta$ where $\Delta = |t_0 - t_i|$. However, when $t_0 - \Delta$ is less than $t_{\min}$, the point $(t_{\min} + t_0)/2$ is used instead. When t_0 is very small (i.e. close to $t_{\min}$), this point is still very near the peak, and may result in overfitting the peak at the expense of the rest of the curve.

5.5 *Extended methods: benchmarking*

We evaluated the performance of lcfit on both real and simulated data.

We used `nestly` (McCoy *et al.*, 2012) and the Bio++ 2.2.0 suite (Dutheil *et al.*, 2006; Dutheil and Boussau, 2008) of C++ libraries and binaries to perform simulation. We began by generating random 10-leaf bifurcating trees using the function `rtree` from the R package `ape` (Paradis *et al.*, 2004), with branch lengths sampled from an exponential distribution. We generated one set of trees with the exponential mean $\mu = 0.1$, and another set with $\mu = 0.01$. Each set contains 10 independent replicates. For each tree, we generated a 1000-site sequence alignment with `bppseqgen` from the Bio++ suite using an evolutionary model from Table S1 and a rate distribution of either uniform or discretized gamma ($n = 4, \alpha = 0.2$). We then optimized the branch lengths of each tree with `bppml`. The evolutionary model, tree, and alignment were then fed into our `lcfit-compare` utility. `lcfit-compare` loops over each branch of the tree and uses Bio++ to get an empirical log-likelihood function parameterized by the branch length. It then fits an lcfit model to the empirical log-likelihood function, and both the empirical and surrogate log-likelihood functions are sampled in the neighborhood of the peak.

We estimated the Kullback-Leibler (KL) divergence from the original likelihood function to the surrogate function by sampling these functions over 501 evenly spaced points in the neighborhood of the peak. This neighborhood is found as the region where the log-likelihood curve is above 10% of its peak value, bounded by $t_{\min}$ and $t_{\max}$. Probabilities are computed from the relative log-likelihoods as

$$P_i = \frac{\exp(\ell(t_i) - \ell(t_0))}{\sum_j [\exp(\ell(t_j) - \ell(t_0))]}$$

and

$$Q_i = \frac{\exp(f(t_i) - f(t_0))}{\sum_j [\exp(f(t_j) - f(t_0))]}$$

where t_0 is the maximum-likelihood branch length. The KL divergence from the discretized model distribution Q to the discretized empirical distribution P is then calculated as

$$D_{KL}(P \| Q) = \sum_i P_i \log_2 \left(\frac{P_i}{Q_i} \right).$$

Instructions for running these simulations and the analysis can be found in the `sims` subdirectory of the lcfit repository at `https://github.com/matsengrp/lcfit`.

We also tested the performance of lcfit on real data, in the manner of Aberer *et al.* (2016), and compared lcfit to the gamma and Weibull proposal distributions described in their work. To accomplish this, we incorporated lcfit fitting directly into the ExaBayes code used to generate data for their analysis. We then compared these results to the ExaBayes results, which were shared with us by André Aberer. Our fork of ExaBayes 1.3.1 used for these experiments can be found at `https://github.com/matsengrp/exabayes-1.3.1-lcfit`. We tested 12 out of the 14 DNA datasets they examined. One of the datasets not included in our analysis (dat-354) was missing gamma and Weibull distribution fit parameters in the data provided for some edges of the tree. The other dataset not included (dat-125) yielded a few invalid estimated acceptance rates (i.e., much greater than 100%). We attributed these errors to a numerical stability issue in the estimated acceptance rate calculations, and chose to omit the dataset from the analysis entirely rather than present a subset of its results. The remainder of the datasets contain between 24 and 500 taxa, with sequence lengths ranging from approximately 100 to 30,000 bases. We reproduced the estimated acceptance rate calculations for gamma and Weibull proposals using the method described in their supplemental material, then applied the same method to lcfit proposals. We then used the aggregated results to produce Fig. 2 (analogous to Fig. 2 in Aberer *et al.* (2016)).

5.6 *Relationship to entropy*

Here we establish a simple relationship between the ML value of the surrogate function and Shannon entropy of a corresponding sequence alignment under the BSM model. This is not used in practice, but is simply provided here for interest. Continuing in the setting of the lcfit2 parameterization and with that same notation,

$$\tilde{\theta}^{-1} = \exp(-r\tilde{t})$$

such that

$$f(\tilde{t}) = c\log\left(c+m+\nu\right) + m\log\left(c+m-\nu\right) - (c+m)\log(2(c+m))$$

where

$$\nu := \frac{c-m}{\tilde{\theta}}.$$

At $t = t_0$, $\tilde{\theta} = 1$, so the corresponding $\nu_0 = c - m$. Also,

$$\begin{aligned} f(t_0) &= c\log(c+m+\nu_0) + m\log(c+m-\nu_0) \\ &\qquad\qquad - (c+m)\log(2(c+m)) \\ &= c\log(2c) + m\log(2m) - (c+m)\log(2(c+m)) \\ &= c\log c + m\log m - (c+m)\log(c+m). \end{aligned}$$

Shannon entropy is defined as

$$S := -\sum_i p_i \log p_i.$$

Since the $(c + m)$ sites in the model are i.i.d., consider that the probability of observing a substitution at a single site is $p = m/(c + m)$, and the probability of observing no substitution is $1 - p = c/(c + m)$. Then

$$\begin{aligned} S &= -\left[(1-p)\log(1-p) + p\log p\right] \\ &= -\left[\frac{c}{c+m}\log\left(\frac{c}{c+m}\right) + \frac{m}{c+m}\log\left(\frac{m}{c+m}\right)\right] \\ &= -\frac{1}{c+m}\left[c\log c + m\log m - (c+m)\log(c+m)\right] \\ &= -\frac{1}{c+m}f(t_0). \end{aligned}$$

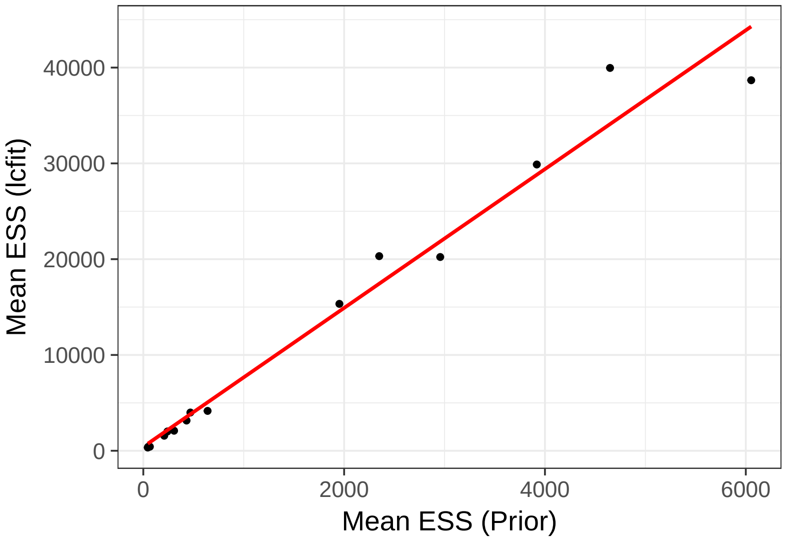


Figure S3: Mean effective sample size (ESS) using lcfit-based and prior-based proposals in the sequential Monte Carlo algorithm. The mean ESS is calculated using a combination of data sets of different sizes (10, 50, 100 sequences) and particle populations of varying sizes (750, 3750, 7500, 37500, and 75000 particles). The linear regression line relating both proposals is superimposed on the scatter plot and has a slope of about 7.

5.7 *sts experiment*

Figures S3 and S4 justify the claims in the Discussion concerning the improvement of using lcfit in sts versus drawing from the prior. For details on sts, we refer the reader to Fourment *et al.* (2017).

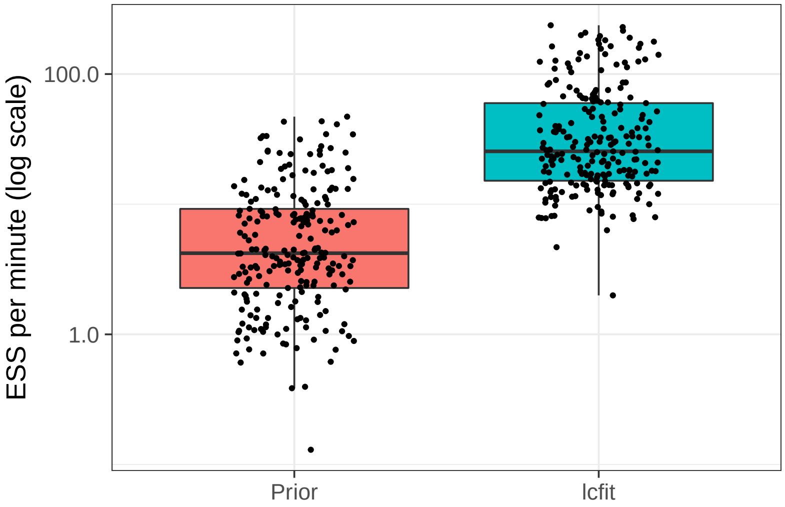


Figure S4: Effective sample size (ESS) per minute using lcfit-based and prior-based proposals in the sequential Monte Carlo algorithm. These results are for five data sets containing 100 sequences and 750, 3750, 7500, 37500, and 75000 particles.

Name	Data Type	Parameters	Reference
Binary-1.0	binary	$\kappa = 1$	see caption
Binary-4.0	binary	$\kappa = 4$	see caption
JC	DNA		(Jukes and Cantor, 1969)
HKY85	DNA	$\kappa = 2.0$, equal base freqs	(Hasegawa *et al.*, 1985)
JTT92	amino acid		(Jones *et al.*, 1992)
LG08	amino acid		(Le and Gascuel, 2008)
YN98	codon	$\kappa = 2.0, \omega = 5.0$	(Yang and Nielsen, 1998)
Nonhomogeneous	DNA	7 edges with T92 model, 6 edges with TN93 model, 5 edges with GTR	Tamura (1992); Tamura and Nei (1993); Tavaré (1986)

Table S1: The models used in Fig. 3. The binary model is parametrized as in the Bio++ documentation, such that a binary model with parameter κ has stationary distribution $(1/(\kappa + 1), \kappa/(\kappa + 1))$.

References

Aberer, A. J., Kobert, K., and Stamatakis, A. 2014. ExaBayes: massively parallel bayesian tree inference for the whole-genome era. *Mol. Biol. Evol.*, 31(10): 2553–2556.

Aberer, A. J., Stamatakis, A., and Ronquist, F. 2016. An efficient independence sampler for updating branches in bayesian markov chain monte carlo sampling of phylogenetic trees. *Syst. Biol.*, 65(1): 161–176.

Brent, R. 1973. *Algorithms for minimization without derivatives*. Prentice-Hall.

Davis, P. J. and Polonsky, I. 1964. Numerical interpolation, differentiation, and integration. In M. Abramowitz and I. A. Stegun, editors, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, chapter 25. U.S. Government Printing Office, tenth edition.

Dinh, V., Darling, A. E., and Matsen, IV, F. A. 2016. Online bayesian phylogenetic inference: theoretical foundations via sequential monte carlo.

Dutheil, J. and Boussau, B. 2008. Non-homogeneous models of sequence evolution in the Bio++ suite of libraries and programs. *BMC Evolutionary Biology*, 8(1): 255.

Dutheil, J., Gaillard, S., Bazin, E., Glémin, S., Ranwez, V., Galtier, N., and Belkhir, K. 2006. Bio++: a set of C++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics. *BMC bioinformatics*, 7(1): 188.

Fourment, M., Claywell, B. C., Dinh, V., McCoy, C., Matsen IV, F. A., and Darling, A. E. 2017. Effective online Bayesian phylogenetics via sequential Monte Carlo with guided proposals. *bioRxiv*, page 145219.

Galassi, M. and Gough, B. 2003. *GNU Scientific Library: Reference Manual : Edition 1.6*. Network Theory.

Gilks, W. R. and Wild, P. 1992. Adaptive rejection sampling for gibbs sampling. *J. R. Stat. Soc. Ser. C Appl. Stat.*, 41(2): 337–348.

Groussin, M., Boussau, B., and Gouy, M. 2013. A Branch-Heterogeneous model of protein evolution for efficient inference of ancestral sequences. *Syst. Biol.*

Hasegawa, M., Kishino, H., and Yano, T. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, 22(2): 160–174.

Johnson, S. G. 2014. The NLopt nonlinear-optimization package. `http://ab-initio.mit.edu/nlopt`.

Jones, D. T., Taylor, W. R., and Thornton, J. M. 1992. The rapid generation of mutation data matrices from protein sequences. *Comput. Appl. Biosci.*, 8(3): 275–282.

Jukes, T. H. and Cantor, C. R. 1969. Evolution of protein molecules. In H. N. Munro, editor, *Mammalian protein metabolism*, pages 21–132. Academic Press, New York.

Kraft, D. 1994. Algorithm 733: TOMP–Fortran modules for optimal control calculations. *ACM Trans. Math. Softw.*, 20(3): 262–281.

Lartillot, N. and Philippe, H. 2004. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Molecular biology and evolution*, 21(6): 1095–1109.

Le, S. Q. and Gascuel, O. 2008. An improved general amino acid replacement matrix. *Mol. Biol. Evol.*, 25(7): 1307–1320.

Levenberg, K. 1944. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2: 164–168.

Marquardt, D. 1963. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 11(2): 431–441.

McCoy, C., Gallagher, A., Hoffman, N., and Matsen, F. 2012. nestly– a framework for running software with nested parameter choices and aggregating results. *Bioinformatics*.

Paradis, E., Claude, J., and Strimmer, K. 2004. APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20: 289–290.

Semple, C. and Steel, M. 2003. *Phylogenetics*. Oxford University Press.

Tamura, K. 1992. Estimation of the number of nucleotide substitutions when there are strong transition-transversion and G+C-content biases. *Mol. Biol. Evol.*, 9(4): 678–687.

Tamura, K. and Nei, M. 1993. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol. Biol. Evol.*, 10(3): 512–526.

Tavaré, S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on mathematics in the life sciences*.

Wainwright, M. J. and Jordan, M. I. 2008. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2): 1–305.

Wang, H.-C., Susko, E., and Roger, A. J. 2014. An amino acid substitution-selection model adjusts residue fitness to improve phylogenetic estimation. *Mol. Biol. Evol.*

Yang, Z. and Nielsen, R. 1998. Synonymous and nonsynonymous rate variation in nuclear genes of mammals. *J. Mol. Evol.*, 46(4): 409–418.

Zoller, S. and Schneider, A. 2012. Improving phylogenetic inference with a semiempirical amino acid substitution model. *Molecular Biology and Evolution*.