A SURROGATE FUNCTION FOR ONE-DIMENSIONAL PHYLOGENETIC LIKELIHOODS

BRIAN C. CLAYWELL, VU C. DINH, CONNOR O. MCCOY, AND FREDERICK A. MATSEN IV

ABSTRACT. Phylogenetics has seen an steady increase in substitution model complexity, which requires increasing amounts of computational power to compute likelihoods. This model complexity motivates strategies to approximate the likelihood functions for branch length optimization and Bayesian sampling. In this paper, we develop an approximation to the one-dimensional likelihood function as parametrized by a single branch length. This new method uses a four-parameter surrogate function abstracted from the simplest phylogenetic likelihood function, the binary symmetric model. We show that it offers a surrogate that can be fit over a variety of branch lengths, that it is applicable to a wide variety of models and trees, and that it can be used effectively as a proposal mechanism for Bayesian sampling. The method is implemented as a stand-alone open-source C library for calling from phylogenetics algorithms; it has proven essential for good performance of our online phylogenetic algorithm sts.

1. INTRODUCTION

The increasing availability of large molecular sequence data sets poses a challenge for current phylogenetic algorithms. At the same time, phylogenetic substitution models are becoming more realistic and consequently, more complex (Lartillot and Philippe, 2004; Zoller and Schneider, 2012; Groussin et al., 2013; Wang et al., 2014). The combination of a large and increasing amount of phylogenetic likelihood calculation along with increasing complexity of models motivates research into useful approximations to the phylogenetic likelihood function.

One simple opportunity for efficiency improvement is in optimization of, or sampling from, the likelihood function as parametrized by a single branch length while fixing other parameters. In this case the likelihood function is simply a function that takes a non-negative real input and gives out another real number. One common approach for numerical maximization of such functions ℓ is to sample an ℓ at a number of points, fit a simple curve to those points, and then use the fit as an approximation to ℓ. We will call ℓ the original function and the fitted function f the surrogate function. Such an approach is useful if the original function is expensive to evaluate, but the surrogate function can be quickly fit to the sample points and evaluated. It is already being used implicitly in phylogenetics by inference programs that use Brent's method (Brent, 1973) for likelihood maximization, a method which effectively uses linear interpolation via the secant method. Recent work by Aberer et al. (2016) shows that proposals built using common probability distribution functions (PDFs) as surrogates, in particular the Γ distribution, can have high acceptance rates. Bayesian statistics in general has benefited from the

Date: June 5, 2017.

use of likelihood function approximations, such as for variational analysis (Wainwright and Jordan, 2008).

Although known functions can provide useful surrogates in phylogenetics, one might desire a class of surrogate functions that is specialized to the task. Indeed, phylogenetic likelihood functions parameterized by a single branch length have special characteristics: they asymptote as the branch length becomes long, and sometimes achieve infinite slope as the branch length becomes short. Neither of these features can be true for any polynomial, nor are they true for PDFs of common distribution functions.

In this paper, we show that a slight generalization of the likelihood function for the binary symmetric model (BSM) on a two-taxon tree can serve as a useful surrogate function for likelihood functions parameterized by branch lengths. We call this surrogate the lcfit function, short for "likelihood curve fit." With only four parameters, it can be easily and efficiently fit in a least-squares sense with standard algorithms; even more robust fitting can be achieved using the ML branch length and corresponding second derivative. We show via experiments with simulated and real data that it is readily fit and does a good job of approximating even complex models, making it a useful tool when those models are expensive to evaluate. Our code to use lcfit is available as an open-source C library.

2. RESULTS

2.1. **Surrogate formula and fitting.**

The lcfit surrogate function f evaluated at branch length t is

$$f(c, m, r, b; t) = c \log[(1 + e^{-r(t+b)})/2] + m \log[(1 - e^{-r(t+b)})/2] \tag{1}$$

for any positive values of the lcfit coefficients c, m, r, and non-negative b. It can be considered as an abstract surrogate function that takes a set of shapes resembling those of phylogenetic likelihood curves (Fig. 1). However, when b is zero this function is the log likelihood function for the binary symmetric model (BSM; see, e.g., Semple and Steel, 2003) where c is the number of constant sites, m is the number of substituted sites, and r is the substitution rate. The inclusion of the b term simply serves to truncate the likelihood function on the left, which is helpful in fitting likelihood functions for trees with more than two taxa. Indeed, without truncation the limit of f as branch lengths go to 0 is always negative infinity; this does not typically make for a good fit to likelihood functions parameterized by branches of non-trivial phylogenetic trees. As the branch length becomes long, f approaches an asymptote of $-(c + m) \log(2)$.

We will assume that $r > 0$ and $b \geq 0$, so that $e^{-r(t+b)}$ as a function of non-negative t goes from some positive value down to zero. The maximum of the log likelihood function for this setting is

$$t_0 = -b + \log[(c + m)/(c - m)]/r. \tag{2}$$

This has a finite real solution exactly when $c > m$. In the BSM interpretation this means that the number of constant sites strictly exceeds the number of substituted sites. Other characteristics of the lcfit function f are easily derived, such as the second derivative at the maximum, and the inflection point when it exists (see Supplement for formulas and derivations). Using such formulas we have found it useful in some cases to re-parameterize f in terms of the original c, m, f's maximum t_0, and the second derivative at this maximum value $f''(t_0)$.

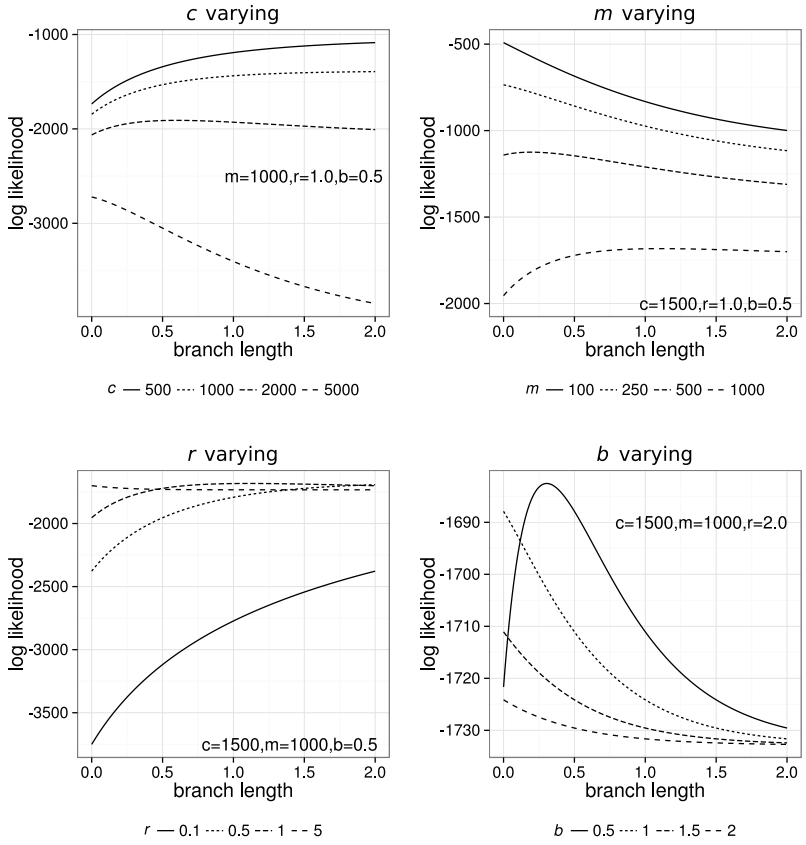


FIGURE 1. How each of the four parameters changes the shape of our surrogate function f defined in (1).

Briefly, our fitting methods combine two strategies to fit the parameters of the lcfit function (details provided in the Supplement). Both use least-squares fitting of sampled branch lengths and their likelihoods. The first strategy (lcfit2) applies when the maximum likelihood branch length is positive, and uses the second derivative at this branch length to eliminate two parameters so that only two parameters need to be fit. The second strategy (lcfit4) simply fits the lcfit parameters using least-squares directly.

We can simply multiply an lcfit curve by a branch length prior to get an approximate (unnormalized) PDF. For sampling from this lcfit PDF we have used a simple rejection sampling strategy with an exponential proposal distribution. Although this may require many proposals for an acceptance for certain lcfit shapes, individual lcfit evaluations are computationally cheap so we have not found this to be a significant burden in practice.

C library code with unit tests, continuous testing, simulation framework, and documentation is available at `https://github.com/matsengrp/lcfit`.

2.2. **Performance.** We obtain slightly better results than Aberer et al. (2016) in terms of acceptance rate for branch length proposals using their benchmarking

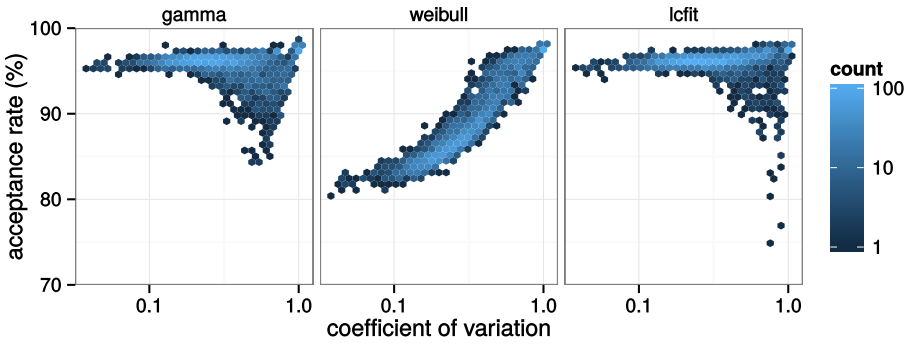


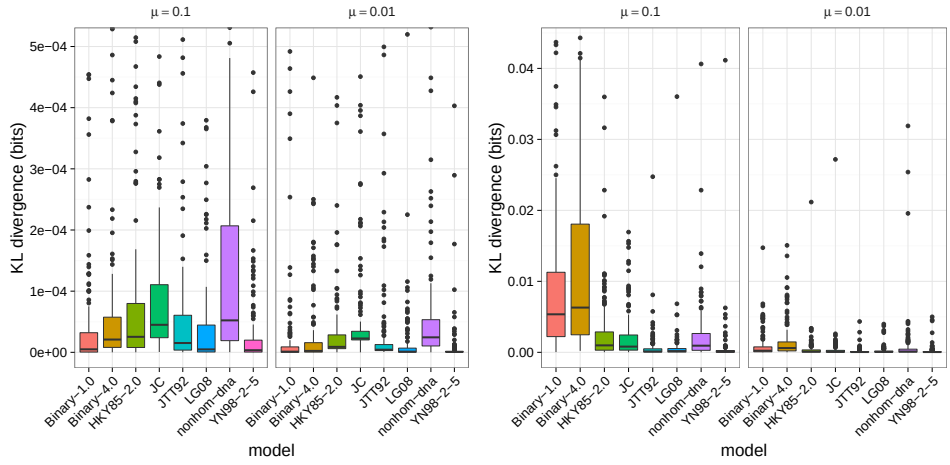
FIGURE 2. Expected acceptance rate for maximum-likelihood fits of gamma, Weibull, and lcfit distributions versus coefficient of variation of sampled single-branch-length posterior distributions for 12 datasets tested by Aberer et al. (2016). Fit parameters for the gamma and Weibull distributions were obtained directly from data provided by Aberer et al. (2016); those results reproduced here for comparison to lcfit.

strategy (Fig. 2). Briefly, we re-used their acceptance rate results for their Γ and Weibull proposals and used the same trees and likelihoods to compute the lcfit surrogate function (see Supplementary Methods for details). In terms of computational time, both our method and the method of Aberer et al. (2016) require the maximum of the likelihood function to be found, along with the second derivative. This computational effort dominates the required effort, and thus they are approximately equal in terms of computational cost.

We then performed simulation to explore how well the lcfit surrogate fits a broader range of models. To do so, we simulated data under a variety of models, and fit lcfit to the resulting likelihood curves under the same models. We quantified the divergence between the two curves using Kullback-Leibler (KL) divergence. We found that KL divergence for complex models is similar to KL divergence for data simulated under binary model (Fig. 3). Surprisingly, we found that lcfit performance by this metric was worse for variants of the binary model (e.g. the non-symmetric binary model or a mixture of rates) than for more complex models.

3. DISCUSSION

In this paper we present lcfit, the first surrogate function specialized to the case of one-dimensional phylogenetic likelihood functions, and how it can be useful. Our work shares goals with those of Aberer et al. (2016), however there are several aspects of our framework that make it appealing. This previous work uses several standard probability distributions as surrogate functions for posteriors. In particular, they fit normal, lognormal, Weibull, and Γ distributions to approximate per-branch posterior distributions in order to obtain efficient proposals. With the best performing of these distributions (typically Γ) they obtain high acceptance rates. However, there are inherent limitations using standard distributions. For example, the Γ and Weibull have two different shapes, depending on if their shape



(A) Uniform (B) Discretized Gamma

FIGURE 3. Estimated Kullback-Leibler divergence from the original likelihood function to the surrogate function. Simulations done using (a) uniform rates across sites and (b) discretized Gamma distributed rates across sites (4 categories, $\alpha = 0.2$). Branch lengths are either drawn from an exponential with mean either $\mu = 0.1$ or $\mu = 0.01$. See Table S1 for a list of model name abbreviations. Some outlier points excluded for clarity (Table S2).

parameter is greater and less than one; when the shape parameter is greater than one, the value at zero is zero, and when it is less than one then the first derivative at zero is negative. Neither of these need hold for phylogenetic likelihood curves or posteriors. Indeed, likelihood curves for internal branches are typically nonzero at zero and have a nonzero modes, for example, see Fig. 1c of Aberer et al. (2016). The truncated normal can take this shape, but its symmetry makes it a bad choice in this setting. In addition, lcfit matches real per-branch likelihoods by enabling a nonzero asymptote, whereas the Aberer et al. (2016) surrogates are all zero at infinity.

In addition to theoretical advantages of the lcfit framework, there are several practical advantages. Aberer et al. (2016) develop a fitting procedure using a linear relationship between the second derivative of the likelihood function and the standard deviation of the posterior density of the branch length. However, to use this relationship the parameters of this linear relationship must be inferred. Because it is inefficient to infer these parameters on the fly, Aberer et al. (2016) use consensus values and a somewhat complex tuning procedure, whereas in most cases we simply fit two coefficients using standard least-squares methods. We also note that lcfit is implemented as a stand-alone library for incorporation into other software, whereas the independence sampler of Aberer et al. (2016) is baked into ExaBayes (Aberer et al., 2014).

We have found lcfit to be essential for an efficient implementation (Fourment et al., 2017) of Online Phylogenetic Sequential Monte Carlo (Dinh et al., 2016); this

work also points the way to needed extensions. Here we have focused on approximating phylogenetic likelihood as a function of a single branch length at a time, but one could similarly concoct surrogate functions for other low-dimensional settings. For example, one could maximize three branches around an internal node by using a surrogate function based on the BSM likelihood function for a three taxon tree, or consider branch length changes and nearest-neighbor interchange moves simultaneously by using a surrogate function based on the BSM likelihood function for a four taxon tree.

4. ACKNOWLEDGEMENTS

The authors would like to thank Steve Evans, Vladimir Minin, Aaron Darling, Chris Warth, André Aberer, Julien Dutheil and Bastien Boussau. This work was supported by National Science Foundation awards DMS-1223057 and CISE-1564137, and National Institutes of Health grant U54GM111274. The research of Frederick Matsen was supported in part by a Faculty Scholar grant from the Howard Hughes Medical Institute and the Simons Foundation.

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. C., McCoy, C. O., Matsen IV, F. A., and Darling, A. E. 2017. Effective online Bayesian phylogenetics via sequential Monte Carlo with guided proposals. In preparation.

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

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.

5. Supplementary Methods

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

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

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

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

or

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

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 $\exp(br) \leq (\sqrt{c} + \sqrt{m})^2)/(c-m)$
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

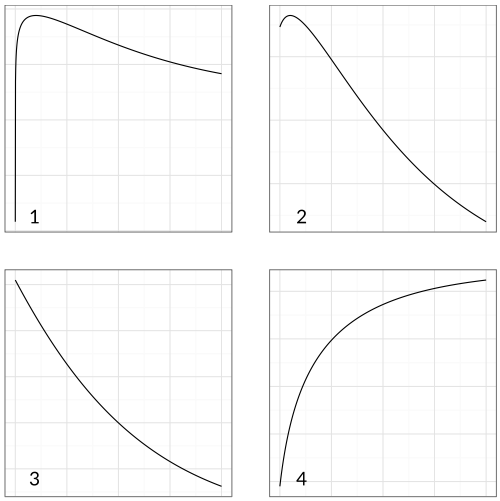


FIGURE S1. The various regimes of a likelihood function for the BSM parameterized by branch length.

lengths, note that the probability of having something in (4) goes to zero as sequences become long.

5.2. **lcfit2 parameterization.** It can be useful to use an alternative parameterization to 1. 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 $\theta_0 = (c+m)/(c-m)$:

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

Also recall

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

At the ML point t_0, note

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

so

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

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 online-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 $c > 0$. For 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).

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.

5.4. **Fitting methods.** We have found it useful to use a combination of two methods for fitting. 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. 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) 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.

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,

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

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

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

6. SUPPLEMENTARY TABLES

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 $(1/(\kappa + 1), \kappa/(\kappa + 1))$.

	Rate Distribution	Branch Length Mean	Count	Plot Threshold	Mean	Median	Maximum
1	gamma4-0.2	mu == 0.1	23	0.0415	0.0673	0.0577	0.2485
2	gamma4-0.2	mu == 0.01	5	0.0415	0.1011	0.0898	0.1235
3	uniform	mu == 0.1	123	0.0005	0.0073	0.0013	0.2020
4	uniform	mu == 0.01	65	0.0005	0.0629	0.0016	3.6035

TABLE S2. Outlier points excluded from Fig. 3. The counts are out of 1370 edges evaluated for each rate distribution/branch length mean combination.